# Cancer Molecular Subtype Classification by Graph Convolutional Networks on Multi-omics Data

Bingjun Li
bingjun.li@uconn.edu
University of Connecticut
Storrs, Connecticut, USA

Tianyu Wang
tianyu.wang@uconn.edu
University of Connecticut
Storrs, Connecticut, USA

Sheida Nabavi
sheida.nabavi@uconn.edu
University of Connecticut
Storrs, Connecticut, USA

## ABSTRACT

Cancer has been a second leading cause of death in the United States for decades and an accurate classifier of cancers' molecular profiles is a key predictor for patients' survival. Recently The Cancer Genome Atlas research networks have identified a new cancer taxonomy based on molecular tumor subtypes over 33 types of cancer. Several studies have reported classification models for traditional tissue-of-origin cancer type classification or classification of subtypes of a cancer type. In this study, we propose a novel end-to-end deep learning model that incorporates prior biological knowledge into the model and integrates multi-omics data to classify pan-cancer molecular subtypes. Our proposed model consists of three sections: i) a graph convolutional network that takes a genet interaction network, representing prior knowledge, as its input graph where genes are nodes and multi-omics data are the node features, to extract localized features; ii) a fully connected neural network to extract global features from the data; and iii) a classification layer that takes the combination of localized features and global features as input. We examined building the input graph using gene-gene interaction networks, protein-protein interaction networks, and gene co-expression networks. We also investigated the effect of input graph size (number of genes/nodes) on the performance of the model. We evaluated the performance of the proposed model in terms of prediction accuracy, precision, recall, and F1 score; and compared the performance of our model with those of three state-of-the-art deep learning models and two conventional machine learning models. The results show that the proposed model outperforms the baseline models at each level of the number of genes. Our model achieves not only a better prediction accuracy but also a lower false-negative rate, which is important for cancer patients treatments. Our model also shows the benefit of employing multi-omics data compared with employing only single-omic data.

## CCS CONCEPTS

• **Applied computing** → **Computational genomics**; **Bioinformatics**; • **Computing methodologies** → *Feature selection.*

## KEYWORDS

Deep Learning, Graph Convolutional Network, Multi-omics Data, Cancer Classification, TCGA

**ACM Reference Format:**
Bingjun Li, Tianyu Wang, and Sheida Nabavi. 2021. Cancer Molecular Subtype Classification by Graph Convolutional Networks on Multi-omics Data. In *12th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics (BCB '21), August 1–4, 2021, Gainesville, FL, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3459930.3469542


*BCB '21, August 1–4, 2021, Gainesville, FL, USA*

ACM ISBN 978-1-4503-8450-6/21/08…$15.00
https://doi.org/10.1145/3459930.3469542

## 1 INTRODUCTION

Cancer has been the second leading cause of death in the United States for almost 90 years according to the Centers for Disease Control [3, 9]. It is predicted that there would be about 1.9 million new cancer cases and over 608 thousand cancer deaths in 2021 [16]. Based on the data from 2015-2017, 40.5% of the male Americans and 38.9% of the female Americans are expected to be diagnosed with invasive cancer at least once in their lifetime. Researches have shown that early-stage diagnosis is a key predictor of the patients' survival rate and has a significant impact on the society; and an accurate and insightful classification of cancers is the foundation of early-stage diagnosis [8, 13]. In 2014, The Cancer Genome Atlas (TCGA) Research Network proposed a new clustering method of cancers based on their integrated molecular subtypes that share mutations, copy-number alterations, pathway commonalities, and micro-environment characteristics instead of their tissue of origin, and found 11 subtypes from 12 cancer types [7]. In 2018, the same group applied the new clustering method to 9,759 samples in TCGA and found 28 molecular subtypes from 33 cancer types [6]. The study estimated that about 10% of the cancer patients would be classified and treated differently if this new kind of molecular subtype classification is adopted.

Our work is inspired by the TCGA Research Networks's new cancer taxonomy and is aimed to provide a powerful classifier model to predict the molecular subtypes. There have been multiple studies of using machine learning models, deep learning models or various conventional methods for traditional cancer type classification and stage diagnosis [1, 10–12, 15, 22], but there isn't much research focusing on the classification based on molecular subtypes, especially with the use of multi-omics data. In 2017, Li et al. proposed a $k$-nearest neighbor model with a genetic algorithm for gene selection that yielded an overall 95.6% classification accuracy for 31 cancer types using TCGA dataset [11]. Lyu et al. proposed a convolutional neural network (CNN) model after embedding RNA-seq data into 2-D images that yielded a 95.59% classification accuracy for all 33 cancer types on TCGA samples [12]. In 2020, Ramirez et

al. proposed a graph convolution network (GCN) with prior knowledge in the form of protein-protein interaction networks and gene co-expression networks and obtained a 94.71% classification accuracy for 33 cancer types and normal tissue on TCGA data [15]. In the same year, Chen et al. proposed a Fusion Lasso framework for stage and subtype classification on multi-omics data. Their method formulates variable selection and data integration as a weighted constrained optimization problem [1].

Traditional convolutional networks are only suitable to extract hidden patterns of data in the Euclidean domain [21]. Due to the complex nature of biological organisms, the inner structure among genes is better represented in the form of graphs instead of in the Euclidean domain. Since many researchers have similar cases of applications to handle data with a graph structure, new ways of generalized convolution on graph data have been studied extensively and evolved rapidly in recent years. Graph neural networks (GNN) in the early days learn a node's features by iteratively propagating information from the neighboring nodes until convergence [21]. Two major disadvantages of these models are high computation costs and the learning filters' lack of the localization property. In 2016, Defferrard et al. proposed a spectral-based GCN (ChebNet) using Chebyshev polynomial as localized learning filters and for reducing the computation cost into linear complexity [2]. The proposed model in this study was inspired by the GCN model with fast localized filters and was developed upon the ChebNet as a foundation.

In this work, we propose a novel end-to-end deep learning model that incorporates prior biological knowledge, such as gene-gene interaction (GGI) networks, protein-protein interaction (PPI) networks, or gene co-expression networks, and integrates multi-omics data for molecular subtype classification. Based on our previous work on utilizing a GCN for classifying cells using single-omic single-cell data [19], we developed a GCN-based model to integrate multi-omics data. We trained and tested the proposed model under different conditions on TCGA data, which consists of 9,759 samples with several types of genomic data including gene expression and copy number variation (CNV) data. We generated the prior knowledge graphs from GGI dataset downloaded from the BioGrid database, and PPI and co-expression datasets downloaded from the STRING database [14, 17]. We assessed the performance of the proposed model employing six different knowledge graphs generated by GGI, PPI, and gene co-expression networks with and without single nodes (singleton) in terms of classification accuracy, precision, F1 score and recall. We examined the effect of integrating multi-omics data in comparison with using only gene expression data. We also compared the performance of the proposed model with those of other state-of-the-art deep learning models, such as fully connected neural networks (FC-NN) and convolutional neural network (CNN), and also conventional machine learning models such as random forest (RF) and support vector machine (SVM).

