# GRAPH REPRESENTATION LEARNING FOR HETEROGENEOUS MULTIMODAL BIOMEDICAL DATA

by

NHAT CHAU TRAN

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON

December 2022

To my mother Yen and my father Long

who set the example and who made me who I am.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank God, without whom I would be nothing. This dissertation is the culmination of seven years of learning, perseverance, failures, prayers, and achievements. However, this work would not have been impossible without the people who have guided, supported, and contributed to my doctoral pursuit.

I am proud to have been mentored by Dr. Jean Gao, who is a fantastic professor, researcher, guide, and advisor to have by my side. Her relentless enthusiasm for significant original research inspired me to grow as an independent researcher. I also wish to thank Dr. Manfred Huber, Dr. Junzhou Huang, and Dr. Dajiang Zhu for their service to my dissertation committee, contributions of ideas, and interest in my research.

I thank all the high school teachers and the Computer Science and Engineering professors who taught me during my early education. I am incredibly grateful for the financial support from the US Department of Education and UT Arlington that has made my academic endeavors possible. I thank Leah, Michael, and Ashley from the UTA Writing Lab for their encouragement and practical advances in writing and completing this dissertation.

Finally, I would like to express my deep gratitude to my wife for her constant love and support for my graduate studies. She has been there through everything, and without her faith and emotional and mental support, I could not have done it. I am eternally grateful to my parents and sister for their teachings, patience, endless support, and prayers. I thank my extended family for their encouragement

and inspiring me to pursue biomedical studies. Lastly, I thank my B-FAM breaking family, swing dancing friends, and Brazilian jiu-jitsu teammates who have supported me and motivated me to stick it out through my Ph.D. journey.

December 09, 2022

# ABSTRACT

## GRAPH REPRESENTATION LEARNING FOR HETEROGENEOUS MULTIMODAL BIOMEDICAL DATA

Nhat Chau Tran, Ph.D.

The University of Texas at Arlington, 2022

Supervising Professor: Jean X. Gao

The emergence of high-throughput sequencing technology has generated a wealth of "multi-omics" data, capturing information about different types of biomolecules at multiple levels. Since large-scale genomics, transcriptomics, and proteomics data are becoming publicly available, integrated systems analysis utilizing these data sources has taken the front seat in deriving valuable insights for identifying cancer biomarkers or predicting interactions and functions for novel molecules such as LncRNAs. The graph representation learning paradigm can address these challenging tasks as among the most promising approaches to improve predictions over sparsely annotated molecular entities and to provide representation capacity and interpretability over heterogeneous and hierarchically structured data. This dissertation investigates novel graph machine learning approaches for biomarker discovery in microRNA co-expression graphs, functional representation of LncRNA sequences for link prediction, aggregation of heterogeneous relations to predict protein functions, and the pipelines to enable reproducible graph integration of public biological databases.

Prior works on multi-omics integrative analysis have had significant shortcomings in addressing the challenges due to the heterogeneity and scale of graph-based datasets. For instance, univariate analyses cannot produce robust results when identifying biomarkers for genetically heterogeneous cancer diseases with multi-omics data without considering the interconnectivity between the various omics. We constructed the MicroRNA Dysregulatory Synergistic Network to extract features from aberrant MicroRNA-MessengerRNA interactions and applied a multivariate technique that considers the grouping effect of biomarkers. Aside from inferring gene-disease associations, we also proposed the rna2rna method to predict the regulatory interactions and the functional similarities of non-coding RNAs (ncRNAs) where there are non-existent annotations for novel sequences. By leveraging the diverse array of interaction, sequence, annotation, and expression multimodal data, our method can characterize the functional similarity and interaction topologies of a novel ncRNA from sequence. Then, we formulated a generalized algorithm named LATTE to deal with the complexity of heterogeneous networks, where multiple node types are connected in various ways. This graph neural network method is applied to the automatic protein function prediction problem in an architecture called LATTE2GO that aims to aggregate information from higher-order relations to extract integrated representations of protein-protein networks and the hierarchical Gene Ontology. Finally, as data integration and feature engineering are vital steps in large-scale bioinformatics projects, we developed an open-source software called OpenOmics. Our tool assists in systematically integrating heterogeneous multi-omics datasets and interfacing with popular public annotation and interaction databases for increased reproducibility and standardization of biomedical data integration. The performance evaluation of our proposed methods, algorithms, and tools validates the utility and effectiveness compared to existing state-of-the-art methods.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

# CHAPTER 1

# INTRODUCTION

## 1.1 Background and Preliminaries

Regulatory long non-coding RNAs (lncRNAs) and microRNAs (miRNAs) that influence gene expression post-transcriptionally by interacting with target messenger RNAs (mRNA) form a complex network of transcriptomic interactions. These heterogeneous families of non-coding RNAs (ncRNAs) are associated with nearly all cellular processes, including cell division, senescence, differentiation, stress response, immune activation, and apoptosis [49, 45, 20, 82]. Studying the interconnectivity of these biomolecules has enabled researchers to understand the intricacies of regulatory mechanisms where protein-coding RNAs alone do not offer the complete picture.

MicroRNAs are approximately 22nt long and post-transcriptionally target messenger-RNAs (mRNAs) to regulate the translation of target genes. Recently, it has been found that microRNAs have the potential as both biomarkers and therapeutic targets for lung cancer [70, 138]. On the other hand, lncRNAs are also gaining considerable attention as the largest and most diverse non-coding RNA class, encompassing nearly 30,000 discovered transcripts in humans. They are classified as $>$ 200nt transcribed RNA molecules, which has a diverse influence upon the function of other ncRNAs and regulation of protein-coding RNAs. Among many of their known functional interaction mechanisms, lncRNAs are known to act as miRNA decoys, derepress gene expression by competing with miRNAs for shared mRNA targets, or directly regulate gene expression [149]. Additional studies have indicated that miRNAs can regulate

lncRNAs by triggering decay [88], and some processed lncRNAs can even generate miRNAs [39].

These types of functional interactions between lncRNAs and miRNAs to co-regulate gene expressions highlight the complexity of the non-coding RNA regulatory network. Although the miRNA's regulatory mechanisms and a few lncRNAs have been studied, the discovery of the functional roles for a large number of ncRNAs in the human transcriptome is still at a preliminary stage. Determining the function of individual ncRNAs remains challenging as most of these RNA transcripts are currently unannotated, and their known interactions are sparse. Recent advances in RNA sequencing (RNA-Seq), deep sequencing (CLIP-seq, LIGR-Seq), and other high-throughput methods have allowed for an unprecedented analysis of such transcripts and enabled researchers to generate large-scale interaction and annotation databases. However, the interaction networks generated from such experimental data are often scant and incomplete in the number of ncRNAs covered. For instance, although many long non-coding RNAs (lncRNAs) have been identified, only a few hundred have had functional and molecular mechanisms determined to date, as observed in annotation databases such as lncRNAdb [6]. Thus, *in silico* prediction of RNA-RNA interactions has been widely applied in the task of predicting or inferring missing functional interactions, where experimental studies are in short supply due to time and cost.

## 1.2 Motivation and Challenges

As the vast array of ncRNAs datasets are becoming available in the public repositories, several computational challenges are being imposed on bioinformaticians. To infer functions of novel ncRNAs, many graph-theoretic methods have been applied to biological networks with the intuition that RNAs close together in the interaction topology are more likely to be involved in many of the same functions [33]. Typically,

the approach is mining the neighborhood structure of nodes in the network topology to suggest that two nodes are likely to be functionally similar if they share many of the same co-interacting neighbors. Additionally, results [78] have shown that genomic and functional annotation information can facilitate the process of suggesting the interactions of the presently unknown RNAs. Other results have shown that integrating multi-omics data provides information on heterogeneous biomolecules from different layers, rather than considering each biological feature independently, which seems promising to understand complex biology systematically and holistically [145].

Most prior works on multi-omics analysis have had significant shortcomings in addressing the challenges due to the heterogeneity and scale of integrated datasets. The reasons for this include: 1) the RNA-RNA interaction network can be highly sparse; 2) there is a lack of consideration for the directionality of RNA-RNA interactions; 3) a lack of an integrative approach for sequence and annotation data; and 4) the predictions are transductive, i.e., constrained among only nodes with a connection to existing nodes in the training set. These shortcomings can often affect the model's capacity to model heterogeneous relationships, limit the capacity to capture multi-modal representations or hinder its generalizability for inductive predictions.

To address these challenges, this dissertation explores the application of recent advancements in machine learning called "network embedding." I aim to study various approaches to learn from transcriptome-wide RNAs' interaction topology and attributes to predict RNA-RNA functional interactions accurately. Mathematically, our ground truth knowledge about RNA interactions can be represented by a directed adjacency matrix, whose rows and columns correspond to individual lncRNAs, miRNAs, and mRNAs. This matrix's binary entries indicate whether an RNA was observed to have a functional interaction with another RNA, supported by experimentally-validated interaction databases. The matrix is exceptionally sparse, especially among

lncRNAs, i.e., out of millions of possible interactions, only a few thousand have been identified. A significant fraction of newly discovered ncRNAs lacks any identified interactions or functional annotations besides its basic genomic attributes such as locus biotype, and primary transcript sequence [37]. These genes might support essential biological cell functions and potentially serve as targets for genomic, diagnostic, or therapeutic studies. Thus, to functionally characterize these "hypothetical" ncRNAs, the essential tasks are integrating the various multi-modal attributes and the representation of the multi-omics interactions.

## 1.3 Dissertation Organization

In Chapter 2, I proposed a pipeline to analyze the deviation in miRNA-mRNA interactions between various lung cancer subtypes to assess their potential as a predictor of this heterogeneous disease. Integrating the MicroRNA and MessengerRNA transcriptomics data have the potential to pinpoint biomarkers for the development of novel prognostic and therapeutic targets in lung cancer diseases. However, most prior approaches relied only on univariate differential analyses, examining individual RNAs for the significant deviation between normal and tumor samples. Instead, our method integrated miRNA and mRNA expression profiles, extracted features from miRNA regulatory interactions, and constructed a network of functional similarities to identify miRNA synergistic modules. The predicted synergistic microRNA modules lead to a more relevant selection of microRNA biomarkers and considerably improved early-stage lung cancers' prediction accuracy. Our method's overall result demonstrated that considering the interaction pattern between microRNAs and their targets led to a more robust selection of miRNA biomarkers for tumors of all subtypes rather than considering each class of transcriptomics individually.

In Chapter 3, I proposed a novel deep learning framework, rna2rna, which extracts features from RNA sequences to produce a low-dimensional embedding that preserves proximities in the interaction topology and the functional affinity topology. In this proposed embedding space, the two-part "source and target context" captures the receptive fields of each RNA transcript to encapsulate heterogeneous cross-talk interactions between lncRNAs and microRNAs. The proximity between RNAs in this embedding space also uncovers the second-order relationships that allow for accurate inference of novel directed interactions or functional similarities between RNA sequences. Our method performs better in a prospective evaluation than state-of-art approaches at predicting missing interactions from several RNA-RNA interaction databases. Additional results suggest that our proposed framework can capture a manifold for heterogeneous RNA sequences to discover novel functional annotations.

In Chapter 4, I developed a Python library named OpenOmics for integrating heterogeneous multi-omics data and interfacing it with popular public annotation databases, e.g., GENCODE, Ensembl, BioGRID. The library is designed to be highly flexible to allow the user to parameterize the construction of integrated datasets, interactive to assist complex data exploratory analyses, and scalable to facilitate working with large datasets on standard machines. OpenOmics can also facilitate network-based and graph-theoretic analyses of DNA, RNA, and protein interactions in a high-throughput manner. Along with the wide-ranging use cases of OpenOmics, modern software practices were implemented to maximize the integrated framework's usability and reproducibility.

In Chapter 5, I explored an architecture–Layer-stacked ATTention Embedding (LATTE)–that automatically decomposes higher-order meta relations at each layer to extract the relevant heterogeneous neighborhood structures for each node. Additionally, by successively stacking layer representations, the learned node embedding

offers a more interpretable aggregation scheme for nodes of different types at different neighborhood ranges. I conducted experiments on several benchmark heterogeneous network datasets. In both transductive and inductive node classification tasks, LATTE can achieve state-of-the-art performance compared to existing approaches while offering a lightweight model. With extensive experimental analyses and visualizations, the framework can demonstrate the ability to extract informative insights on heterogeneous networks.

In Chapter 6, I extended the proposed LATTE model to the problem of automatic protein function prediction. As current graph-based methods aim to learn protein representation only by considering homogeneous protein-protein interaction (PPI) networks, more information can be encoded by specifying the semantics of specific types of protein-protein association. Thus, using the OpenOmics framework, relationships among genes, transcripts, and proteins can be integrated with Gene Ontology hierarchical structure as a heterogeneous graph. With this data structure, Layer-stacked ATTEntion for protein-function predictions on Gene Ontology (LATTE2GO) was developed to aggregate information among multiple relations to learn representation for proteins and GO terms within the same graph neural network. In experiments on the standardized CAFA benchmark, LATTE2GO achieved a significant performance boost compared to methods that do not consider multi-relational PPI or higher-order relations.

# CHAPTER 2

# MICRORNA DYSREGULATIONAL SYNERGISTIC NETWORK: DISCOVERING MICRORNA DYSREGULATORY MODULES ACROSS SUBTYPES IN NON-SMALL CELL LUNG CANCER

## 2.1 Introduction

Lung cancer accounts for more than 1.5 million deaths globally per year and is the leading cause of cancer-related mortality. About 87% of the lung cancer cases are classified as Non-Small Cell Lung Cancer, and the 5-year survival rate of all stages is below 17% because the majority of lung cancer patients (57%) are diagnosed at later stages since the early stage is typically asymptomatic [38]. Even when diagnosed early, the only recommended treatment is surgical resection, despite that up to 30% of those successfully treated will still die within five years of initial diagnosis [24]. Therefore, the development of early diagnosis and treatment strategy is critical and essential for the control of this deadly disease. Recently, it has been found that microRNAs have the potential as both biomarkers and therapeutic targets for lung cancer [70, 138].