Our main contributions are listed as following:

- A novel end-to-end GCN-based classifier with both localized and global learning filters that incorporates prior knowledge.
- A deep learning classifier that integrates multi-omics data.
- A cancer molecular subtype classifier that provides a more insightful and accurate molecular similarity representation compared to traditional tissue-of-origin cancer taxonomy.

## 2 METHODS

Our proposed model consists of three sections: i) a GCN to extract localized features from multi-omics data based on prior knowledge of interactions among genes; ii) a shallow fully connected (FC) neural network to extract global features; and iii) a classification layer to concatenate the localized and global features and make a class prediction. The overall structure of the proposed model is shown in Figure 1.

### 2.1 Prior Networks

We built three weighted gene adjacency matrices (GGI, PPI, and co-expression) to represent the input graph. For each adjacency matrix, we considered two variations: with and without singleton – a total of six input graphs for the GCN model. All adjacency matrices are $N \times N$, where $N$ is the number of selected genes. The GGI network was built from the data provided by the BioGrid database [14], and both the PPI network and co-expression network were built from the data provided by the STRING database [17]. All genes are assumed to be self-connected in the input graph, thus the diagonal elements of all the adjacency matrices in the study are 1.

**GGI adjacency matrix**: The element, $A_{ij}$, in the GGI adjacency matrix $\mathbf{A} \in R^{N \times N}$ is such that

$$A_{ij} = \begin{cases} 1 & \text{if there is a connection between } i\text{th and } j\text{th genes} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Genes with no connection are considered singletons. We selected $N$ genes with the highest variations in their expression values, which we will go into more details in the Experimental Result: Datasets & Data Preprocessing section.

**PPI adjacency matrix**: The elements in the PPI adjacency matrix represent interactions among proteins from the STRING protein dataset. We normalized the interactions to 0-1 scale and we kept only strong interactions (>0.6) [18].

**Co-expression adjacency matrix**: The co-expression similarities between genes from the the STRING dataset were used to generate this adjacency matrix. We filtered out weak interactions (correlation <0.6).

### 2.2 Graph Convolutional Network

As shown in Figure 1, we developed a graph autoencoder model to fully utilize the prior network knowledge and to integrate different omics data. The input graph of the model is the knowledge graph represented by an adjacency matrix described in the previous section. In the input graph, nodes are genes and edges are assigned by the adjacency matrix. Each node is represented by a feature vector that is a combination of both gene expression and CNV data. The encoder part of the graph autoencoder model consists of a graph convolutional layer with a max-pooling layer, a flatten layer and a FC layer. The decoder part of the graph autoencoder model consists of one FC layer to reconstruct the input features.

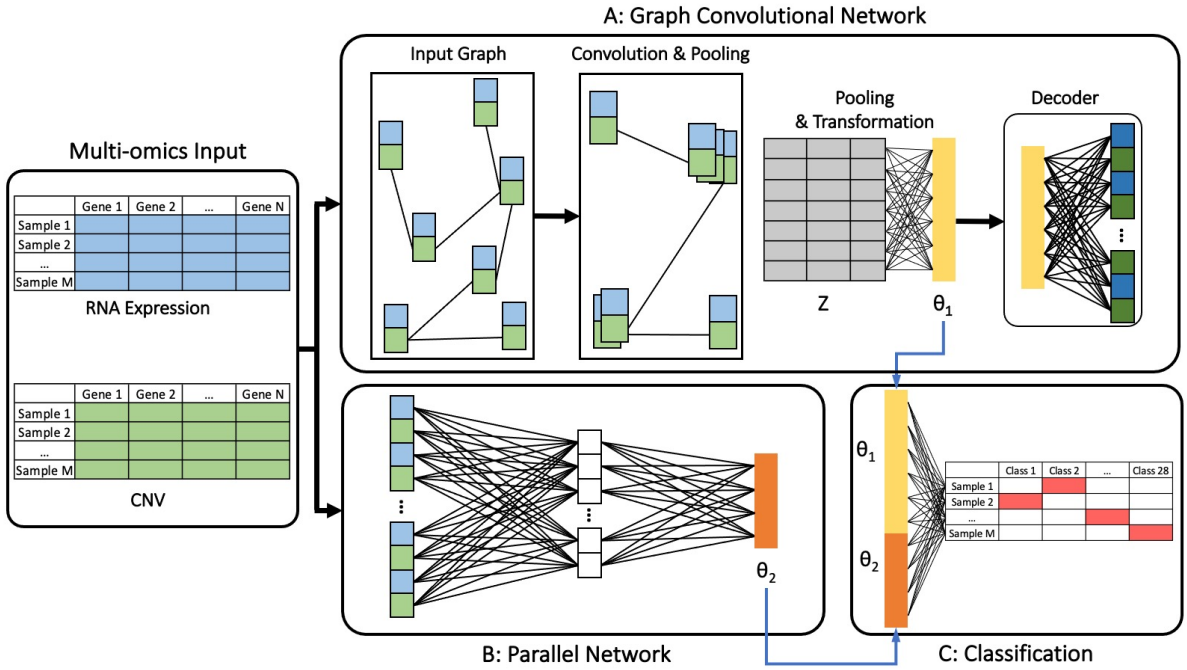


**Figure 1: The overall structure of our proposed model is separated into three major parts. Top section (Box A) is the graph convolutional network (Section 2.2), which is in a form of graph autoencoder and uses a decoder for training purpose. Each node as a pair of blue and green boxes represents the gene expression data and the CNV data on that node. The darker blue and darker green boxes in the decoder represent reconstructed data. Bottom left section (Box B) is a FC neural networks (Section 2.3). Bottom right section (Box C) is the classification layer (Section 2.4).**

We used the ChebNet approach to implement the proposed GCN model [2]. ChebNet is a computationally efficient approximation of GCNs that uses the Chebyshev's polynomial to generate localized filters instead of global filters. For our problem setup, let's denote the multidimensional input matrix as $\mathbf{X} \in R^{N \times M \times O}$, where $N$ is the number of genes, $M$ is the number of samples, and $O$ is the number of omics. The graph can be represented as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a list of vertices with the dimension of $O$ and $\mathcal{E}$ is a list of the edges between the vertices, the connections among the genes. The adjacency matrix generated in the previous section 2.1, $\mathbf{A} \in R^{N \times N}$, is used to represent the edges. The normalized Laplacian matrix can be expressed as

$$\mathbf{L} = \mathbf{I} + \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{1/2}, \tag{2}$$

where $\mathbf{I} \in R^{N \times N}$ is an identity matrix, and $\mathbf{D} \in R^{N \times N}$, the degree matrix, is a diagonal matrix. The diagonal elements in $\mathbf{D}$ represent the number of edges that connect to a node. The normalized Laplacian $\mathbf{L}$ is a real symmetric positive-semidefinite matrix and thus it allows an eigendecomposition of itself as

$$\mathbf{L} = \mathbf{U}\Lambda\mathbf{U}^T, \tag{3}$$

where $\mathbf{U} = (\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n)$ represents $n$ orthonormal eigenvectors of $\mathbf{L}$, $\mathbf{U}\mathbf{U}^T = \mathbf{I}$, and $\Lambda = diag(\lambda_1, \lambda_2, \ldots, \lambda_n)$ represents the eigenvalue matrix [2].

The graph Fourier transformation is defined as $\hat{\mathbf{X}}_j = \mathbf{U}^T\mathbf{X}_j$, $j = 1, \ldots, M$ for a sample $\mathbf{X}_j \in R^{N \times O}$, which is the feature matrix of $j$-th sample in our case. Then, the inverse Fourier transformation can be written as $\mathbf{X}_j = \mathbf{U}\hat{\mathbf{X}}_j$. Thus, graph convolution is defined in the Fourier domain as

$$\mathbf{X}_j * \mathbf{h} = \mathbf{U}(\mathbf{U}^T\mathbf{h} \odot \mathbf{U}^T\mathbf{X}_j), \tag{4}$$

where $\odot$ is the Hadamard (element-wise) product. Thus, it follows that $\mathbf{X}_j * \mathbf{h} = \mathbf{U}h(\Lambda)\mathbf{U}^T\mathbf{X}_j$, where $h(\Lambda)$, a non-parametric filter, is a diagonal matrix. The non-parametric filter has two major disadvantages, one is it is not localized in space and the other is its learning complexity is $O(N)$, where $N$ is the feature dimension of one sample [2]. To overcome these two disadvantages, the non-parametric filter is approximated by Chebyshev's polynomial

$$h(\Lambda) = \sum_{k=0}^{K-1} \beta_k T_k(\tilde{\Lambda}), \tag{5}$$

where $\beta_k$ is a parameter that is learned in training, $\tilde{\Lambda} = 2\Lambda/\lambda_{max} - \mathbf{I}$ is the rescaled diagonal eigenvalue matrix of $\mathbf{L}$, and $T_k(\tilde{\Lambda})$ is the $k^{th}$ order of the Chebyshev polynomial which can be computed by the stable recurrence relation of $T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$ with $T_0 = 1$ and $T_1 = x$ [5]. This spectral filter by $K^{th}$-order polynomials of the Laplacian is exactly $K$-localized which means it learns information from $K^{th}$-order of neighbours [2].

By substituting equation 5 into equation 4, we can rewrite the learning filter as

$$\mathbf{X}_j * \mathbf{h} = \mathbf{U} \sum_{k=1}^{K-1} \beta_k T_k(\tilde{\mathbf{\Lambda}}) \mathbf{U}^T \mathbf{X}_j = \sum_{k=0}^{K-1} \beta_k T_k(\tilde{\mathbf{L}} \mathbf{X}_j), \tag{6}$$

where $\tilde{\mathbf{L}} = 2\mathbf{L}/\lambda_{max} - \mathbf{I}$, and $T_k(\tilde{\mathbf{L}}) = 2\tilde{\mathbf{L}} T_{k-1}(\tilde{\mathbf{L}}) - T_{k-2}(\tilde{\mathbf{L}})$ with $T_0(\tilde{\mathbf{L}}) = \mathbf{I}$ and $T_1(\tilde{\mathbf{L}}) = \tilde{\mathbf{L}}$. Applying this localized learning filter greatly reduces the computation cost of the graph convolution. $K$ can be considered as a hyper-parameter which is set to 5 in our study. By using multiple kernels ($F = 5$), the output of the graph convolutional layer is $N \times F$. Then a maxpooling layer with $p = 8$ is used to reduce the number of nodes.

The output of the graph convolutional and pooling layers is passed through an activation function to obtain a lower-dimensional sample representation, $\mathbf{Z}$, for the multi-omics data as

$$\mathbf{Z}^T = \sigma\left(\sum_{k=0}^{K-1} \beta_k T_k(\tilde{\mathbf{L}}) \mathbf{X}_j\right), \tag{7}$$

where $\sigma(\cdot)$ represents the activation function, and $\mathbf{Z}$ is in $R^{N/p \times F}$. Then, $\mathbf{Z}$ is connected to a flatten layer that transforms a matrix in $R^{n \times m}$ to a vector of size $nm$ and a FC layer. The final output, indicated as $\boldsymbol{\theta_1}$, represents the extracted features. It is the input to the decoder part of the graph autoencoder, and a FC layer is used to reconstruct the input data $\tilde{\mathbf{X}}_j$.

## 2.3 Parallel Fully Connected Network

As demonstrated in the previous section, the quality of the extracted features by the graph convolutional layer depends on the completeness of the prior knowledge –the genomics interaction network in this case– due to the localization property of the learning filters in the ChebNet. Since the knowledge network we used in the GCN is not a complete gene interaction network, using only GCN as a feature extractor would neglect some global patterns in the data. On the other hand, a FC network is able to extract global features of the data while neglecting the inner interactions of genes. To overcome the limitations of both methods and to obtain a better overall performance of the classification model we used an FC network in parallel to the GCN model shown as Box B in Figure 1. Combining the localized extracted features and the global extracted features will compensate the limitations of GCN and FC networks for classification.

Each input of the FC network, $\mathbf{x}_C$, is a vector concatenating multi-omics data as expressed bellow.

$$\begin{aligned} \mathbf{x}_C &= [\mathbf{x}_1^T, \mathbf{x}_2^T, \ldots, \mathbf{x}_N^T] \\ &= [\mathrm{Exp}_1, \mathrm{CNV}_1, \mathrm{Exp}_2, \mathrm{CNV}_2, \ldots, \mathrm{Exp}_N, \mathrm{CNV}_N] \in R^{2N} \end{aligned} \tag{8}$$

The output of the FC layer, $\boldsymbol{\theta_2}$ is a vector. To further examining the effectiveness of the parallel FC network, we also conducted a model without the parallel network and used it as a baseline model in Section: Experimental Results 3.

## 2.4 Classification Layer

The localized features extracted by the GCN and the global features extracted by the FC-NN are concatenated to build a lower-dimensional representation of the input data. The combined extracted features, $[\boldsymbol{\theta_1}, \boldsymbol{\theta_2}]$ as shown in Figure 1 are then connected to a single layer neural network with a softmax activation function to output probability for each class shown as Box C in Figure 1. The class with the highest probability is assigned as the prediction label for the sample.