MicroRNAs (miRNAs) are a recently discovered class of small noncoding RNA. Approximately 22nt, miRNAs post-transcriptionally target messenger-RNAs (mRNAs) to regulate the translation of target genes. They have been found to play a critical role in various biological functions such as proliferation, differentiation, and apoptosis [20]. Thus, abnormal miRNA regulatory events can cause a significant impact on various cellular functions, ultimately resulting in complex events leading to

cancer. Increasing evidence suggests that miRNAs can have a causal role in tumorigenesis [43].

Due to the significant role of miRNAs found in cancer biology, many existing lung cancer studies use miRNA expression profiles for accurate prediction of lung cancer stages or subtypes [11, 112]. In a typical differential expression analysis, a univariate statistical method (e.g., student's $t$-test, false discovery rate threshold) is performed to select miRNAs with a significant deviation between normal and tumor sample groups. However, the results are not always satisfactory, as large-scale multi-omics analysis of non-small cell lung adenocarcinoma (LUAD) revealed distinct interactions of miRNA to target mRNA that are specific to histological subtypes [98]. In other words, an identified miRNA biomarker may correctly classify tumor based on analyses done on one particular subtype but may misclassify cases of other subtypes, where it may target a different set of mRNAs. Therefore, for a more robust selection of miRNA biomarker, analysis of the deviation in miRNA-target interactions between various lung cancer subtypes should be considered to assess their potential as predictor to this heterogeneous disease.

Experimental evidence has shown that multiple miRNAs can potentially target a gene through synergism, in which two or more miRNAs can cooperatively co-regulate an individual gene [141]. Studying the synergism of miRNAs within a specific cellular environment is another critical step to determine their disease-specific functions at the system level. Construction of the miRNA co-regulation network by considering regulatory targets with similar functions [142] revealed a miRNA-miRNA functional synergistic network; however, the study of the changes in miRNA-target interactions between different cancer subtypes has mainly left uncovered.

To further our understanding of the role of miRNAs in lung cancers, we aim to identify differentially expressed miRNAs while considering miRNA-target dysreg-

ulations among different cancer subtypes. We extended the brilliant miRNA-target dysregulation idea from Xu et al. [143] and proposed a novel miRNA clustering strategy to identify miRNA dysregulatory modules. We hypothesize that by identifying the context-specific group structures among the miRNAs, the differential analysis procedure can benefit from a more robust selection of miRNA biomarkers that can accurately predict cancer stages across different subtypes.

## 2.2 Methods

### 2.2.1 Dataset and Notations

We denote the miRNA and mRNA expression profiles as column vectors $\mathbf{x_i} = [x_i^1, x_i^2, \ldots, x_i^s]^\top$ and $\mathbf{y_j} = [y_j^1, y_j^2, \ldots, y_j^s]^\top$ to represent the expression level of miRNA $i$ and mRNA $j$ across $s$ samples, respectively. To represent miRNA and mRNA expressions for a specific group of samples, we denote column vectors $\mathbf{x_i^C} = [x_i^1, x_i^2, \ldots, x_i^{n_C}]^\top$ and $\mathbf{y_j^C} = [y_j^1, y_j^2, \ldots, y_j^{n_C}]^\top$, respectively, where $n_C$ is the number of samples attributed with a particular phenotype $C$, e.g., normal, stage I cancer, stage II cancer, etc. Note, boldface variables are to represent vectors and non-boldface for scalars. Also, for expression data, we use subscripts to identify a specific miRNA or mRNA expression level, and superscripts to identify a sample group.

### 2.2.2 Identification of miRNA Biomarkers for Lung Cancer

As an overview of our pipeline, illustrated in Fig. 2.1, we developed a novel approach to identify miRNA dysregulation modules by detecting changes in miRNA-target associations between different cancer subtypes. First, we identify significant deviations in miRNA-target correlations between two sample groups. For each miRNA-target pair found significantly deviated, we form a connection to build a miRNA-target dysregulation association matrix. From the identified miRNA-target dysregu-

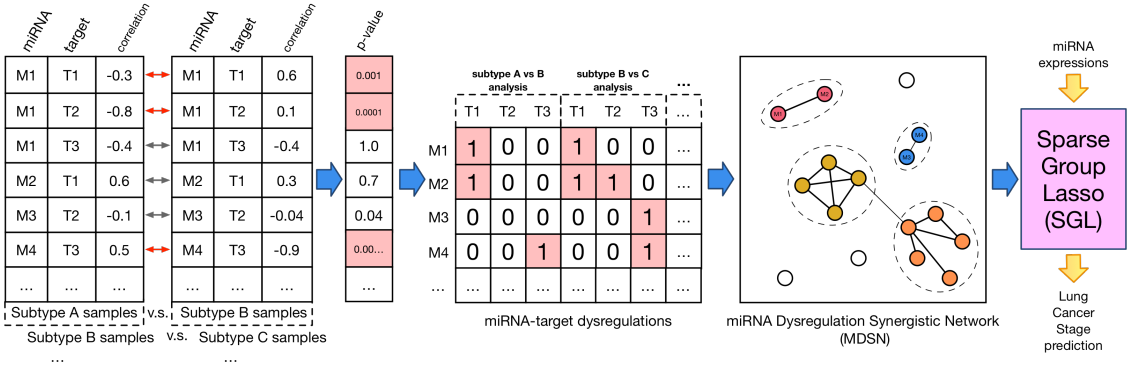



Figure 2.1: **Overview of the MicroRNA Dysregulational Synergism Network pipeline**. In the first step, multiple differential analyses between different subtype groups identified miRNA-target dysregulations. Then, the miRNA dysregulations across multiple subtype analyses is used to form the miRNA-target dysregulation association matrix. Next, the MDSN network is constructed by computing miRNA-miRNA similarity measures, and is used to extract miRNA modules by a graph partitioning method. Finally, provided the extracted miRNA modules, the Sparse Group Lasso performs classification of the cancer stage given a sample's miRNA expression profile.

lations, miRNA modules are extracted such that functionally similar miRNAs belong in the same module if they dysregulate similar targets across multiple cancer subtypes. To accomplish this, a miRNA-miRNA Dysregulational Synergism Network (MDSN) is constructed, and a graph partitioning method is applied to identify significant miRNA modules. At the final step, classification analysis predicts cancer stage and selects relevant biomarkers only from miRNA expression profile data. A Sparse Group Lasso regularization is applied with the intuition that if a miRNA is relevant, the rest of miRNAs in the same module are probably also relevant.

### 2.2.3 Step 1: Identifying miRNA-Target Dysregulations Between Subtypes

For every putative miRNA-target pairs, we incorporated sample-matched miRNA expression and mRNA expression data from distinct sample groups to identify aberrant miRNA-target interactions. More specifically, the aim is to find regulatory

changes by differential analysis of the miRNA-target pair's correlation values between two sample groups of different lung cancer subtypes. This Dysregulation criterion was proposed by Xu *et al.* [143], which defines the difference of the Pearson's correlations between a tumor and a non-tumor group for miRNA $i$ and target $j$ as:

$$Dys_{ij}^{AB} = \frac{\text{cov}(\mathbf{x_i^A}, \mathbf{y_j^A})}{(n_A - 1)\sigma_{x_i^A}\sigma_{y_j^A}} - \frac{\text{cov}(\mathbf{x_i^B}, \mathbf{y_j^B})}{(n_B - 1)\sigma_{x_i^B}\sigma_{y_j^B}} \tag{2.1}$$

where $\sigma_{x_i^A}$ and $\sigma_{x_i^B}$ denote the standard deviation of miRNA $i$ expressions of sample groups $A$ and $B$, respectively. To determine whether the deviation of the correlation between the two groups is significant, Xu *et al.* randomly assigned patients to the two groups and recalculated $Dys$ 10,000 times, and obtained a p-value by the frequency of the random $Dys$ being higher than the actual $Dys$.

To improve the computational performance of obtaining a significance value for the deviation between two correlation coefficients, we instead applied Fisher's transformation [36] as utilized in our previous publication [127]. To summarize, for a given miRNA $i$ and target $j$, we calculated the two Pearson's correlation values $r_A$ and $r_B$ from each sample group then obtained their corresponding z-values $z_A$ and $z_B$ through Fisher's transformation $z = \frac{1}{2}\ln(\frac{1+r}{1-r})$. The z-value for the difference between $z_A$ and $z_B$ is obtained by

$$z_{AB} = \frac{z_A - z_B}{\sqrt{1/(n_A - 3) + 1/(n_B - 3)}}$$

Finally, we can convert the absolute value of $z_{AB}$ to a p-value (two-tailed) and thereby obtain a statistical significance of the difference between two miRNA-target correlations. The cut-off for the p-value threshold was chosen at 0.001, as it has been commonly used as a threshold in several correlation studies.

### 2.2.4 Step 2: Building the miRNA-Target Dysregulation Association Matrix

One primary function of miRNAs is the cleavage of the transcript of its target gene to regulate gene expression. Thus, in the task of identifying aberrant miRNA-target interactions, the inverse correlation should be a prerequisite for candidate miRNA and target pairs to avoid false-positives. In other words, only miRNA-target pairs which have a negative Pearson's correlation in at least one of the sample groups, $A$ or $B$, were considered.

Furthermore, since the primary goal of this study is to discover novel miRNA biomarkers to help understand cancer stage progression, it is essential to consider as many miRNAs as possible. In this study, the miRNA-target relationship prediction algorithms, e.g., TargetScan 7.1 [89] and miRanda [55], were not utilized as the interaction databases only covered a total of 263 miRNAs out of 1881 miRNAs present in the miRNA expression profiles.

For each putative miRNA $i$ and target $j$ considered, we repeated the dysregulation analysis procedure in Step 1 between all pairs of different lung cancer subtypes as independent dysregulation analyses. Then, all miRNA-target dysregulations found significant were encoded by constructing a matrix $\mathbf{A}$ with entry $A_{ij}$ equal to 1 if the p-value of the miRNA $i$ and target $j$ dysregulation passes the p-value threshold and 0 otherwise. For each independent dysregulation analyses, the matrix $\mathbf{A}$ is concatenated. This matrix is interpreted as a new feature set, where each row characterizes a miRNA's dysregulation targets that were present across multiple cancer subtypes dysregulation analyses.

### 2.2.5 Step 3: Calculating miRNA-miRNA Dysregulation Functional Similarity

As it has been reported, miRNAs that are functionally similar tend to have the same targets. Using the identified miRNA-target dysregulations, we inferred

the context-specific functional similarity between two miRNAs by considering their mutual dysregulated targets. The functional similarity score between two miRNAs $p$ and $q$ is calculated by cosine similarity, defined as

$$s(p,q) = \frac{\mathbf{A}_{p\cdot}\mathbf{A}_{q\cdot}^{\top}}{\|\mathbf{A}_{p\cdot}\|_2 \|\mathbf{A}_{q\cdot}\|_2} \tag{2.2}$$

where $\mathbf{A}_{i\cdot}$ is a row vector indicating the dysregulated targets of miRNA $i$. The cosine similarity value ranges $[0, 1]$ and can be interpreted as the number of mutual dysregulation targets shared between two miRNAs normalized by their total connections. By calculating the similarity between every miRNA-miRNA pairs, an adjacency matrix is produced to construct a miRNA-miRNA similarity network. Since it is difficult to uncover cluster structures when the network is dense, it is necessary to prune the weaker miRNA-miRNA connections.

### 2.2.6 Step 4: Constructing the MDSN and Pruning with Scale-free Thresholding

The scale-free topology property exists in most biological graphs, including miRNAs [159], which indicates that the miRNA-miRNA network connections follow a power-law distribution in which more miRNAs tend to have fewer neighbors and fewer miRNAs tend to have more neighbors. A well-known framework, Weighted Gene Co-expression Network Analysis (WGCNA) is utilized to prune lower weight edges with a threshold chosen such that the graphs scale-free property still holds while preserving as many edges as possible.

After all miRNA-miRNA pairs' cosine similarity scores are computed, they are used as edge weights in the MDSN. This is constructed by an adjacency matrix $\mathbf{M}$ with entries $M_{pq} = s(p,q)$ for all miRNAs $p, g$. Similar to the approach used in most biological networks, the miRNA node degrees is expected to exhibit a scale-free distribution under some thresholding. We applied the hard-thresholding technique in

WGCNA [154] by removing from the network any edge with weight lower than the threshold, which was chosen to be the least stringent threshold such that the degree distribution maintains a desirable power-law fitting score.

### 2.2.7 Step 5: Identifying miRNA Dysregulation Modules with Community Detection

After pruning of the MDSN, we utilized the graph partitioning approach to extract miRNA modules by assigning miRNA nodes into communities using a modularity objective proposed in the Louvain method [12]. Using a fast greedy iterative procedure, the Louvain method assigns nodes into communities by optimization of the modularity objective, which measures the density of links inside communities compared to links between communities.