## 2.5 Loss Function

For the proposed model, the loss function is a linear combination of three different loss functions as

$$L = \lambda_1 L_{entropy} + \lambda_2 L_{reconstruction} + \lambda_3 L_{regularization}, \tag{9}$$

where $\lambda_1$, $\lambda_2$ and $\lambda_3$ are parameters for $L$, $L_{entropy}$ is the cross-entropy loss for classification, $L_{reconstruction}$ is a mean squared error between reconstructed $\tilde{\mathbf{X}}$ and input data $\mathbf{X}$ for the graph autoencoder, and $L_{regularization}$ is the squared $l^2$ norm of the complete model parameter vector to penalize the number of parameters to avoid overfitting. $L_{entropy}$ for each sample is defined as

$$L_{entropy} = -\sum_{i=1}^{c} t_i \log(p_i), \tag{10}$$

where $c$ is the total number of molecular subtype classes, $t_i$ is the true label for the sample, and $p_i$ is the probability of class $i$ from the softmax layer. $L_{reconstruction}$ is defined as

$$L_{reconstruction} = \sum_{i=1}^{M} \sum_{j=1}^{O} (\mathbf{x}_{i,j} - \tilde{\mathbf{x}}_{i,j})^2, \tag{11}$$

where $\mathbf{x}_{i,j}$ is the vector of $j$th omic feature for sample $i$ and $\tilde{\mathbf{x}}_{i,j}$ is the corresponding reconstructed vector. Let's denote $\mathbf{W}$ as the vector consists of all the parameters in the model and the $L_{regularization}$ is defined as

$$L_{regularization} = \sum_{w_i \in \mathbf{W}} w_i^2. \tag{12}$$

# 3 EXPERIMENTAL RESULTS

## 3.1 Datasets & Data Preprocess

We downloaded the TCGA Pan-cancer RNA-seq and CNV data, and the molecular subtype assignments from the Xena website hosted by the University of California Santa Cruz [4]. The batch effect normalized RNA-seq dataset includes the collection of 11,060 samples containing 10,323 cancer samples and 737 normal samples, where each sample has 20,531 gene features. The CNV dataset via GISTIC2 method consists of 10,845 cancer samples with no normal samples. The molecular subtype labels are available for 9,759 cancer samples [4, 20]. We filtered out the samples that do not have the corresponding molecular subtype labels and resulted 9,759 samples with known molecular subtype labels. We also included 17,946 genes that are common in both the gene expression data and the CNV data. The numbers of samples and main characteristics of each of the 28 molecular subtypes are shown in Fig 2, with numbers of cases ranging from 34 to 762, average number of cases is 349, and median number of cases is 306. It can be observed that there is an imbalance among 28 classes and the effect of such imbalance on the model performance will be discussed in detail in section 3.3. 10% of the samples is randomly selected as the test set and all 28 classes are present in the test set.

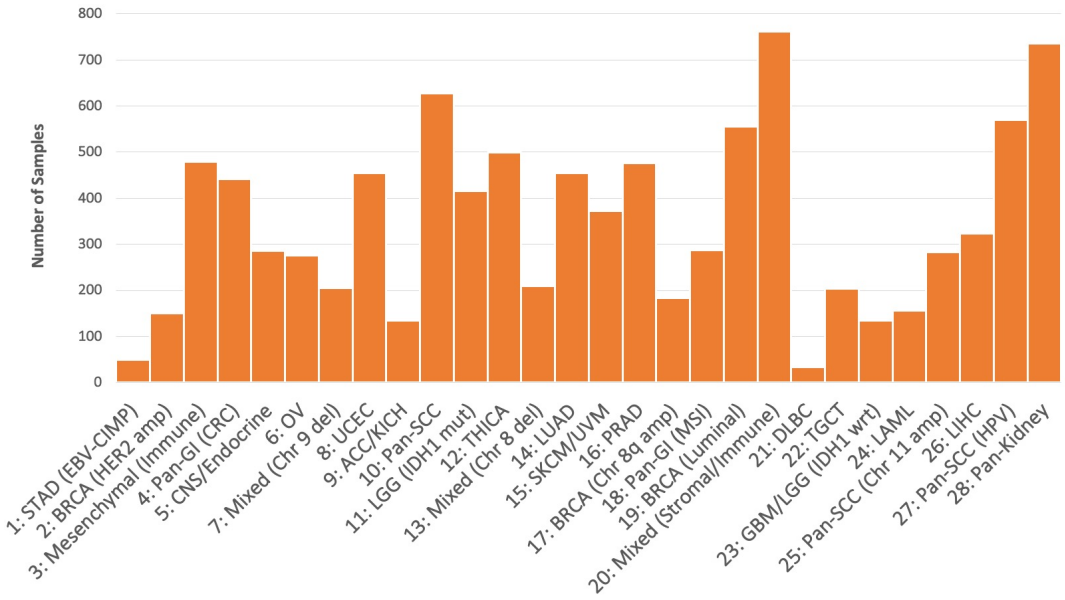


**Figure 2: Number of Samples for 28 Molecular Subtypes out of the 9,759 TCGA Pan-cancer Samples.**

In our experiments, we considered four different levels of the number of genes, $N$: 1,000, 2,000, 5,000, and 7,000. After normalizing both the gene expression data and the CNV data across samples, we selected the top $N$ genes based on their variances in the expression data. To construct the knowledge graphs without singleton, we excluded genes that are not connected to other genes and we selected top $N$ high variance genes with at least one connection to other genes as the input features.

## 3.2 Experimental Setting

The multi-omics data have an input dimension of $N \times 2$ for each sample (gene expression and CNV data of $N$ genes). The GCN consists of one convolutional layer, one pooling layer, and one FC layer with the size of 32 in the encoder part; and in the decoder part it consists of one FC layer with the size of $2N$. For the parallel FC network the two-dimensional multi-omics data are flattened to a vector of size $2N$ as shown in Equation 8. In the parallel FC network, we used two FC layers with the size of 256 and 32 to extract global features, then the localized features and global features are concatenated together for the classification network. For loss function $L$, $\lambda_1$ and $\lambda_2$ are set to 1 and $\lambda_3$ is set to 0.0001. Our model has been tested on GGI with singleton with 100, 200, 500 epochs and found no significant gain in performance beyond 100 epochs. Thus all the models mentioned in the following sections were trained with 100 epochs. We used PyTorch 1.2 package in Python to implement the proposed GCN-based classifier model employing different input knowledge graphs. The source code is available via Github at https://github.com/NabaviLab/GCN-on-Molecular-Subtype.

We compared the performance of our model with those of CNN, FC-NN, RF, SVM, and pure GCN models. The pure GCN model has the same architecture as the GCN section of our proposed model (Box A in Figure 1). We implemented the CNN and FC-NN models with the Keras 2.4 package in Python, and we implemented the RF and SVM models with the scikit-learn 0.24 package in Python. The FC-NN model has 3 FC layers with the size of 512, 256, 128. The CNN model transforms the data of each omic into a two dimensional image-like data as input. It has 3 convolutional layers with $3 \times 3$ filters.

## 3.3 Performance Comparison & Analysis

The prediction accuracies, weighted precisions, recalls and F1 scores of the GCN model using six different knowledge graphs on single-omic data (expression values) are shown in Table 1. The overall best performance is achieved by employing the GGI network without singletons as the input knowledge graph. The GCN model shows the same trend in prediction accuracies using all the six input graphs and the best accuracy is achieved at 5,000 genes. It shows that the use of more genes with less expression variation across samples is not adding more information. There is no significant difference in performance between the models using graphs with and without singleton. Models using the gene co-expression and GGI networks as input graphs generally perform better than those using the PPI network in terms of prediction accuracy. This can be explained by noting that the interactions in the PPI dataset only include genes that are mapped to proteins and have interactions. The non-coding genes are not mapped to proteins and they don't have any interactions in the PPI dataset. Such performance difference was also reported by other researchers using PPI data [15]. There is no significant difference between the overall performance of models employing GGI or co-expression networks as prior knowledge. The weighted precisions, weighted recalls, and weighted F1 scores are relatively consistent with the values of the prediction accuracy, which indicates our model performs well in minimizing the number of both cases of false positive and false negative.

**Table 1: Performance of the Proposed Model Using the Six Knowledge Networks on Single-omic (Gene Expression) Data**

| Metrics | Prediction Accuracy | | | | Weighted Precision | | | | Weighted Recall | | | | Weighed F1 Score | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Input Graphs | 1000 | 2000 | 5000 | 7000 | 1000 | 2000 | 5000 | 7000 | 1000 | 2000 | 5000 | 7000 | 1000 | 2000 | 5000 | 7000 |
| GGI | **80.8%** | 80.4% | **85.0%** | 83.7% | 0.80 | 0.81 | 0.85 | 0.84 | 0.81 | 0.80 | 0.85 | 0.84 | 0.80 | 0.80 | 0.85 | 0.83 |
| GGI+S[1] | 78.3% | 80.7% | 84.1% | 83.7% | 0.78 | 0.81 | 0.84 | 0.84 | 0.78 | 0.81 | 0.84 | 0.84 | 0.78 | 0.80 | 0.84 | 0.83 |
| PPI | 79.1% | 80.9% | 84.1% | 83.6% | 0.79 | 0.81 | 0.84 | 0.83 | 0.79 | 0.81 | 0.84 | 0.84 | 0.78 | 0.80 | 0.84 | 0.83 |
| PPI+S[1] | 78.8% | 80.7% | 84.1% | 84.3% | 0.78 | 0.81 | 0.84 | 0.84 | 0.79 | 0.81 | 0.84 | 0.84 | 0.78 | 0.80 | 0.84 | 0.84 |
| Co-exp | 79.5% | **81.1%** | 84.1% | 83.6% | 0.79 | 0.81 | 0.84 | 0.83 | 0.79 | 0.81 | 0.84 | 0.84 | 0.79 | 0.81 | 0.84 | 0.83 |
| Co-exp+S[1] | 78.8% | 80.8% | 84.1% | **84.6%** | 0.78 | 0.81 | 0.84 | 0.84 | 0.79 | 0.81 | 0.84 | 0.85 | 0.78 | 0.80 | 0.84 | 0.84 |

[1] GGI (PPI/Co-exp) +S represents the input graph with singleton nodes.

**Table 2: Performance of Other Deep Learning, Machine Learning and Conventional Classification Methods on Single-omic Data**

| Metrics | Prediction Accuracy | | | | Weighted Precision | | | | Weighted Recall | | | | Weighted F1 Score | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Method | 1,000 | 2,000 | 5,000 | 7,000 | 1000 | 2000 | 5000 | 7000 | 1000 | 2000 | 5000 | 7000 | 1000 | 2000 | 5000 | 7000 |
| GCN[1] | 79.3% | 80.0% | 81.4% | 78.5% | 0.79 | 0.81 | 0.81 | 0.79 | 0.79 | 0.80 | 0.81 | 0.78 | 0.79 | 0.80 | 0.81 | 0.78 |
| CNN | 77.8% | 74.5% | 80.4% | 79% | 0.75 | 0.74 | 0.80 | 0.79 | 0.75 | 0.74 | 0.79 | 0.78 | 0.76 | 0.75 | 0.79 | 0.79 |
| FC-NN | 78.6% | 79.1% | 82.3% | 81.5% | 0.77 | 0.78 | 0.81 | 0.80 | 0.78 | 0.79 | 0.81 | 0.81 | 0.77 | 0.79 | 0.81 | 0.81 |
| RF | 74.04% | 74.5% | 78.7% | 77.7% | 0.68 | 0.67 | 0.73 | 0.69 | 0.72 | 0.73 | 0.77 | 0.76 | 0.67 | 0.68 | 0.73 | 0.71 |
| SVM | 74.3% | 78.6% | 81.5% | 80.7% | 0.69 | 0.72 | 0.74 | 0.74 | 0.71 | 0.77 | 0.78 | 0.77 | 0.69 | 0.73 | 0.75 | 0.74 |

[1] This model has the same architecture as the network in the GCN section of our proposed model.

We accessed the performance of five baseline models, CNN, FC-NN, RF, SVM, and pure GCN, for comparison. Their performances are shown in Table 2. Our proposed model without the parallel network (pure GCN) is trained with GGI network without singleton as GGI network performs the best for our proposed model. It can be observed that the prediction accuracy of the proposed GCN-based model, using any of the prior gene interaction networks, is higher than those of the baseline models by 2% to 8% at all four settings for the number of input genes $N$. The pure GCN baseline model performs the best for 1,000 and 2,000 genes and FC-NN baseline model performs the best for 3,000 and 4,000 genes. The weighted precisions, weighted recalls, and weighted F1 scores of all four baseline models are lower than our models at all four levels of the number of genes. All baseline models' F1 scores are lower than their prediction accuracies (Table 2), but the proposed model achieves an F1 score close to the prediction accuracy. This indicates that there are a higher number of cases of false positives and false negatives for the baseline models compared to the proposed model at the same level of prediction accuracy. Our proposed model outperforms the pure GCN model in the most cases, which demonstrates the effectiveness of the parallel network in our model.

We also examined miss-classification across all molecular subtype classes. To evaluate the miss-classification for each individual class, we computed the row-standardized confusion matrix of the best performing GCN-based model (using the GGI network with singletons at 5000 genes) as shown in Table 3. The confusion matrix is standardized by the row sum, the size of each class in test set. The top six most miss-classified molecular subtype classes in red boxes are Class 7: Mixed (Chr 9 del), Class 13: Mixed (Chr 8 del), Class 17: BRCA Chr 8q amp, Class 2 (BRCA HER2 amp), Class 25: Pan-SCC (Chr 11 amp) and Class 1: STAD(EBV-CIMP). Three of these classes are mixed cancer classes, which contain more than one traditional tissue of origin cancer type, and the rest two are subtypes of BRCA. The result shows, as expected, the mixed cancer classes can be hard to predict with only single-omic data.

As mentioned previously, the numbers of cases are imbalanced across 28 molecular subtype classes. The numbers of cases of the six most miss-classified classes range from 49 to 283, which all fall into the smaller sample size half of the 28 classes, but not all among the lowest sample size classes. Since the genomic data are high-dimension and the lower bound of the numbers of cases is only 34, either sub-sampling or over-sampling isn't a viable option to address the problem of imbalance data.

Using only mRNA expression data may not be sufficient for molecular subtype classification, as the TCGA Research Networks' study in 2018 also integrated the data from 5 genome-wide platforms to assign the subtypes [6]. This has motivated us to develop a model to integrate multi-omics data.