To summarize the algorithm, initially, each node is assigned to its own community. At the first phase, node $i$ consider each of its neighbor $j$ and evaluate the gain of modularity if $i$ is placed in $j$'s community, and then selects the neighbor $j$ with the maximum modality gain. This first phase repeats iteratively until convergence. The algorithm then alternates to the second phase to build a new network whose nodes are the newly formed communities found in the first phase. The first and second phase are repeated iteratively until there is only one community that includes all nodes. In the final result, the algorithm gives a hierarchical community structure of all nodes in the MDSN network. The partition in this dendrogram with the highest modularity value by the Louvain algorithm is selected as the miRNA modules assignment.

### 2.2.8 Step 6: Classification of Cancer Stage with Identified miRNA Modules

It is known that a classifier with $\ell$-1 norm regularization is typically used for feature selection in problems with "small n, large p." However, for problems known to have grouped features, adding group information as prior knowledge can improve

feature selection and classification performance. We applied a multi-class logistic classifier with Sparse Group Lasso (SGL) with the intuition that if a miRNA predictor to cancer stage is found relevant, other miRNAs in the same group are also likely relevant since they share similar dysregulation targets across the cancer subtypes.

SGL is a linear logistic classifier with combined $\ell$-1 and Group Lasso $\ell$-2 norm regularization to achieve a sparse solution at both the group and within group level [117]. We used an indicator vector $c_i \in \{0,1\}^k$ to represent the $i^{th}$ sample's reported cancer stage. In this study, $k$ is 5, indicating whether a sample is labeled as normal, stage I, II, III, or IV. The objective function is as follows:

$$\min_{W} \frac{1}{s} \sum_{i=1}^{s} \log(1 + e^{-c_i(W^\top \mathbf{x^i})}) + \lambda\alpha||W||_1 + \lambda(1-\alpha)GL(W) \tag{2.3}$$

where $\lambda$ is the sparsity coefficient, $\alpha$ is the mixing coefficient between $\ell$-1 and Group Lasso $\ell$-2 norm, which is defined as:

$$GL(W) = \sum_{g=1}^{G} \sqrt{|g|} \cdot ||W_g||_2 \tag{2.4}$$

where $|g|$ is the size of the group. The Python package *pylearn-parsimony* was used to train the logistic regression classifier with SGL regularization.

## 2.3 Result

### 2.3.1 Applications in TCGA Non-Small Cell Lung Adenocarcinoma Dataset

We downloaded miRNA and mRNA expression data of the LUAD cohort from The Cancer Genome Atlas (TCGA) [98], utilizing the TCGA-Assembler tool [165]. Expression quantitation of miRNAs was calculated from the BCGSC miRNA profiling pipeline. The mRNA expression profiles were obtained using Illumina HiSeq RNA-Seq (v2). The Read Per Million miRNA Mapped (RPKM) values were log2 transformed and scaled to zero-mean and standard deviation. In total, there were 1881 miRNA

Table 2.1: Sample size characteristics of the TGCA LUAD dataset

| ***Phenotype*** | ***Sample size*** |
|---|---|
| normal (matched) | 20 |
| stage I | 277 |
| stage II | 121 |
| stage III | 84 |
| stage IV | 24 |
| Acinar* | 18 |
| Bronchioloalveolar* | 24 |
| Clear Cell | 2 |
| Colloid* | 10 |
| Micropapillary | 3 |
| Mucinous | 2 |
| Papillary* | 23 |
| Signet Ring | 1 |
| Solid | 5 |
| Mixed Subtype | 107 |
| Not Otherwise Specified | 320 |

*Histological subtypes selected for dysregulation analysis for their sufficient sample size.

expressions and 20,484 mRNA expressions profiled. The sample size characteristics of LUAD subjects are shown in Table 2.1.

#### 2.3.1.1 Identified miRNA-Target Dysregulations Between LUAD Subtypes

We identified significant dysregulations for every miRNA-target pair between 1881 miRNAs and 20,484 mRNAs. Each miRNA-target pair is tested for significant change in correlations between different subtype sample groups. Due to insufficient sample size in some subtypes, only four histological LUAD subtypes were selected for subtypes dysregulation analysis, as outlined in Table 2.1. To build the miRNA-target dysregulation matrix, we performed an independent dysregulation analysis for each pair-wise combination of the four subtypes.

Setting the p-value threshold parameter at $p < 0.001$, we obtained a sum of 1,896,631 miRNA-target dysregulations from a union of six independent dysregulation analyses for the Acinar, Bronchioloalveolar, Colloid, and Papillary subtypes. In other words, we identified miRNA-target dysregulations between Acinar vs. Bronchioloalveolar, Bronchioloalveolar vs. Colloid, Acinar vs. Colloid, and so on. Since it is very likely that false-positives exist among the identified miRNA-target dysregulations, we accounted for this by careful selection of the threshold parameter to prune weaker miRNA synergism similarities.

#### 2.3.1.2 Selection of Threshold Parameter for the Scale-free Topology of MDSN for LUAD Cohort

After identifying miRNA-target dysregulations among the lung cancer subtypes, we computed the miRNA-miRNA cosine similarity score for every pair of miRNAs to construct the MDSN. For every pair of the 1314 miRNAs (found dysregulated), we computed a total of $754,086$ cosine similarity scores. The power law fitting score [154] is defined as $corr(log_{10}(s), log_{10}(p(s)))^2$ where $s$ is the similarity scores and the distribution $p(s)$ is modeled by a histogram of binned data samples. The $R^2$ score computed over all miRNA-miRNA pairs was 0.9135, which satisfies the $R^2 > 0.8$ criterion and indicates the network has a scale-free topology. The similarity score power parameter was kept at $\beta = 1$.

Next, we proceeded to select a hard-threshold parameter to prune edges from the MDSN with a trade-off between maximizing the scale-free topology fit score and maintaining information in the network for modules discovery. The trade-off can be visualized in Fig. 2.3a. We selected the threshold at 0.55, where the scale-free topology score is above 0.8, and pruned all edges which have cosine similarity score lower than 0.55. After edge pruning, the number of non-isolate miRNA nodes remaining

in the MDSN was 423. From the reduced MDSN network, we applied the Louvain community detection method to identify miRNA modules, and the assignment of miRNAs to the module is indicated by color as shown in Fig. 2.2.

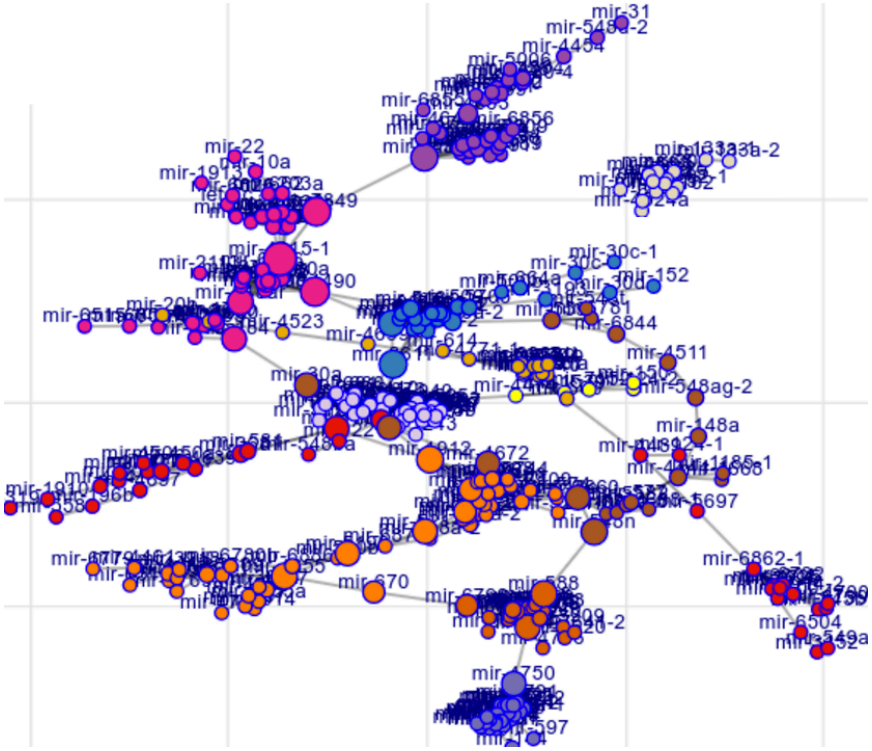



Figure 2.2: **Graph force-layout of the MDSN.** Nodes are positioned closer together if their interconnectivity is high. MiRNA modules assignment, denoted by node color, is determined from the Louvain community detection method which maximizes the modularity objective. It is observed that miRNAs in the same family tend to be grouped as a clique.

### 2.3.2 Applications in the TCGA Lung Squamous Cell Carcinoma Dataset

We also obtained matched miRNA and mRNA expression profiles from the TCGA Lung Squamous Cell Carcinoma (LUSC) cohort [97]. The preprocessing procedure of miRNA and mRNA expression profiles are the same as in the LUAD cohort. An overview of the sample sizes and clinical characteristics is summarized in Table

2.2. According to the clinical data compiled by TCGA-Assembler [165], only less than 20 samples had a histologic subtype labeled, and the majority of samples were labeled as Not Otherwise Specified. Thus, we could not perform the miRNA-target dysregulation analyses from the provided LUSC histological subtypes information due to the insufficient sample size of labeled data.

One reason for this issue is that it has been known the lung squamous cell carcinoma is clinically and genetically heterogeneous, and it is challenging to sub-stratify this heterogeneity. However, a study by Wilkerson *et al.* [139] discovered reproducible and clinically significant LUSC subtypes that can be predicted from the mRNA expression profiles. A representative expression profile for each of the four subtypes, Primitive, Classical, Basal, and Secretory, were summarized by a cluster centroid consisting of 196 genes. Using the cluster centroids representing the four LUSC subtypes, we performed subtype prediction for all LUSC samples using the nearest-centroid classification algorithm proposed in [64].

#### 2.3.2.1 Identified miRNA-Target Dysregulations Between LUSC Subtypes

After the subtype prediction of the LUSC samples were obtained, we tested for significant dysregulation for every miRNA-target pair between 1870 miRNAs and 20,472 mRNAs. Six independent dysregulation analyses were performed for every pairwise combination of the four subtypes, e.g., Primitive vs. Classical, Basal vs. Secretory, Primitive vs. Basal, and so on. A union of the six analyses revealed a sum of 1,560,419 miRNA-target dysregulations found at the p-value cut-off of 0.001.

Table 2.2: Sample size characteristics of the TGCA LUAD dataset

| ***Phenotype*** | ***Sample size*** |
|---|---|
| normal (matched) | 37 |
| stage I | 155 |
| stage II | 125 |
| stage III | 50 |
| stage IV | 3 |
| Lung Basaloid SCC | 10 |
| Lung Papillary SCC | 5 |
| Lung Small Cell SCC | 2 |
| Not Otherwise Specified | 353 |
| Primitive* | 59 |
| Classical* | 96 |
| Basal* | 156 |
| Secretory* | 53 |

*Predicted lung squamous cell carcinoma subtypes selected for dysregulation analyses.

#### 2.3.2.2 Selection of Threshold Parameter for the Scale-free Topology of the MDSN for LUSC Cohort

For every pair of the 1490 miRNAs found with dysregulation patterns across multiple LUSC subtypes, we computed a total of 754,086 cosine similarity scores. Similar to the procedure applied to the network in LUAD cohort, we selected the edge-prune threshold at 0.50, where the scale-free topology criterion $R^2$ score is higher than 0.8, shown in Fig. 2.3b. The number non-isolate miRNA nodes that remained in the MDSN is 391.

### 2.3.3 Extracted miRNA Modules are Consistent Between Independent Subtypes Dysregulation Analyses

To evaluate the consistency of the extracted miRNA modules resulting from independent differential analyses, we compared the miRNA module assignments be-

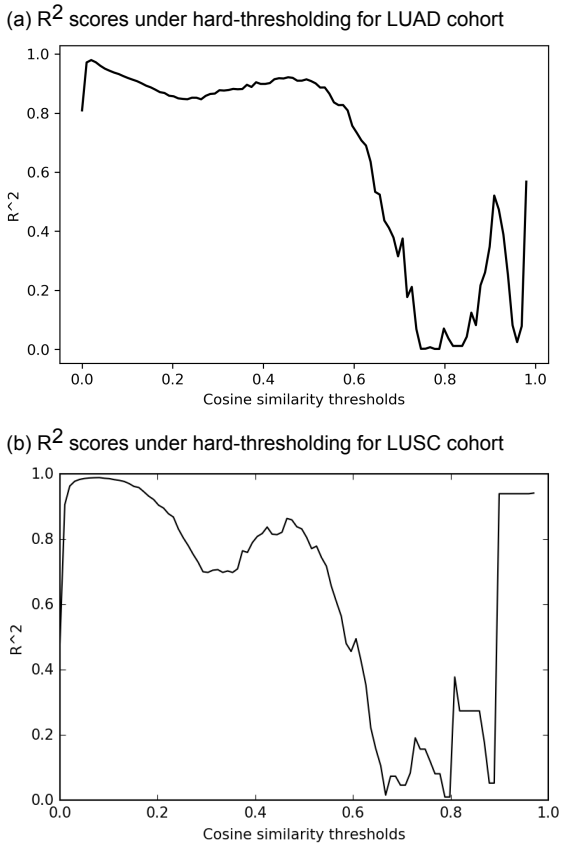



Figure 2.3: **The $R^2$ scale-free criterion fit score at different hard-thresholds.** Edges in the MDSN are pruned if their cosine similarity score is lower than the threshold.

tween different pairwise subtypes dysregulation analyses, combined analyses of all subtypes, normal-tumor dysregulation analysis, and miRNA family information. The score which measures the agreement between two clustering assignments is the Normalized Mutual Information (NMI) metric. As shown in Fig. 2.4, the extracted miRNA modules showed agreement in some of the independent subtypes dysregulation analyses for both LUAD and LUSC cohorts. For example, in Fig. 2.4a, after identifying dysregulations between Bronchio vs. Colloid subtypes and forming the MDSN, the extracted miRNA modules have a similar clusters structure to that of the modules extracted in Acinar vs. Colloid. This may indicate the same groups

of miRNA are dysregulated in the Acinar, Bronchioloalveolar, and Colloid subtypes. Similarly in the LUSC cohort shown in Fig. 2.4b, extracted miRNA modules identified from "Classical vs. Primitive" are highly similar to those from "Basal vs. Primitive," indicating the same groups of miRNA are dysregulated in these three subtypes. Notably, tumor vs. normal miRNA modules were not similar to any of the subtypes dysregulation analyses.