The performance of the proposed GCN-based model using the six input knowledge graphs on multi-omic data (expression values and CNV data) is given in Table 3. As can be seen, using multi-omics data improves the performance of the proposed model compared to using single-omic data. Similar to the model using the single-omic data, the different input knowledge graphs do not affect the classification accuracies significantly. Also, the best prediction accuracy is achieved for the data with 5,000 genes. However, the performance on multi-omics data with only 2,000 genes is very close to that on single-omic data with 5,000 genes, which demonstrates that the multi-omics data provide additional information compared to single-omic data at a fixed number of genes.

To evaluate the performance of the proposed multi-omics GCN-based model, we used three baseline models, pure GCN, CNN and FC-NN models, for comparison and their performances are shown in Table 4. The RF and SVM models are not scaleable and perform

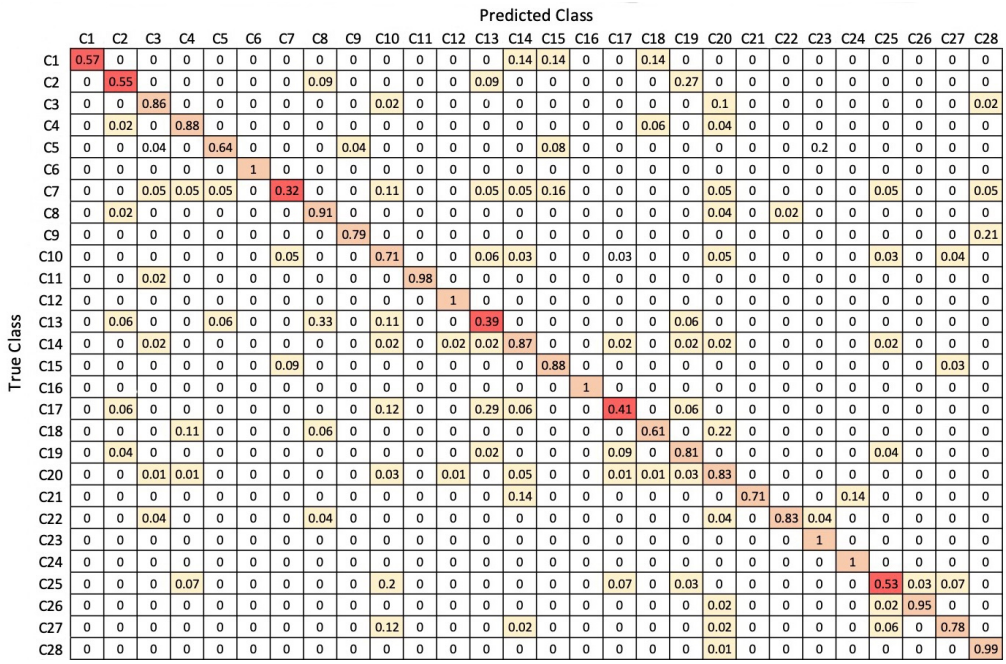


**Figure 3: The Row-standardized Confusion Matrix of the Predicted Labels by the GGI Model with Singletons and N=5000 on Single-omic. Red Boxes are used to show the data for the top six most miss-classified classes.**

**Table 3: Performance of the Proposed Model Using the Six Knowledge Networks on Multi-omics Data (Gene Expression and CNV)**

| Metrics | Prediction Accuracy | | | | Weighted Precision | | | | Weighted Recall | | | | Weighted F1 Score | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Input Graphs | 1000 | 2000 | 5000 | 7000 | 1000 | 2000 | 5000 | 7000 | 1000 | 2000 | 5000 | 7000 | 1000 | 2000 | 5000 | 7000 |
| GGI | 84.0% | **86.4%** | 86.6% | 84.6% | 0.84 | 0.86 | 0.87 | 0.85 | 0.84 | 0.86 | 0.87 | 0.84 | 0.84 | 0.86 | 0.86 | 0.85 |
| GGI+S[1] | **85.7%** | 85.1% | 86.3% | 85.1% | 0.86 | 0.85 | 0.87 | 0.86 | 0.85 | 0.85 | 0.86 | 0.86 | 0.85 | 0.85 | 0.87 | 0.86 |
| PPI | 84.4% | 85.6% | 86.5% | 85.3% | 0.84 | 0.86 | 0.87 | 0.85 | 0.84 | 0.86 | 0.86 | 0.85 | 0.84 | 0.85 | 0.86 | 0.85 |
| PPI+S[1] | 85.1% | 84.9% | 86.6% | 85.3% | 0.85 | 0.85 | 0.87 | 0.85 | 0.85 | 0.85 | 0.87 | 0.85 | 0.84 | 0.84 | 0.86 | 0.85 |
| Co-exp | 85.0% | 84.6% | 86.7% | 85.7% | 0.85 | 0.86 | 0.87 | 0.86 | 0.85 | 0.86 | 0.87 | 0.85 | 0.85 | 0.86 | 0.87 | 0.85 |
| Co-exp+S[1] | 85.1% | 85.8% | **86.8%** | **85.9%** | 0.85 | 0.86 | 0.87 | 0.86 | 0.85 | 0.86 | 0.87 | 0.86 | 0.85 | 0.86 | 0.87 | 0.86 |

[1] GGI (PPI/Co-exp) +S represents the input graph with singleton nodes.

**Table 4: Performance of Other Deep Learning Classification Methods on Multi-omics Data**

| Metrics | Prediction Accuracy | | | | Weighted Precision | | | | Weighted Recall | | | | Weighted F1 Score | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Method | 1000 | 2000 | 5000 | 7000 | 1000 | 2000 | 5000 | 7000 | 1000 | 2000 | 5000 | 7000 | 1000 | 2000 | 5000 | 7000 |
| GCN[1] | 81.7% | 83.2% | 83.4% | 84.0% | 0.82 | 0.84 | 0.84 | 0.84 | 0.82 | 0.83 | 0.83 | 0.84 | 0.82 | 0.83 | 0.83 | 0.84 |
| CNN | 79.3% | 78.9% | 80.5% | 81.2% | 0.79 | 0.77 | 0.79 | 0.80 | 0.78 | 0.78 | 0.79 | 0.81 | 0.78 | 0.77 | 0.79 | 0.80 |
| FC-NN | 81.6% | 82.7% | 81.9% | 83.6% | 0.80 | 0.80 | 0.80 | 0.81 | 0.79 | 0.80 | 0.79 | 0.80 | 0.78 | 0.79 | 0.79 | 0.81 |

[1] This model has the same architecture as the network in the GCN section of our proposed model.

poorly by adding input features (not included in the table). The pure GCN model was also trained with the GGI network without singleton. It can be observed that the proposed model outperforms the baseline models by 2% to 5% in terms of classification accuracy. Despite being the best performing model among all three baseline models, the pure GCN model still performs poorly in terms of all four metrics compared to the proposed model, which proves the effectiveness of the parallel network with the multi-omics data. The weighted precisions, weighted recalls, and weighted F1 scores of the baseline models are still lower than the proposed model at all four levels of the number of genes. The performance of both baseline models is also improved by using multi-omics data, but with a lower rate compared to the proposed model. The performance improvement by using multi-omics data is higher when using less

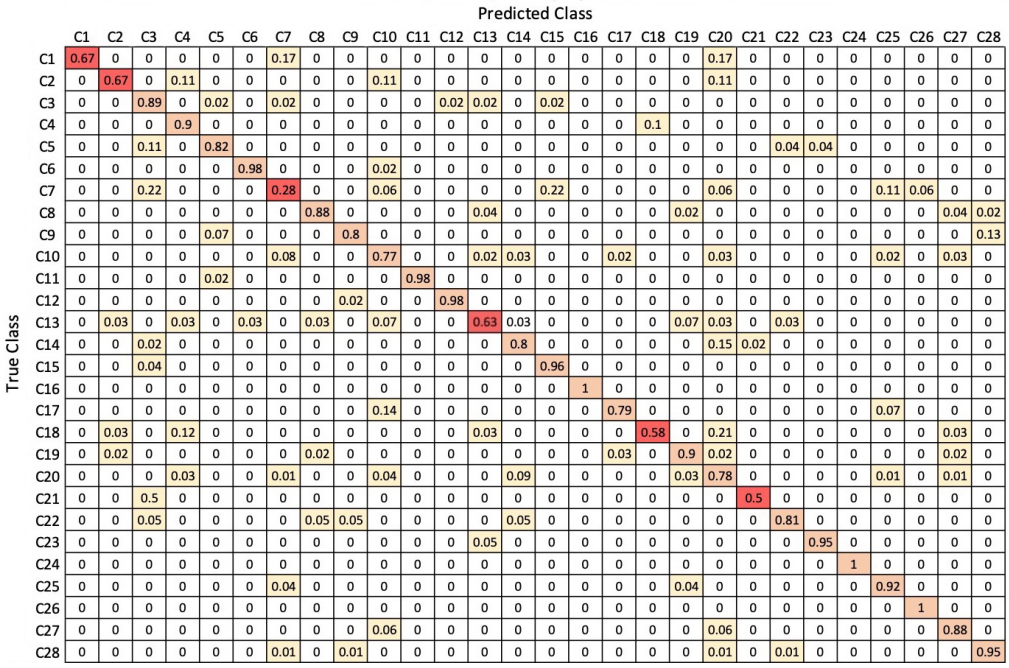


**Figure 4: The Row-standardized Confusion Matrix of the Predicted Labels by the Co-expression Model with Singletons and N=5000 on Multi-omic. Red Boxes are used to show the data for the top six most miss-classified classes.**

number of genes. It shows the benefit of using more genomic data types in extracting effective information.

To evaluate the miss-classification for each individual class using multi-omics data, we computed the row-standardized confusion matrix of the best performing GCN-based model on multi-omics data (using gene co-expression network with singletons at 5000 genes) as shown in Table 4. The top six most miss-classified molecular subtype classes (including two tied classes) in red boxed are Class 7: Mixed (Chr 9 del), Class 21: DLBC, Class 18: Pan-GI (MSI), Class 13: Mixed (Chr 8 del), Class 1: STAD(EBV-CIMP) (tied), ,and Class 2: BRCA(HER2 amp) (tied). Three of those classes are mixed cancer classes and the prediction accuracies on most miss-classified classes are significantly improved compared to the model with single-omic data. Thus, we can conclude that using multi-omics data helps to improve the model performance on poorly classified classes from the prediction of the model using single-omic data. The numbers of cases of the six most miss-classified classes range from 34 to 288, which also all fall into the smaller sample size half of the 28 classes (but not all among the lowest sample size classes). With the improved overall performance at these most miss-classified classes, we can conclude that the imbalance problem in the data can be mitigated by introducing other omic features for the model.

## 4 CONCLUSIONS

In this study, we proposed a novel end-to-end deep-learning method for the molecular subtype classification that uses multi-omics data and incorporates prior biological data. To the best of our knowledge, this is the first study of such classification model on the new type of cancer taxonomy. We designed the model's structure such that it combines the localized features extracted by a graph autoencoder and general features extracted by a parallel shallow FC network to provide a better prediction. The proposed model incorporates prior biological knowledge on interactions among genes in a form of GGI, PPI, or gene co-expression network into the graph autoencoder model as the input graph, and integrates multiple omics data as the attributes of the nodes in the graph autoencoder model.

Comparing the performance of our proposed model with those of the baseline models, CNN, FC-NN, RF and SVM models, we demonstrated that the GCN part of the proposed model can extract additional features from prior gene interaction knowledge resulting in a better performance. The results demonstrate that the proposed model achieves not only a better prediction accuracy, but also a better F1 score compared to baseline models. This is specially important due to the significant class imbalance in the data and the deteriorating consequence of false negatives.

In addition, comparing the performance of the proposed GCN-based model using multi-omics data with those of using single-omic data, we showed that the use of multi-omics can significantly improve the performance of classification. The proposed model is scalable in integrating more omics data since omics data are represented as node attributes (features) in the proposed GCN model.

We observed that the proposed model preforms relatively similar using any of the GGI, PPI and gene co-expression networks as the input knowledge graph. However, using PPI results in slightly poorer performance. That can be because the PPI dataset only includes protein coding genes. It indicates that the completeness of the biological prior knowledge is very important for extracting

comprehensive effective features. To better incorporate incomplete biological prior knowledge is also another focus of our future research. We also observed that for the best classification we can use a small subset of genes with high variance across the samples. The proposed model achieved the best prediction performance with 5,000 genes for both single-omic and multi-omics data regardless of the input knowledge graphs. It shows employing more genes with less expression variation across samples doesn't contribute to extracting more information.

The proposed model also has some limitations, mostly due to the availability of the public genomic data and the focus of the study. By the nature of biomedical data, having an imbalanced dataset is very common. In our study, the data were not specifically processed for the imbalance problem. However, our model shows robust and effective performance in dealing with imbalanced data, especially when adding more omics data. Also, in this study we focused mostly on the functional DNA elements. To add more features for the functional elements, genes, we introduced CNV data for genes into the model, and we ignored the CNV data for the non-coding regions. Better handling the imbalanced data and better utilizing CNV data will be the future research direction of our team. We will also work on including more omics data such as somatic point mutations, microRNA expression, and methylation as part of our future work.

In summary, incorporating prior biological knowledge and integrating multi-omics data improve molecular subtype classification, and GCN methodologies can be used as a means to extract localized features from biological networks and to efficiently integrate several types of genomic data as node attributes.

## ACKNOWLEDGMENTS

This study was supported by the National Science Foundation (NSF) under grant No. 1942303, PI: Nabavi; and Bingjun Li's CIGNA Graduate Fellowship from the Computer Science and Engineering department at the University of Connecticut.

## REFERENCES

[1] Zhong Chen, Andrea Edwards, and Kun Zhang. 2020. Fusion Lasso and Its Applications to Cancer Subtype and Stage Prediction. In Proceedings of the 11th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics. 1–8.