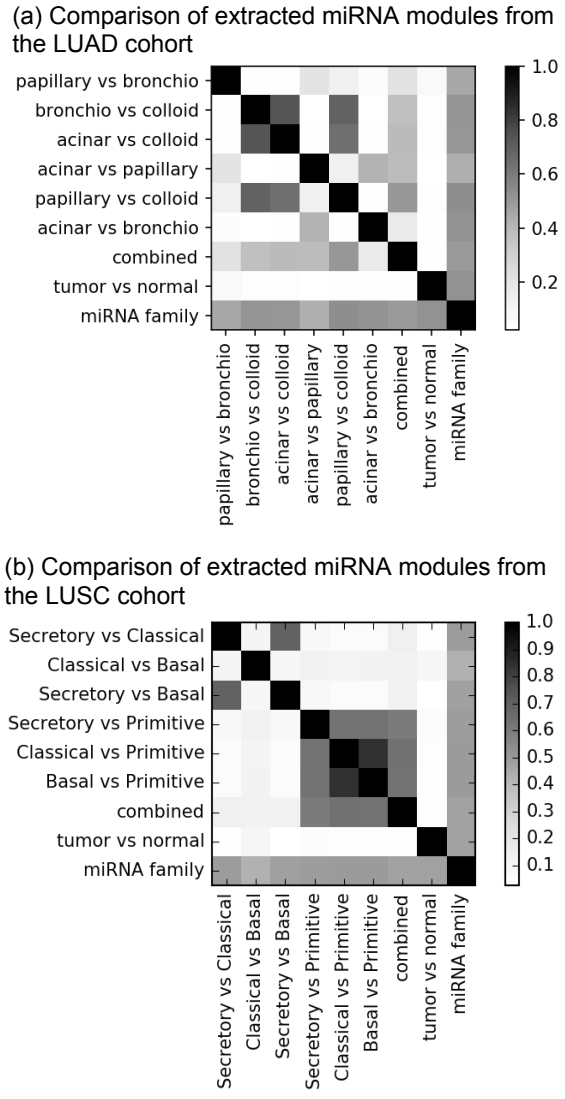



Figure 2.4: Comparison of extracted miRNA modules from the LUAD cohort and the LUSC cohort

### 2.3.4 Incorporating miRNA Modules Information Improves Prediction of LUAD Lung Cancer Stage

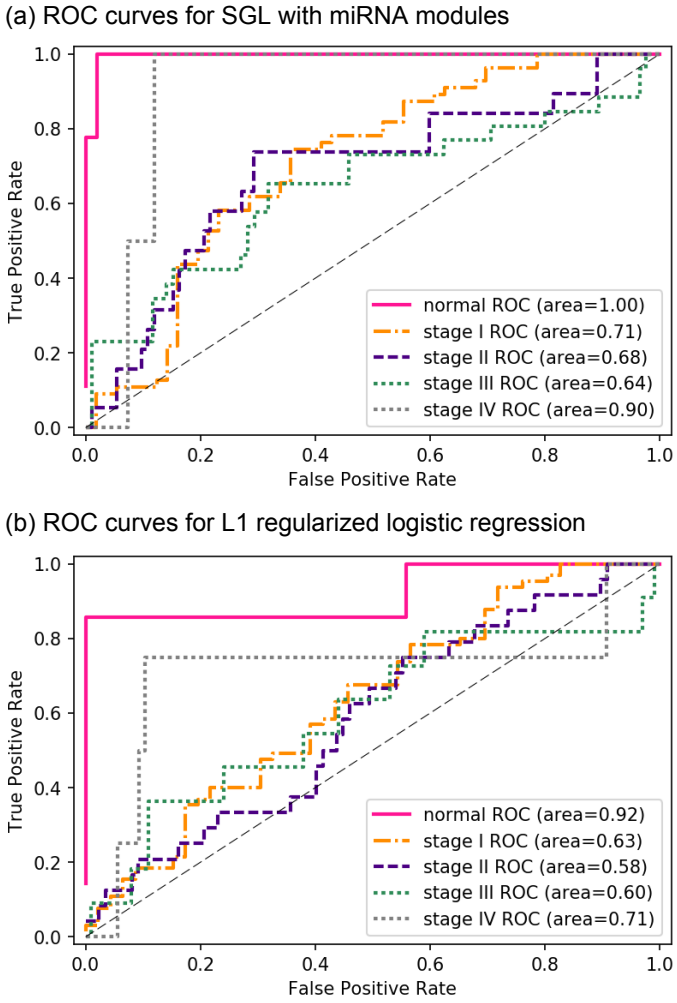



Figure 2.5: **ROC area under the curve scores for prediction of LUAD stages.** Comparison result in multi-stage classification performance shows improved accuracy when incorporating learned miRNA modules to the SGL classifier.

We applied the logistic classifier with SGL using the extracted miRNA modules as prior information to the Sparse Group Lasso regularization. Using a one-vs-rest scheme for multi-class classification, SGL classifies between normal, stage I, stage II,

stage III, and stage IV samples, with numbers of samples corresponding to the first column of Table 2.1. We empirically set the sparsity parameters $\lambda = 1.0$ and $\alpha = 0.5$ that were found to give the best prediction performance from 5-fold cross-validation tests.

To assess whether adding miRNA clusters information improves stage prediction performance, we compared cross-validation scores between SGL and a logistic regression classifier with only $\ell$-1 regularization. With each classifier, we computed the area under the ROC curve rates for each stage from a train-test split of 20%, as shown in Fig. 2.5.

### 2.3.5 MicroRNA Groups Lead to Higher Recall and Precision of Candidate miRNA Biomarkers

To validate whether the extracted miRNA modules aid the SGL classifier in selecting relevant miRNA biomarkers, we investigated how many of candidate miRNA biomarkers selected are known LUAD-associated miRNAs. We utilized a benchmark database of differentially expressed LUAD miRNAs from the dbDEMC [148]. Last updated June 2014 as of this writing, the dbDEMC contains 545 miRNAs reported by high-throughput experiments to be differentially expressed in LUAD. In a normal vs. tumor binary classification experiment using SGL which incorporates the extracted miRNA modules, we showed high precision and recall rates of top-ranked candidate miRNAs to known differentially expressed LUAD miRNAs from the dbDEMC database in Fig. 2.6.

## 2.4 Discussion

In this study, we integrated paired miRNA and mRNA expression data to detect aberrant miRNA-target interactions between lung cancer subtypes to discover novel

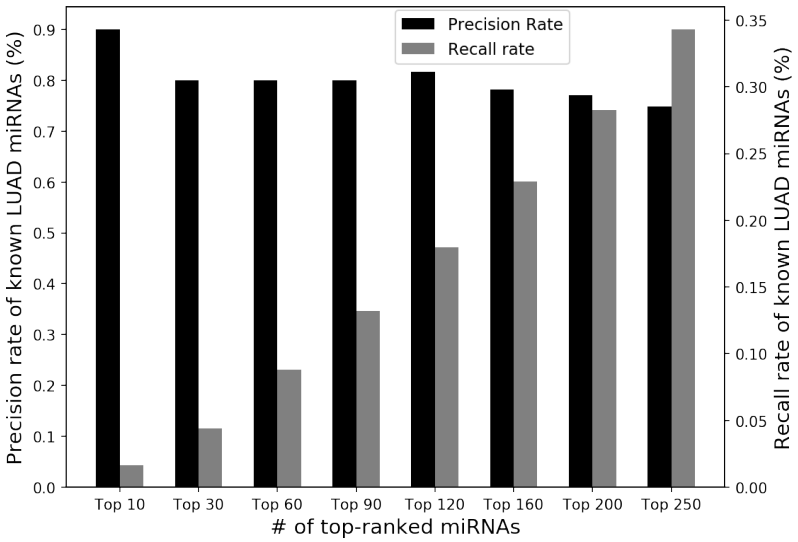



Figure 2.6: **Precision and recall rates of candidate miRNAs selected by SGL.** Among all 246 candidate miRNAs selected by SGL to classify normal vs. tumor, we selected $k$ top-ranked miRNAs by sorting top $k$ coefficients by absolute value. The left y-axis (black bars) represents the percentage of known LUAD miRNAs in the top-ranked set. The right y-axis (gray bars) represents the percentage of miRNAs recalled from known LUAD miRNAs.

miRNA biomarkers to predict lung cancer stages. We have developed an efficient method to identify dysregulations among millions of potential regulatory relationships between 1,881 miRNAs and more than 20,000 mRNAs across multiple lung cancer subtypes. Among all the regulatory relationships considered, 4.9% of the miRNA-target pairs were found to have aberrant behavior across the different subtypes of the lung cancer diseases. Since the LUAD and LUSC are clinically and genetically heterogeneous diseases, utilizing this information would provide a glimpse into the miRNAs' role in cancer pathogenesis in some specific lung cancer subtypes. This was apparent in Fig. 2.4, which shows that some specific lung cancer subtypes possessed similar groups of dysregulated miRNA modules across multiple independent subtypes dysregulation analyses. For instance, note that the Primitive subtype in LUSC has high NMI values between the Secretory vs. Primitive, Classical vs. Primitive, and

Basal vs. Primitive analyses. This indicates that in the Primitive subtype samples, there are possibly a few groups of miRNAs that have a consistent set of dysregulated targets, exclusive to all other LUSC subtypes. It would be interesting to report an analysis on such group of miRNA-target dysregulations in this Primitive subtype, which coincidently has the worst survival outcome ($p < 0.05$) than the other three subtypes [139]. Such an observation may not be apparent with only a normal vs. tumor differential analysis, as it is shown in Fig. 2.4 where the NMI values are near zero in the normal vs. tumor dysregulation analysis compared to all other subtypes dysregulation analyses.

Despite that a growing number of miRNAs have been rigorously studied, the functions of most miRNAs are still unknown. Furthermore, only a small fraction of miRNAs were considered in the target prediction algorithms that provide a database of putative miRNA-mRNA relationships. By considering all potential miRNAs and their targets, our method can be used for novel miRNA functions discovery. However, a primary concern of this task is that selection of various thresholding hyper-parameters may produce unstable results. We performed the miRNA-target dysregulation analysis with varying p-value threshold at 0.01 and 0.001 and found similar patterns in the NMI similarity comparison from extracted miRNA modules in Fig. 2.4. Furthermore, all subtypes dysregulation analyses showed high NMI similarity with the miRNA family assignments without having incorporated this prior knowledge. This implies that despite possible false-positives in identifying miRNA-target dysregulations, the pruned MDSN can still be an excellent tool to reveal miRNA-miRNA functional synergism when inferring novel miRNA functions.

## 2.5 Conclusions

By utilizing a dysregulation metric that allows for analysis of multiple cancer subtypes, we proposed a pipeline to cluster miRNAs with high functional synergism. The extracted miRNA modules, when applied to grouped feature selection, can improve phenotype prediction and result in biomarkers with high precision and recall rate to known LUAD-associated miRNAs. Furthermore, the predicted miRNA modules extracted from different subtype analyses can be used to reveal common miRNA dysregulations across multiple subtypes in heterogeneous cancer types. Since miRNA-target dysregulations are implicated in many cancers, where multi-modal differential analyses between multiple cancer subtypes have mainly left undiscovered, we believe this tool can have broad applications in the development of new diagnosis and treatment strategies.

# CHAPTER 3

# NETWORK REPRESENTATION OF LARGE-SCALE HETEROGENEOUS RNA SEQUENCES WITH INTEGRATION OF DIVERSE MULTI-OMICS, INTERACTIONS, AND ANNOTATIONS DATA

## 3.1 Introduction

Regulatory long non-coding RNAs (lncRNAs) and microRNAs (miRNAs) that influences gene expression post-transcriptionally by interacting to target messenger RNAs (mRNA) form a complex network of transcriptomic interactions. These heterogeneous families of noncoding RNAs are associated with nearly all cellular processes, including cell division, senescence, differentiation, stress response, immune activation, and apoptosis [82, 49, 45]. Recently, lncRNAs are gaining considerable attention as the largest and most diverse class of non-coding RNA, encompassing nearly 30,000 discovered transcripts in human. They are classified as $> 200$ nt transcribed RNA molecules which has a diverse influence upon the function of other non-coding RNAs as well as regulation of protein-coding RNAs. Among many of their known functional interaction mechanisms, lncRNAs are known to act as miRNA decoys, derepress gene expression by competing with miRNAs for shared mRNA targets, or directly regulate gene expression [149]. Additional studies have also indicated that miRNAs can regulate lncRNAs by triggering decay [88], and moreover, some processed lncRNAs can even generate miRNAs [39]. These types of cross-talk functional interactions between lncRNAs and miRNAs to co-regulate gene expressions highlight the complexity of the non-coding RNA regulatory network. Despite that miRNA's repression to target mR-

NAs has been well studied, the discovery of functional interactions for a large number of lncRNAs in the human transcriptome is still at a rather preliminary stage.

Determining the function of the individual lncRNAs remains a challenge as most of these RNA transcripts are currently unannotated, and their known interactions are sparse. Recent advances in RNA sequencing (RNA-Seq), deep sequencing (CLIP-seq, LIGR-Seq), and computational methods allow for an unprecedented analysis of such transcripts and have enabled researchers to generate large-scale interaction and annotation databases. However, the interaction networks generated from such data are often scant and incomplete in the number of lncRNAs covered. Although a large number of lncRNAs have been identified, only a few hundreds have had functional and molecular mechanisms determined to date, as observed in annotation databases such as lncRNAdb [6]. Thus, *in silico* prediction of RNA-RNA interactions have been widely applied in the task of predicting or inferring missing functional interactions, where experimental studies are in short supply due to time and cost. Many graph-theoretic methods have been applied to biological networks with the intuition that RNAs close together in the interaction topology are more likely to be involved in many of the same functions [33]. Typically, the approach is mining the neighborhood structure of nodes in the network topology, in order to suggest that two nodes are likely to be functionally similar if they share many of the same co-interacting neighbors. The positive results utilizing this approach [78] give the motivation that perhaps if ground-truth functional annotation information can be incorporated, it can facilitate the process of suggesting the interactions of the presently unknown RNAs.

To address this challenge, this paper explores the application of a recent advancement in machine learning called "network embedding". It enables learning from the interaction topology and attributes of transcriptome-wide RNAs to accurately predict RNA-RNA functional interactions. Mathematically, our ground truth knowledge

about RNA interactions can be represented by a directed adjacency matrix, whose rows and columns correspond to individual lncRNAs, miRNAs, and mRNAs. In this matrix, its binary (1 or 0) entries can indicate whether or not an RNA was observed to have a functional interaction to another RNA, supported from experimentally-validated interaction databases. The matrix is exceptionally sparse, especially among lncRNAs, i.e., out of millions of possible interactions, only a few thousands have been identified. Currently, a signification fraction of newly discovered lncRNAs lack any identified interactions or functional annotations besides its basic genomic information such as locus biotype and primary transcript sequence [37]. These genes might support important biological cell functions and could potentially serve as targets for genomic, diagnostic, or therapeutic studies. Thus, in the effort to functionally characterize these "hypothetical" lncRNAs, an essential task is to accurately predict their RNA-RNA interactions from sequence.

We propose an algorithm that integrates various existing biological annotation data while simultaneously identifies the complex patterns in the RNA transcript sequences that would allow for accurate prediction of the missing interactions. In this work, we present rna2rna, a novel framework that combines the network-based and deep learning-based approaches to extract a latent representation from nucleotide sequence in order to accurately predict RNA-RNA interaction and identify functional similarity. Our framework processes human lncRNA, miRNA, and mRNA on a transcriptome-wide scale, and was shown to out perform state-of-the-art methods at predicting future interactions for RNA with presently unknown interactions. In summary, the main contributions of our method include:

1. A low-dimensional representation for heterogeneous RNA transcript sequences by integrating existing biological annotation databases, which captures a functional affinity between RNA embeddings.

2. A two-part embedding space to represent the "source context" and "target context" of an individual RNA. The learned embeddings can simultaneously preserve directed cross-talk functional interactions and undirected functional affinities in the lncRNA-miRNA-mRNA topology.
3. An inductive prediction model for novel RNA sequences of any length, applicable for tasks such as inferring missing interactions and clustering of functional similar RNAs.

To our knowledge, no other tool can simultaneously predict heterogeneous lncRNA-miRNA, miRNA-lncRNA, lncRNA-mRNA, miRNA-mRNA, and mRNA-RNA interactions from sequence, while integrate various biological annotation data to characterize RNA-RNA functional similarity.

## 3.2 Related Work

Several network embedding methods have been proposed to predict unobserved links in a network by learning the high-order proximity in its topology. The state-of-the-art network embedding methods, e.g., LINE [125] and node2vec [56] utilizes the second-order proximity, which assume that nodes sharing many of the same second-order neighbors have a high affinity to each other. By learning the neighborhood structure similarity between nodes, their semantic similarity can be identified and is used to predict novel connections. Although such techniques have demonstrated competitive link prediction performance in networks of various domains [53], their prediction performance in biological networks is rather subpar. The reasons for this include: 1) the gene-gene interaction network can be extremely sparse; 2) there is lack of consideration for directionality of RNA-RNA interactions; 3) a lack of an integrative approach for sequence and annotation data; and/or 4) the predictions are transductive in nature, i.e., constrained among only nodes with a connection to

existing nodes in the training set. The method proposed in this paper tackles these limitations and aim to accurately estimate the association strength between every possible RNA-RNA pair.

A number of network embedding methods have been applied to biological networks to either predict gene-gene interactions or to infer biological functions. Due to the extreme sparsity of the known interaction network among lncRNAs to miRNAs and mRNAs, it is pertinent to unravel the functional association between lncRNAs by considering its gene/transcript annotation, functional family annotations, gene-disease association, and sequence similarity [27]. Several efforts in recent studies have already been made to meet the urgent need in this area. For example, Kishan *et al.* [81] uncovered the second-order proximity relationship between interacting genes by integration of the gene regulatory network and gene expression as side information. Additionally, Cho *et al.* [29] proposed a diffusion-based method to predict a protein's function by propagating information through direct and indirect associations in the interaction network. It is important to note that our method differs from these techniques in that it incorporate heterogeneous directed RNA-RNA interaction types, as well as RNA sequence data and annotation attributes as side information. On the other hand, several structure-free sequence-based methods [7, 108, 101] have also been proposed for prediction of protein binding sites, family classification, structure prediction, or RNA-RNA interaction prediction from RNA sequence. These methods utilize machine learning models to learn a latent feature representation of target sequences. Motivated by these works, our method aims to unravel the complex hidden features from the RNA sequence that plays a factor in characterizing its functional similarity and interaction to other RNAs.

## 3.3 Methods

### 3.3.1 Defining the Heterogeneous lncRNA-miRNA-mRNA Interaction Network

We formally define the heterogeneous network of lncRNA, miRNA, and mRNA interactions and functional similarity as two networks of directed and undirected edges. We denote the two networks, $G_1(V, E^d)$ and $G_2(V, E^u)$, having the same set of nodes $V$ (also called vertices) and two set of edges $E^d$ and $E^u$. The set of nodes $V = \{v_1, \ldots, v_n\}$ can also be expressed as $V = \{L, M, N\}$ s.t. $|L| + |M| + |N| = n$, where $L, M, N$ are the sets containing the lncRNA, microRNA, and mRNA heterogeneous nodes, respectively. The set of directed edges $E^d = \{e^d_{ij}\}^n_{i,j=1}$ represent directed regulatory interactions that each specify a source and a target. The undirected edges $E^u = \{e^u_{ij}\}^n_{i,j=1}$ $s.t.$ $e^u_{ji} = e^u_{ji}$ represents the undirected functional affinity associated with the heterogeneous RNA nodes. Each edge $e_{ij}$ is associated with a weight such that $0 \leq e_{ij} \leq 1$, indicating the strength of the connection between RNA $i$ and RNA $j$. If $e_{ij} > 0$, we consider the edge a positive interaction/affinity, and if $e_{ij} = 0$, we consider the edge a negative (non)interaction/affinity. In this paper, we consider weights $e_{ij}$ to be binary, indicating whether RNA $i$ and RNA $j$ has an interaction/affinity.

Furthermore, every node also has an associated RNA sequence in a condensed word embedding representation. An RNA sequence is denoted as $x_i \in \{1, 2, 3, 4\}^{l_i}$, where $l_i$ is the sequence length of RNA $v_i$, and the integer number at each entry indexes the four RNA nucleotides.

### 3.3.2 Directed lncRNA-miRNA-mRNA Interaction Edges by Integrating Various Data Sources

The directed edges represent the directed regulatory interactions between lncRNAs, miRNAs, and mRNAs. Some interactions can be considered as bi-directional, however, in this study, we interpret the directionality as the regulatory effect of

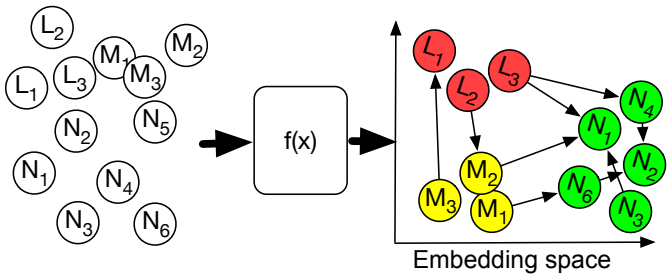



Figure 3.1: An illustration of the heterogeneous lncRNA-miRNA-mRNA tri-module network.

one RNA transcript's abundance causing a direct inhibition/repression/promotion to another RNA transcript's abundance. For instance, we can effectively encode miRNA-lncRNA interactions (e.g., miRNA inducing lncRNA decay) to be separate from lncRNA-miRNA interactions (e.g., lncRNAs acting as miRNA decoys) by using directed edges to represent different types of functional interaction. In this study, the different types of interaction collected from various experimentally-verified interaction databases in the lncRNA-miRNA-mRNA interactome considered are:

- lncRNA-miRNA interaction via miRNA-sponging decoy function of competing endogenous lncRNAs.
- lncRNA-mRNA post-transcriptional gene regulation.
- miRNA-lncRNA interaction by where a binding miRNA causes decay of a lncRNA transcript.
- miRNA-mRNA post-transcriptional interactions causing degradation of target mRNAs.
- mRNA-mRNA interactions in the gene regulatory network.

These heterogeneous interactions are combined into an integrated network, and the associated set of edges is $E^d$, where the binary edge weight $e_{ij}^d \in \{0, 1\}$ indicates whether a regulatory interaction from RNA node $v_i$ to RNA node $v_j$ has been observed

in the literature. A conceptual illustration of lncRNA-miRNA-mRNA regulatory interactions is realized in Fig. 3.1.

### 3.3.3 Undirected RNA-RNA Functional Affinity Edges

Although the directed interaction edges $E^d$ are given, the set of undirected functional affinity edges $E^u$ must be derived from various biological annotation data associated with the RNA nodes. Each node can have up to $K$ annotation data associated with it. For an annotation $k \in K_i$ that is associated with a node $v_i$, we denote a feature vector $a_i^k$ of binary entries representing the presence/absence of a particular attribute in this annotation field.

We aim to capture the functional similarity between two RNA nodes by calculating an affinity score as a similarity measure of characteristics, suggesting a resemblance in RNA function or structure. The approach we take to calculate an affinity between pairs of same-class RNA nodes is by examining the matching annotation attributes that both shares. For any categorical text annotation (e.g., disease association, transcript biotype, RNA structure family, or GO terms) that two RNA nodes $v_i$ and $v_j$ both have been annotated, the attributes in this annotation are first transformed to binary feature representation. For a categorical annotation denoted as $k$, the binary 1-D feature vector associated with RNA node $v_i$ is denoted as $a_i^k$. In this vector contains $m$ binary entries that indicate whether or not the RNA node $v_i$ has been associated with each of the $m$ total possible attributes of this annotation. Using the Sørensen-Dice coefficient score [40], a similarity score between two binary vectors for node $v_i$ and node $v_j$ for feature $k$ can be obtained by:

$$s_{ij}^k = \frac{2(a_i^k \cdot a_j^k)}{2(a_i^k \cdot a_j^k) + |a_i^k|_1 + |a_j^k|_1}$$

This similarity measure ranges $[0, 1]$ and gives higher weight to the common attributes present in both RNAs than by the attributes present in only one RNA. Since most RNAs have null annotations, we computed the Dice coefficient score only between pairs of RNA nodes that have both been annotated.

To obtain an aggregate affinity score between a pair of RNA nodes across all $K$ similarity values, we utilized a modified version of the Gower's Similarity score [52]. For each RNA-RNA pair, Gower's similarity aggregates similarity scores across all the annotation features and perform a weighted average. Typically, a similarity score for a pair of RNA nodes that do not have any associated annotation would be considered a 0, but in this study, we remove these null pairwise similarity from consideration. Thus, Gower's similarity will only aggregate the available pairwise similarity scores from annotation that exists between both nodes to compute the average. Among the RNA node pairs that have only one associated annotation pairwise similarity, we further compute the global pairwise sequence alignment [118] score between these pairs of sparsely annotated RNAs. The Needleman-Wunsch algorithm is used to computes the highest score for matching sequence alignment, normalized by the sequence length, to approximately measure the homology between transcript sequences. The RNA node pairs that do not have any matching annotations were not included in the pairwise calculation.

This Gower's similarity score is computed between all pairs of same-class RNA nodes, and the resulting pairwise affinity matrix is $\mathbf{A}$, where entries $A_{ij} = \sum_{k}^{K_{ij}} \frac{s_{ij}^{k}}{|K_{ij}|}$ with $K_{ij}$ being the set of annotations present in both nodes $v_i$ and $v_j$. The entries $A_{ij}$ will then be selected as edge weights $e_{ij}^{u}$ that represent the functional similarity edges between node. Since our model currently only considers unweighted binary edges, we selected undirected edges with affinity score close to 1.0 or higher than a chosen hard-threshold to be considered as a positive edge. In our experiments, the hard-threshold

was arbitrarily chosen where the number of positive affinity edges covers no more than 0.1% sparsity of the entire affinity matrix. Then, we also uniformly sampled a set of undirected affinity edges with a weight close to 0, indicating a negative edge that suggests functional dissimilarity between a pair of RNA nodes. The number of negative edges chosen such that the ratio of negative edges to positive edges is between 2.0 and 5.0.