[2] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. arXiv preprint arXiv:1606.09375 (2016).

[3] Centers for Disease Control, Prevention, Centers for Disease Control, and Prevention. 2009. Leading causes of death, 1900-1998. National Center for Health Statistics Web site. www.cdc.gov/nchs/data/dvs/lead1900_98.pdf. Accessed March 5 (2009).

[4] Mary J Goldman, Brian Craft, Mim Hastie, Kristupas Repečka, Fran McDade, Akhil Kamath, Ayan Banerjee, Yunhai Luo, Dave Rogers, Angela N Brooks, et al. 2020. Visualizing and interpreting cancer genomics data via the Xena platform. Nature biotechnology 38, 6 (2020), 675–678.

[5] David K. Hammond, Pierre Vandergheynst, and Rémi Gribonval. 2011. Wavelets on graphs via spectral graph theory. Applied and Computational Harmonic Analysis 30, 2 (2011), 129–150. https://doi.org/10.1016/j.acha.2010.04.005

[6] Katherine A Hoadley, Christina Yau, Toshinori Hinoue, Denise M Wolf, Alexander J Lazar, Esther Drill, Ronglai Shen, Alison M Taylor, Andrew D Cherniack, Vésteinn Thorsson, et al. 2018. Cell-of-origin patterns dominate the molecular classification of 10,000 tumors from 33 types of cancer. Cell 173, 2 (2018), 291–304.

[7] Katherine A. Hoadley, Christina Yau, Denise M. Wolf, Andrew D. Cherniack, David Tamborero, Sam Ng, Max D.M. Leiserson, Beifang Niu, Michael D. McLellan, Vladislav Uzunangelov, Jiashan Zhang, Cyriac Kandoth, Rehan Akbani, Hui Shen, Larsson Omberg, Andy Chu, Adam A. Margolin, Laura J. van't Veer, Nuria Lopez-Bigas, Peter W. Laird, Benjamin J. Raphael, Li Ding, A. Gordon Robertson, Lauren A. Byers, Gordon B. Mills, John N. Weinstein, Carter Van Waes, Zhong Chen, Eric A. Collisson, Christopher C. Benz, Charles M. Perou, and Joshua M. Stuart. 2014. Multiplatform Analysis of 12 Cancer Types Reveals Molecular Classification within and across Tissues of Origin. Cell 158, 4 (2014), 929–944. https://doi.org/10.1016/j.cell.2014.06.049

[8] Javaid Iqbal, Ophira Ginsburg, Paula A. Rochon, Ping Sun, and Steven A. Narod. 2015. Differences in Breast Cancer Stage at Diagnosis and Cancer-Specific Survival by Race and Ethnicity in the United States. JAMA 313, 2 (01 2015), 165–173. https://doi.org/10.1001/jama.2014.17322 arXiv:https://jamanetwork.com/journals/jama/articlepdf/2089353/joi140172.pdf

[9] K. D. Kochanek, J. Xu, and E. Arias. 2020. Mortality in the United States, 2019. NCHS Data Brief 395 (Dec 2020), 1–8.

[10] Konstantina Kourou, Themis P Exarchos, Konstantinos P Exarchos, Michalis V Karamouzis, and Dimitrios I Fotiadis. 2015. Machine learning applications in cancer prognosis and prediction. Computational and structural biotechnology journal 13 (2015), 8–17.

[11] Yuanyuan Li, Kai Kang, Juno M Krahn, Nicole Croutwater, Kevin Lee, David M Umbach, and Leping Li. 2017. A comprehensive genomic pan-cancer classification using The Cancer Genome Atlas gene expression data. BMC genomics 18, 1 (2017), 1–13.

[12] Boyu Lyu and Anamul Haque. 2018. Deep learning based tumor type classification using gene expression data. In Proceedings of the 2018 ACM international conference on bioinformatics, computational biology, and health informatics. 89–96.

[13] Sean McPhail, Sam Johnson, David Greenberg, Mick Peake, and Brian Rous. 2015. Stage at diagnosis and early mortality from cancer in England. British journal of cancer 112, 1 (2015), S108–S115.

[14] R. Oughtred, J. Rust, C. Chang, B. J. Breitkreutz, C. Stark, A. Willems, L. Boucher, G. Leung, N. Kolas, F. Zhang, S. Dolma, J. Coulombe-Huntington, A. Chatr-Aryamontri, K. Dolinski, and M. Tyers. 2021. The BioGRID database: A comprehensive biomedical resource of curated protein, genetic, and chemical interactions. Protein Sci 30, 1 (01 2021), 187–200.

[15] Ricardo Ramirez, Yu-Chiao Chiu, Allen Hererra, Milad Mostavi, Joshua Ramirez, Yidong Chen, Yufei Huang, and Yu-Fang Jin. 2020. Classification of Cancer Types Using Graph Convolutional Neural Networks. Frontiers in physics 8 (2020).

[16] Rebecca L. Siegel, Kimberly D. Miller, Hannah E. Fuchs, and Ahmedin Jemal. 2021. Cancer Statistics, 2021. CA: A Cancer Journal for Clinicians 71, 1 (2021), 7–33. https://doi.org/10.3322/caac.21654 arXiv:https://acsjournals.onlinelibrary.wiley.com/doi/pdf/10.3322/caac.21654

[17] Damian Szklarczyk, Annika L Gable, David Lyon, Alexander Junge, Stefan Wyder, Jaime Huerta-Cepas, Milan Simonovic, Nadezhda T Doncheva, John H Morris, Peer Bork, et al. 2019. STRING v11: protein–protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets. Nucleic acids research 47, D1 (2019), D607–D613.

[18] Christian Von Mering, Lars J Jensen, Berend Snel, Sean D Hooper, Markus Krupp, Mathilde Foglierini, Nelly Jouffre, Martijn A Huynen, and Peer Bork. 2005. STRING: known and predicted protein–protein associations, integrated and transferred across organisms. Nucleic acids research 33, suppl_1 (2005), D433–D437.

[19] Tianyu Wang, Jun Bai, and Sheida Nabavi. 2021. Single-Cell Classification Using Graph Convolutional Networks. bioRxiv (2021). https://doi.org/10.1101/2021.06.13.448259 arXiv:https://www.biorxiv.org/content/early/2021/06/14/2021.06.13.448259.full.pdf

[20] John N Weinstein, Eric A Collisson, Gordon B Mills, Kenna R Mills Shaw, Brad A Ozenberger, Kyle Ellrott, Ilya Shmulevich, Chris Sander, and Joshua M Stuart. 2013. The cancer genome atlas pan-cancer analysis project. Nature genetics 45, 10 (2013), 1113–1120.

[21] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. 2020. A comprehensive survey on graph neural networks. IEEE transactions on neural networks and learning systems (2020).

[22] Xiaoyu Zhang, Jingqing Zhang, Kai Sun, Xian Yang, Chengliang Dai, and Yike Guo. 2019. Integrated multi-omics analysis using variational autoencoders: application to pan-cancer classification. In 2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). IEEE, 765–769.