## 3.4 Network Embedding with Source-Target Contexts

A network embedding is mapping from each RNA node to a low-dimensional representation, denoted as a mapping function $f : v_i \rightarrow y_i \in \Re^d, \quad \forall v_i \in V$, where $d$ is the dimensionality of the embedding such that $d \ll n$. The embedding $y_i$ associated with each node $v_i$ is learned such that, in this embedding space, nodes preserve some meaningful proximities to other nodes according to the given topology in the networks $G_d$ and $G_u$. Given the learned embeddings for all of the nodes, $Y \in R^{n \times d}$, various downstream prediction tasks can be applied, such as graph reconstruction, visualization, clustering, link prediction, and node classification [53].

We aim to obtain a biologically meaningful embedding representation that simultaneously captures both the regulatory interactions and functional affinities between RNAs. In other words, we train the embeddings to fit both sets of edges from the networks $G_d$ and $G_u$. To accomplish this, we propose the embedding space to have two components: source context and target context. That is, each embedding vector $y_i = [s_i, t_i]^\top$ is represented as a concatenation of the "source context", $s_i \in \Re^{d/2}$, and "target context", $t_i \in \Re^{d/2}$. This embedding representation can simultaneously capture directed and undirected edges by the following definition of proximities:

**First-order directed proximity** to represent the directed regulatory interaction between node $i$'s source context and node $j$'s target context, with:

$$d_1(v_i, v_j) = \sqrt{(s_i - t_j)^2} \tag{3.1}$$

**Second-order undirected proximity** to represent the functional affinity between node $i$ and node $j$, with:

$$d_2(v_i, v_j) = \sqrt{(y_i - y_j)^2} \tag{3.2}$$

The value of these proximities is the Euclidean distance, where the embeddings are selected such that if two nodes have a positive (directed or undirected) edge, its respective embeddings will be more similar, i.e. having a smaller distance. Otherwise, if two nodes have a negative or non-interactions, their embeddings should be more dissimilar, incurring a greater distance. Note, $d_1(v_i, v_j)$ can take on a different value than $d_1(v_j, v_i)$.

Conceptually, the desired effects of applying both directed and undirected proximities using the source-context and target-context in the embedding space are twofold. First, the embeddings can automatically possess the second-order directed proximity, where nodes having similar functional annotations (i.e., a high second-order undirected proximity) will also have a similar set of interacting partners (i.e., first-order directed proximities to other nodes). Likewise, the second-order undirected proximity between a pair of nodes having a similar set of directed interactions will also have a high functional affinity. These two desired effects are in line with the primary intuition that RNAs sharing the same interacting partners are more likely to be involved in many of the same functions. Furthermore, as each of two networks, $G_d$ and $G_u$, are highly sparse and incomplete, training the complementary sets of edges $E_d$ and $E_u$ can help to connect the functional affinities between known RNA's to novel

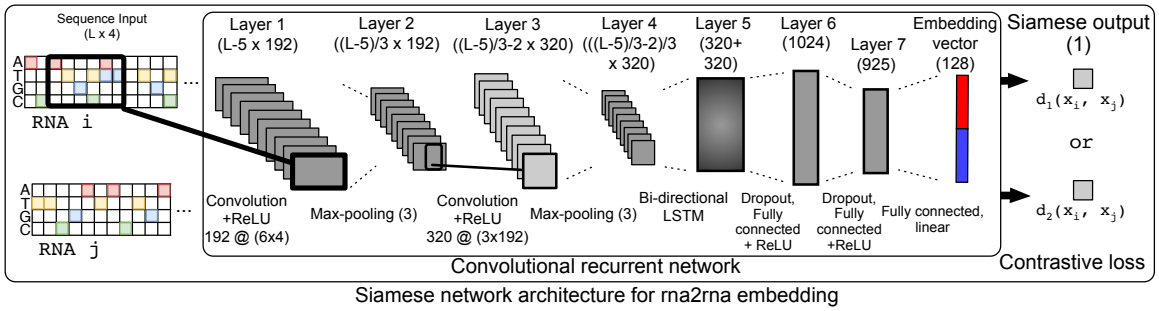



Figure 3.2: **The rna2na network embedding method utilizing Siamese architecture**. Note, the convolutional recurrent network outputs an embedding from a sequence input, while the siamese network takes in two sequence inputs and outputs one number for the (directed or undirected) proximity.

RNA sequences. In the following sections, we demonstrate a methodology to simultaneously apply the two proximity definitions to characterize both the interactions and functional affinity to each RNA transcripts.

### 3.4.1 Representation Learning for RNA Sequences to Reconstruct the Interactions and Functional Topology

Aside from the interaction topology data, each RNA $v_i$ also has an associated transcript sequence, extracted into a one-hot vector representation denoted by $x_i \in \{1, 2, 3, 4\}^{l_i}$, where $l_i$ is the length of the sequence. We propose the network embedding function $f : x_i \rightarrow y_i \in \Re^d$ to be a deep neural network that maps RNA sequence input $x_i$ to an embedding $y_i$ of dimension $d$ that preserves the proximities defined in Eq. 3.1, 3.2. Motivated by its recent successes in facial recognition [115] and speech modeling [123], we repurposed the Siamese network architecture as an interaction network embedding framework in our method rna2rna.

Originally proposed for signature verification [18], Siamese network is an architecture containing an identical pair of the same neural network which shares the same configuration and parameters. A pair of objects can be fed into the two sub-

networks to be encoded, where its resulting embeddings can determine if the two objects are similar or dissimilar. Our goal is to decide the relationship between two RNA sequences by using a convolutional recurrent network to output a real-valued multi-dimensional vector that captures the hidden representation of RNA sequences. More specifically, the network learns to output the embeddings for a pair of RNA sequences, guided by edge $e^u_{ij}$ as the label that indicates whether the pair is functionally similar or dissimilar. For similar pairs of inputs, their embedding is expected to be closer in proximity, and with dissimilar pairs of inputs, their embeddings are to be farther in proximity. Additionally, for an interacting pair of RNAs, the directed edge $e^d_{ij}$ would indicate whether RNA $i$ interacts with RNA $j$ using the corresponding directed proximity. In order for the output embeddings to preserve the proximities across all edges in both $G_d$ and $G_u$ network topologies, we utilize the binary cross-entropy loss function [83], defined as,

$$\begin{aligned} L_1(X, E^d, f) = \sum_{e^d_{ij} \in E^d} & e^d_{ij} \log(d_1(f(x_i), f(x_j))) \\ & + (1 - e^d_{ij}) \log(1 - d_1(f(x_i), f(x_j))) \\ L_2(X, E^u, f) = \sum_{e^u_{ij} \in E^u} & e^u_{ij} \log(d_2(f(x_i), f(x_j))) \\ & + (1 - e^u_{ij}) \log(1 - d_2(f(x_i), f(x_j))) \end{aligned} \tag{3.3}$$

The network weights in $f(x)$ are trained with Stochastic Gradient Descent (SGD) with the standard back-propagation algorithm. Since the subnetworks yield two outputs and their weights are shared, the gradient is summed over the network processing input $x_i$ and network processing input $x_j$. We utilized the RMSprop [126] optimizer to train recurrent network model until convergence. At each SGD iteration, a batch of RNA nodes are sampled along with its associated sets of positive and negative, directed and undirected edges, described further in section 3.4.3.

### 3.4.2 Convolutional Recurrent Network to Obtain Embeddings from Variable-length RNA Sequences

The network inside the Siamese architecture (illustrated in Fig. 3.2) encodes the variable-length RNA sequence inputs through a series of non-linear transformations. Each RNA sequence is represented as a sequence of integers where each element indexes a A,C,T,G nucleotide. The lncRNA, miRNA, mRNA transcript sequences can vary widely in length, between 20 nt to a few thousand nt long. For the network to accept such input, the first layers are 1-D convolutional and pooling layers that yield feature tensors with a timestep dimension that is proportional to the input sequence length. Then, these tensors are passed to a Bidirectional LSTM layer [63], which outputs a fixed-size hidden states vector, and is passed to the next fully-connected layers. With this architecture design, the network is not constrained to only RNA samples with a fixed sequence length specified at training time.

### 3.4.3 Model Optimization with Batch Sampling Strategy

At each training iteration, the model samples a set of training edges and fit the neural network on the RNA sequences associated with these edges. The choice of sampling strategy is hugely important, as the computational complexity is a factor when determining the sampling method that gives the best estimation of the degree distribution in the ground-truth network. It is well-established that most biological interaction networks exhibit the scale-free topology property, apparent in the nodes' degree distribution to follow a power-law degree distribution [4]. In other words, it was frequently observed that a few nodes may have a lot of interactions, and a lot of nodes may have very few interactions. If a set of edges were sampled uniformly, the sampled sub-network would be biased toward nodes with a higher number of connections and nodes with a lower number of connections will be poorly represented.

This problem is further exacerbated by the size imbalance between the interaction sets of well-studied RNA classes and newly-emerged RNA classes. Furthermore, as the size of the network grows, the training approach that iterates through all possible pairs of nodes may become quickly intractable. We instead implement a random node sampling strategy where we first randomly select a set of nodes, then train on the set of edges induced by this sub-graph. However, it was shown that sampling nodes uniformly at random does not retain the power-law distribution [120]. Thus, we employed a biased sampling, where the probability of picking a node $v_i$ is a function of its degree, $r_i$. The probability function proposed by Riad et al. [**?**] is:

$$P(v_i) = \frac{\phi(r_i)}{\sum_{v_j \in V} \phi(r_j)} \tag{3.4}$$

The sampling compression function is chosen to be the square-root function, where $\phi(n) = \sqrt{n}$, to retain the power-law degree distribution while keeping the linear weighting as a ranking for the frequency of each node.

When a batch of nodes $S$ is sampled without replacement from this distribution, each node and its set of positive edges is $\{(v_i, v_j) \mid v_i \in S, v_j \in P_i, P_i \subset S\}$, and negative edges is $\{(v_i, v_k) \mid v_i \in S, v_k \in N_i, N_i \subset S - P_i\}$. In the case of undirected edges, both $P_i$ and $N_i$ are given, however, for directed edges, only $P_i$ are given. To obtain the negative directed edges, we then sample $N_i$ by adopting the approach of negative sampling as proposed in [93], where the ratio of negative edges to that of positive edges incident to each node is between 2.0 and 5.0. To sample the set of nodes $N_i$, we use the distribution given in Eq. 3.4, normalized over nodes in $S$. We allow the negative sampling ratio to be a free parameter to be tuned in our experiments.

Given $S$, the sampled batch of nodes, and $E_S^d, E_S^u$, the set of directed and undirected edges containing both positive and negative interactions incident to $S$, we train the loss function with batch optimization with

$$L(S, E_S^d, E_S^u, f) = L_1(S, E_S^d, f) + \lambda L_2(S, E_S^u, f) \tag{3.5}$$

where $\lambda$ is the coefficient parameter to control the effect of the second-order undirected proximity.

### 3.4.4 Predicting Interaction or Functional Similarity Between Two RNAs

After training is complete, given two RNA sequence inputs $x_i$ and $x_j$, the learned model can output the embeddings $y_i$ and $y_j$, which is used to predict whether a relationship exists between them by computing the respective proximity. Either to predict the existence of an interaction or functional similarity, we use the proximity score $d_1(v_i, v_j)$ or $d_2(v_i, v_j)$, respectively, and then compute a pairwise affinity using a Gaussian kernel:

$$P(v_i, v_j) = \exp(-\gamma * d(v_i, v_j)^2)$$

In our experiments, we calculated all pairwise Euclidean distances and solved for $\gamma$ given $E_u, E_d$, and $Y$. By fitting $\gamma$ to the training set's interactions edges and the predicted pairwise affinity matrix, this method can accurately approximate the distribution of interactions over the whole network.

## 3.5 Results

### 3.5.1 Large-scale Data Integration of lncRNA-miRNA-mRNA Interactions, Annotations, and Sequences

We integrated various experimentally verified interaction databases to build a large-scale lncRNA-miRNA-mRNA interaction network. Additionally, various func-

Table 3.1: **Overview of interaction databases used for data selection, harmonization, and integration for prospective evaluations**. Training sets are comprised of interactions from database versions released before 2015, while validation sets are comprised of updates from the latest database versions. Note, the number source and target RNA nodes listed for validation set are from novel interactions only, which are disjoint from the training set.

| Interaction database | Training Sets | | | |
|---|---|---|---|---|
| | Version | # interactions | # source nodes | # target nodes |
| miRTarBase | 6.0 | 377,318 | 1,618 miRNAs | 14,666 mRNAs |
| DIANA-lncBase | v2 | 53,926 | 631 miRNAs | 2530 lncRNAs |
| NPInter | v2.0 | 85,335 | 12 lncRNAs | 5023 mRNAs |
| lncRNA2Target | v1.0 | 1308 | 79 lncRNAs | 471 mRNAs |
| BioGRID | v3.4 | 313,724 | 13,318 mRNAs | 19,429 mRNAs |

| Interaction database | Validation Sets | | | |
|---|---|---|---|---|
| | Version | # interactions | # novel sources | # novel targets |
| miRTarBase | 7.0 | 64,749 | 12 miRNAs | 702 mRNAs |
| DIANA-lncBase | Predicted | 337,031 | 0 miRNAs | 0 lncRNAs |
| NPInter | v3.0 | 123,054 | 499 lncRNAs | 2346 mRNAs |
| lncRNA2Target | v2.0 | 65,624 | 1037 lncRNAs | 10,825 mRNAs |
| BioGRID | v3.5 | 33,522 | 178 mRNAs | 178 mRNAs |

tional annotation, sequence, disease association, were also integrated to enable extraction of the undirected attribute affinity edges. To maximize the number of genes matched between the different databases, miRNA and mRNA transcripts are indexed by standard gene symbols specified by the MirBase [54] and HUGO Gene Nomenclature Community (HGNC)[106]. LncRNA transcripts are indexed by its Ensembl gene name provided by GENCODE Release 29 [61]. In total, there are 12725 lncRNAs, 1870 microRNAs, and 20284 mRNAs considered in this study, comprised of a comprehensive integration of the various databases illustrated in Table 4.2.

To accomplish the primary task of predicting novel interactions not seen at training time, we propose an experimental setup using prospective evaluation. All models were trained exclusively using the prior version of each interactions databases. Then, we validate the link prediction model by using the set of new interactions

from the latest database version update. This type of evaluation, rarely done in the literature, is extremely important as it allows us to mimic a realistic scenario where the task is to discover novel RNA-RNA interactions, based on our current knowledge.

#### 3.5.1.1 Integration of multiple interaction databases.

In this section, we list the databases utilized for both training and prospective evaluation. In all databases, we selected only the human-species regulatory RNA-RNA interactions and harmonized all miRNA, mRNA, and lncRNA gene names to standardized MirBase, HGNC, and GENCODE gene names.

- **microRNA-target interactions** obtained from miRTarBase [32] database. For training, miRTarBase version 6.0 has a total of 377,318 interactions matched between 1618 microRNAs and 14,666 target mRNAs. For testing, version 7.0 has a total of 64,749 new interactions, of which includes interactions data for 12 novel miRNAs.
- **microRNA-lncRNA interactions** obtained from experimentally verified databases DIANA-lncBase Experimental v2 [102]. There are a total of 53,926 matched interactions between 631 miRNAs and 2530 lncRNAs. Since this database does not have an updated version since the v2 release, we use the DIANA-lncBase Predicted module for evaluation and selected 337,031 interactions with a confidence score greater than 0.9.
- **ncRNA-RNA interactions** from NPInter v2.0 [60], where we filtered only lncRNA-miRNA, lncRNA-mRNA, and miRNA-lncRNA interactions, which resulted in 85,355 interactions between 170 matching lncRNAs and 5023 mRNAs. NPInter v3.0 sharply increased in data and contained 123,486 new interactions between 499 novel lncRNAs and 2346 mRNAs.

- **lncRNA-mRNA interactions** containing lncRNA-mRNA functional regulatory interaction from lncRNA2Target [28], where its latest version v2.0 contained a total of 65,655 interactions between 1037 lncRNAs and 28,866 genes, and its previous version v1.0 contains 1277 interactions between 79 lncRNAs and 471 mRNAs. Note that in this instance, lncRNA2Target v1.0 contained interaction data derived from microarray experiments while v2.0 is from high-throughput RNA-seq experiments, and there is no overlap between the two versions.
- **mRNA-mRNA interactions** obtained from the BioGRID v3.4 database [25], which included more than 313,724 matched interactions among 19,429 mRNAs. For the validation set, BioGRID v3.5 contained 33,522 novel interactions that included 178 novel mRNAs.

After integration of the training databases, self-interactions and redundant interactions edges are removed, and only interactions between RNAs with an associated transcript sequence will be considered. In the validation databases, we selected only interactions that do not overlap with interactions from the training set.

#### 3.5.1.2 Integration of annotation databases to extract undirected attribute affinity edges.

In this section, we outline the annotation databases utilized to provide functional attributes to individual RNAs. After all RNA-RNA pairwise functional affinities were computed, a number of undirected affinity edges were then added to the undirected interactions training set.

- **lncRNA annotations** obtained from the GENCODE Release 28 [61] which contains the transcript biotype annotation. In addition, GO terms for 162 matched lncRNAs were obtained from RNAcentral [34] which aggregated data

from NONCODE [19] and lncipedia [132]. Also, disease associations for 150 lncRNAs were obtained from the LncRNADisease database [26]. After computing the affinities **A** for all lncRNA pairs and filtering second-order undirected affinities at a 0.8 threshold, 65,864 undirected edges were added. With the negative sampling ratio set at 5.0 per positive edge, a total of 329,320 negative edges were added to the undirected edges training set.

- **microRNA annotations** containing miRNA family classified from its seed regions were obtained from the TargetScan Release 7.2 (March 2018) [3]. RNA structure family annotation obtained from Rfam 13.0 [73] and GO terms from RNAcentral were also included. In addition, disease associations for 553 miRNAs were obtained from HMDD miRNA-disease database [68]. After computing the affinities **A** for all miRNA pairs and filtering at a 0.8 threshold, 405 positive edges were added. With the negative sampling ratio set at 5.0 per positive edge, 2025 negative edges were included.
- **mRNA annotations** were obtained from the GENCODE Release 28 [61], and gene annotations were obtained from the HUGO Gene Names database [44].In addition, disease associations for 7577 mRNA genes were obtained from DisGeNet [105]. After computing the affinities **A** for all mRNA pairs and filtering at a 0.8 threshold, 362,362 edges were added. At the negative sampling ratio of 2.0 per positive edge, 724,724 negative edges were included.

#### 3.5.1.3 Preprocessing of RNA Transcript Sequences.

We collected genome-wide human reference lncRNA and mRNA primary sequences from GENCODE Release 29 [61] and miRNA hairpin sequences from miRBase [54]. A transcript is indexed by their gene name, however, many lncRNAs can encode multiple transcript isoform variants. Since some interaction databases do not

identify the specific transcript when referring to the interacting lncRNA, at every training iteration, we uniformly sample from the set of different RNA isoforms for each RNA gene.

In order to speed up the training of the interactions between RNA sequence pairs in the convolutional network, we use batch optimization across multiple GPUs. Due to memory limitations, at training time, we must pad all sequences within the same training batch to one maximum length. The max length was chosen to be 8000 nt long, which would fully represent more than 95% of the longest RNA transcript variants without needing to truncate. For the RNA transcripts exceeding 8000 nt in length, it is important for the network to learn motifs from all regions of the sequence in order to generalize to very long RNA sequences. For this purpose, at each SGD iteration where sequences are sampled, we implement a strategy where a long RNA sequence is truncated either from the first portion or the last portion at random. After training, our model can then process variable-length sequences of any arbitrary length at prediction time. From empirical results, we found this technique does not diminish prediction performance but allows the model to generalize to any RNA sequence length at test time.

### 3.5.2 Comparison Methods

Our experiments include comparative analysis in different evaluation tasks with existing state-of-the-art methods in both varieties of network-based and sequence-based embedding methods. A brief description of the different methods considered are:

- node2vec [56]: A method which preserves higher-order proximity as well as community structure and structural equivalence between nodes.

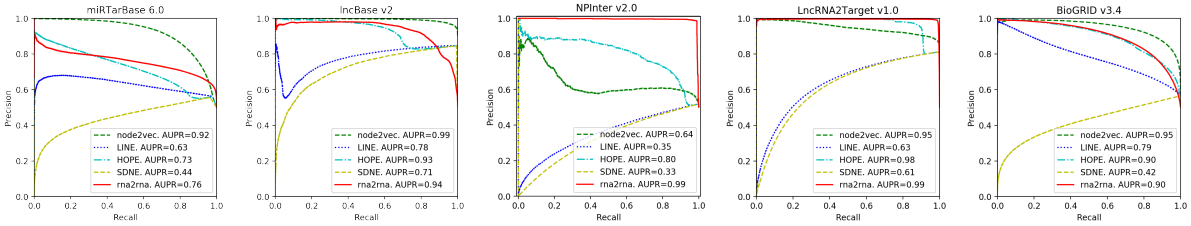



Figure 3.3: Precision-Recall Curve in Graph Reconstruction evaluations to the ground-truth training set of various interaction databases.

- LINE [125]: A method which jointly captures first- and second-order proximities by minimizing the Kullback-Leibler divergence between predicted joint probability distribution for each pair of vertices and the given distribution of training edges.
- HOPE [99]: A method which uses a two-part embedding to reconstruct a directed adjacency matrix, while preserving asymmetric transitive proximity.
- SDNE [133]: An autoencoder-based method that preserves neighborhood proximity between nodes given the network topology.
- BioVec [7]: A word2vec-based model which learns a distributed representation of individual RNA nucleotide sequences by training from a corpus of 3-mers. The corpuses of k-mers were calculated separately for each RNA classes.

In the following experiments, each method were assessed by learning a 128-dimensional embedding representation from the training network. All other free parameters are set according to the default value mentioned in the method's respective papers.

### 3.5.3 Graph Reconstruction.

To assess whether the given methods can efficiently embed nodes from the network to a low-dimensional space while preserving all interactions, we evaluate whether each method can accurately reconstruct the original adjacency matrix of the network

from the training set. After training each method on the training set of interaction databases, all pairwise proximities between the node's embeddings were computed to reconstruct an estimated adjacency matrix. Then, we compute the Precision-Recall Curve of predicted (positive) interactions by validation with the ground-truth positive interactions from the training set. In addition to measuring the recall of the positive interactions, we also sample for non-positive interactions to measure the precision rate, which penalizes for false-positives. Candidate sampling was utilized, where for each node, a number of non-positive interactions are sampled in proportion to the number of positive interactions, where the ratio is $1:1$. Random edges were sampled according to the non-uniform distribution given by Eq. (4) to generate non-interactions for each node, while preventing accidental hits of existing positive interactions. Fig. 3.3 shows a precision-recall curve comparison analysis for the training set across different interaction databases. Additionally, Fig. 3.5 shows the power-law degree distribution of the reconstructed network for each method. Since it is also important for the predicted network to preserve the scale-free topology property from the original interaction network, this result shows rna2rna can reconstruct an interaction network with a power-law degree fit score approximately matching that of the ground-truth training network.

### 3.5.4 Novel Link Predictions.

To evaluate the predictive performance of our model at inferring novel RNA-RNA interactions not seen at training time, we perform a prospective evaluation on the validation set containing future version databases. We compose our training set for this prediction task by a union of all ground-truth interactions set from the miRTarBase 6.0, lncBase v2, NPInter v2.0, lncRNA2Target v1.0, and BioGRID v3.4 databases. All methods then train its model on this combined network. The undi-

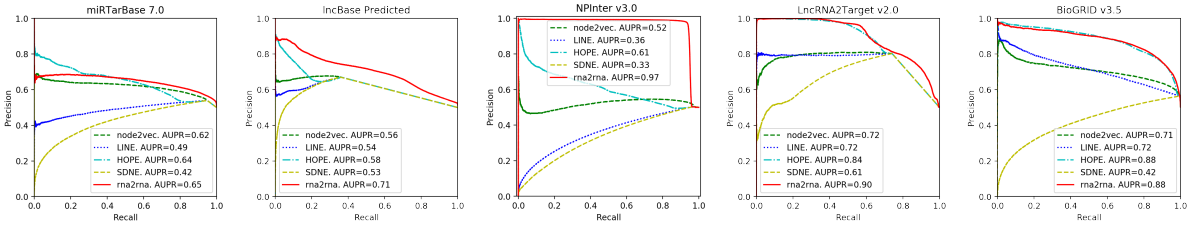



Figure 3.4: **Precision-Recall Curves for Link Prediction**. For each database, each line indicates the Precision-Recall evaluation of a network embedding method to the set of novel ground-truth interactions in that database.

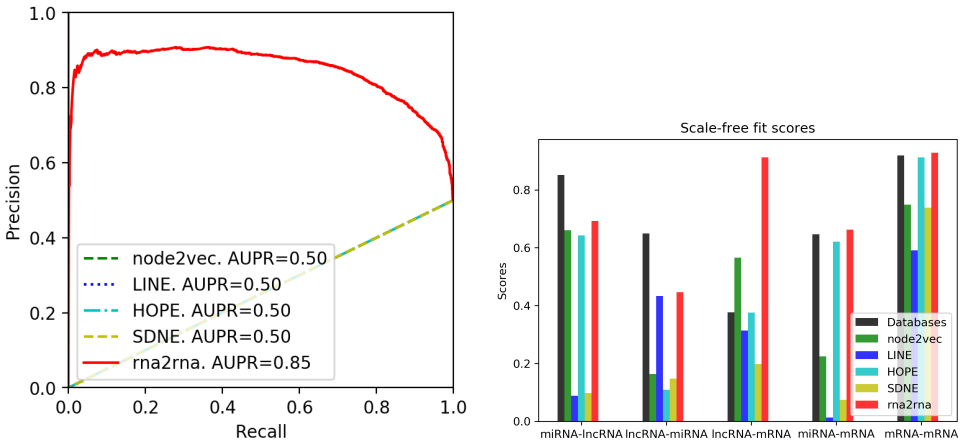
rected functional affinity edges were not included in the training data for any methods besides rna2rna. After the models have been trained, its estimated adjacency matrix is computed and evaluated on the novel interactions from miRTarBase v7.0, lncBase predicted module, NPInter v3.0, lncRNA2Target v2.0, and BioGRID v3.5 databases separately. The set of interactions from the validation set is entirely disjoint from the training set. For a test to differentiate between positive interactions and random noise interactions, we also uniformly sample a number of interactions from the set of all possible pairwise interactions to consider as negative interaction. To do this, we sample from the distribution defined by $P(x, y) = P(x) * P(y), \quad x \in A, y \in B$, where $P(x)$ is from Eq. (4), $A$ and $B$ are the set of source and target RNA nodes respective of the database. This set is denoted as $E^n$, and the number of negative interactions is sampled such that the ratio of negative to positive interactions is 1.0. At evaluation time, the set of ground truth validation edges $E^d$ and random noise $E^n$ edges is used to calculate the precision and recall rates. The true positives are the correctly predicted true interactions, and the false positives are the predicted interactions that are present in the random noise $E^n$ interactions set. Since most network embedding algorithms can yield a predicted probability of the connection, we show the precision-recall curve to evaluate the precision rate at different thresholds of the probability prediction. An area under the precision-recall curve (AUPR) is used to

give a single number indicating the performance of the classifier, which is a good criterion considering it punishes for false-positive predictions. Fig. 3.4 highlights the comparison analysis across five different interaction databases.

In the comparison analysis, all methods were evaluated on the same set of positive and sampled negative interactions. It can be observed that LINE and SDNE do not tend to perform well in this heterogeneous lncRNA-miRNA-mRNA interactions network. For the performance evaluation of predicting the BioGRID mRNA-mRNA gene regulatory interaction set, the interactions between lncRNA and miRNA to mRNA are removed from consideration. It was observed that rna2rna also achieved a superior result in this subnetwork.

#### 3.5.4.1 Inductive Link Prediction to Novel RNA Sequences

In addition to evaluation of predicting missing edges between connected nodes present in the training set, it is also important for our model to infer interactions for novel lncRNAs from sequence. We evaluated the link prediction performance for novel RNA sequences not present in the training set. In this experiment, there are 47 novel lncRNAs with interactions in the validation set that is not present in the training set. We attempt to recall these true interactions only from processing its RNA sequence input and computing their associated interaction to all existing miRNAs and mRNAs. We follow the same procedure proposed above to sample for random negative interactions. Since our method is the only method that can yield an embedding given a novel RNA sequence, the methods node2vec, LINE, HOPE, and SDNE cannot predict from this evaluation as their link prediction is transductive and constrained to only nodes in the largest connected component. After holding out 47 lncRNAs and attempting to recall 3086 its associated true interactions, our method has achieved an average precision score of 0.85, shown in Fig 3.5. Note




(a) (b)

Figure 3.5: (a) Inductive link prediction results for 47 novel lncRNA sequences not seen at training time. (b) Comparison analysis of the power-law degree distribution fit score across multiple RNA-RNA interactions predicted by each methods. The "Databases" bars indicate the scale-free topology fit score of the network composed by the ground-truth edges from lncBase, NPInter, lncRNA2Target, miRTaRBase, and BioGRID, respectively.

that among the 3086 true interactions, which includes lncRNA-miRNA and lncRNA-mRNA interactions, they are comprised of interaction set from the LncRNA2Target v2.0 and NPInter v3.0 updates.

### 3.5.5 Inferring Functional Similarity From Embeddings

The source-target embedding is not only effective at encoding directed RNA-RNA interactions, but it can also capture the undirected functional affinity of RNAs. Since a pair of functionally similar RNAs would have a small Euclidean distance in the embedding space, we can expect a cluster of RNAs to have the same biological functions. To evaluate whether a network embedding method can effectively identify functional similarity, we performed K-means clustering on the learned embeddings and compared the predicted node's clusters to the ground-truth RNA annotations. The ground-truth annotations used for evaluation are RNA functional family [73],

Table 3.2: Clustering Comparison Over 2343 Ground-Truth RNA Functional Family Annotations.

| **Method** | **Homogeneity** | **Completeness** | **NMI** | **# nodes** |
|---|---|---|---|---|
| node2vec | 0.641 | 0.602 | 0.621 | 11735 |
| LINE | **0.689** | 0.614 | 0.650 | 11735 |
| HOPE | 0.525 | 0.571 | 0.570 | 11735 |
| SDNE | 0.613 | 0.588 | 0.600 | 11735 |
| BioVec | 0.376 | 0.467 | 0.417 | 14311 |
| rna2rna* | 0.508 | 0.530 | 0.519 | 14312 |
| rna2rna | 0.685 | **0.620** | **0.651** | 14312 |

rna2rna* denotes the model trained on the directed interactions data alone, without the undirected functional affinity information.

and RNA locus type annotations (e.g., sense intronic lncRNAs, lincRNAs, miRNAs, protein-coding, etc.). If an RNA is known to belong in more than one functional family, we select only the first annotation and discard the rest.

In comparison analysis, we first obtained the embeddings from each of the methods and performed K-Means clustering only on the nodes that have an associated functional annotation. The number of clusters in K-Means is the same as the total number of unique labels in a particular annotation. The evaluation measures used are Homogeneity (higher if nodes are of the same type in each cluster), Completeness (higher if all nodes of the same type are only in one cluster), and Normalized Mutual Information (a mean of the two previous scores). The clustering result of different methods are compared over the RNA family and RNA type annotations in Table 3.2 and Table 3.3. The result shows that although there is a greater number of RNA nodes to assign to clusters, rna2rna embeddings can achieve the highest NMI score over the RNA functional family annotations. In Table 3.3, other methods besides BioVec achieved a lower score, because the local structure of the interaction topology typically contains a mixture of RNA biotypes.

Table 3.3: Clustering Comparison Over 24 Ground-Truth RNA Locus Type Annotations.

| **Method** | **Homogeneity** | **Completeness** | **NMI** | **# nodes** |
|---|---|---|---|---|
| node2vec | 0.147 | 0.089 | 0.111 | 23940 |
| LINE | 0.268 | 0.158 | 0.199 | 23940 |
| HOPE | 0.109 | 0.111 | 0.110 | 23940 |
| SDNE | 0.079 | 0.076 | 0.078 | 23940 |
| BioVec | 0.391 | 0.298 | 0.338 | 32707 |
| rna2rna* | 0.178 | 0.138 | 0.155 | 32530 |
| rna2rna | **0.355** | **0.235** | **0.283** | 32530 |

rna2rna* denotes the model trained on the directed interactions data alone, without the undirected functional affinity information.

#### 3.5.5.1 Training on Interactions Alone Can Reveal RNA Functional Similarity

Here, we test our hypothesis about whether two RNAs are functionally similar if they share the same interacting targets and interacting sources. Since two nodes would have similar embeddings if they have the same set of interacting partners, we can investigate whether training the embeddings from directed interactions alone can produce an embedding that effectively preserves the undirected functional affinity between RNAs. In this evaluation, we trained a rna2rna model on the directed RNA-RNA interaction edges only, while excluding the functional affinities edges. Results in Table 3.3 shows that despite holding out functional annotation information, the resulting RNA embeddings can still approximately preserve cluster structures when compared to ground-truth RNA type annotations.

Table 3.4: **Gene set enrichment analysis over 2000 k-mean clusters**. Each row indicates the highest enriched clustering gene sets comprised of mRNAs, miRNAs, and the candidate lncRNAs for a functional term.

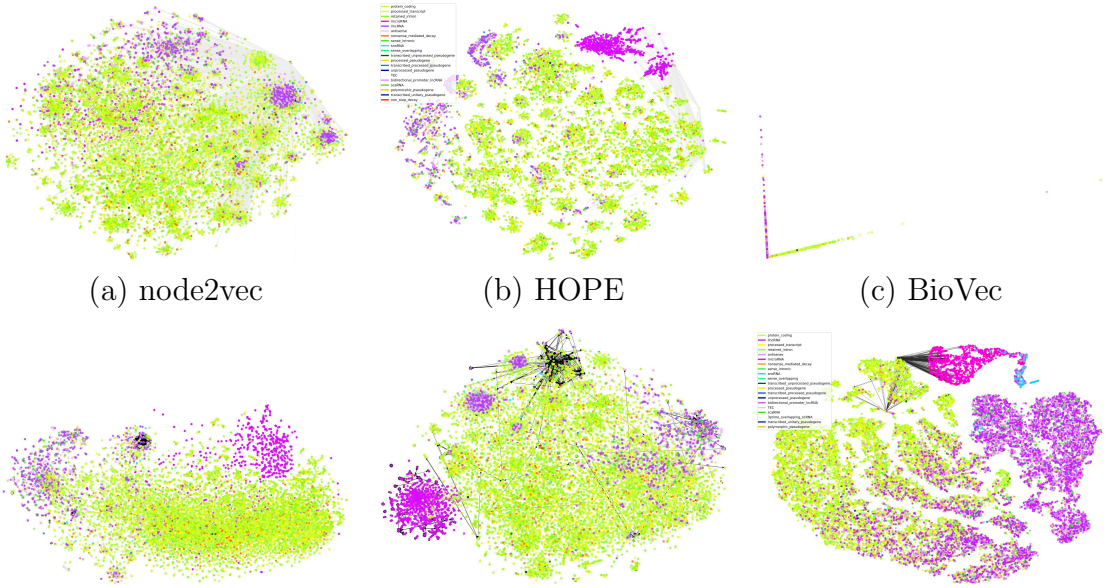
| Gene Set | Candidate lncRNAs | KEGG Term | Overlap | Adj. P-val. |
|---|---|---|---|---|
| ZNF177,ZNF175,ZNF607,ZNF606,ZNF72... | AC022150.4 | Herpes simplex virus 1 infection | 269/492 | 2.47e-323 |
| OR7G2,OR8I2,OR7G1,OR9K2,OR11H1,OR... | AC131571.1,LINC00892,... | Olfactory transduction | 350/444 | 2.47e-323 |
| HIST2H2AA3,HIST2H2AB,HIST1H2AE,HI... | | Systemic lupus erythematosus | 53/133 | 6.935e-77 |
| NOTCH1,HDAC1,CUL2,CBL,HIF1A,EGFR,... | | Pathways in cancer | 25/530 | 2.667e-15 |
| ZNF331,ZNF550,ZNF324B,ZNF490,ZNF7... | | Herpes simplex virus 1 infection | 17/492 | 4.008e-14 |
| GSK3B,HDAC2,PTGER3,PTEN,MAPK8,ERB... | LINC00598 | Pathways in cancer | 25/530 | 1.138e-13 |
| CHRND,PTGIR,EDNRB,MTNR1B,LPAR6,CH... | UCA1 | Neuroactive ligand-receptor inter... | 12/338 | 9.679e-13 |
| RPS15,RPS27,RPS16,RPL31,RPS6,RPL3... | | Ribosome | 11/153 | 1.721e-10 |
| IKBKB,RB1,CDKN1A,SHC1,CTBP1,CDK4,... | | Chronic myeloid leukemia | 11/76 | 4.862e-10 |
| IFNA5,IFNA7,IFNA14,IFNA1,IFNA2,IFNA8 | | Autoimmune thyroid disease | 6/53 | 2.887e-09 |
| RPS28,RPL21,RPS18,RPL11,RPL10L,RP... | | Ribosome | 10/153 | 1.479e-07 |
| PCNA,YWHAQ,CDKN2A,RAD21,MYC,CDK2... | | Cell cycle | 10/124 | 1.551e-07 |
| NFKBIA,PSMD14,PSMC3,PSMD4,PSMC4,P... | | Epstein-Barr virus infection | 11/201 | 2.237e-07 |
| KIR2DS4,KIR2DL1,KIR3DL3,KIR2DL3 | Z99756.1,LINC02346,... | Antigen processing and presentation | 4/77 | 2.625e-07 |
| MAPK9,PRKAB2,PRKAA1,PRKAA2,MAP3K1... | | Tight junction | 10/170 | 4.690e-07 |
| P2RY6,ADORA2A,P2RY2,NMUR1,DRD2,DRD4 | | Neuroactive ligand-receptor inter... | 6/338 | 9.553e-07 |
| CHRNG,HTR1E,PTGER1,KISS1R,NMBR,SSTR5 | TMEM202-AS1, AL355297.4 | Neuroactive ligand-receptor inter... | 6/338 | 2.357e-06 |
| MYH1,MYH2,MYH8,MYH4 | | Tight junction | 4/170 | 2.404e-06 |
| P2RY4,TAAR6,C5AR1,HTR5A,TRHR | AC008125.1 | Neuroactive ligand-receptor inter... | 5/338 | 9.616e-06 |
| NCOA1,MED1,CCND1,SIN3A,NCOA3,PLCG... | | Thyroid hormone signaling pathway | 7/116 | 1.199e-05 |
| RPS9,RPS5,MRPS11,RPL22,RPS3,RPL38... | | Ribosome | 8/153 | 1.232e-05 |
| ABCA3,ABCA4,ABCC10 | | ABC transporters | 3/45 | 1.594e-05 |
| GALR2,GCGR,MLNR,ADRA1B,NTSR2 | | Neuroactive ligand-receptor inter... | 5/338 | 3.511e-05 |
| PARD6B,PRKCI,CTTN,ACTN1,PRKCE,RAP... | | Tight junction | 6/170 | 5.527e-05 |
| LAMA5,TNXB,AGRN | | ECM-receptor interaction | 3/82 | 6.231e-05 |
| TIAM1,CAMK2D,CAMK2A,KDR,CD44,VAV2 | | Proteoglycans in cancer | 6/201 | 1.457e-04 |
| AOC3,PAH | LINC01940, ALDH1L1-AS2 | Phenylalanine metabolism | 2/17 | 2.195e-04 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |

#### 3.5.5.2 Clustering of RNA Embeddings Reveal Highly Enriched Gene Sets

Since rna2rna embeddings have demonstrated functionally similarity in the experiments above, an important next step is to assign putative biological functions to novel lncRNAs. To do this, we perform gene set enrichment analysis on clusters of RNAs, select the cluster with the highest enriched functional term, then associate the lncRNAs belonging in this cluster with this term. In this experiment, the embeddings are trained from both training set and validation set, which includes all known functional interactions and Gene Ontology annotation terms associated with the lncRNAs, miRNAs, and mRNAs. We performed k-means clustering over the embeddings of 32,741 different RNAs, where the number of clusters is 2000. We then performed enrichment analysis on these 2000 clusters using Enrichr [84] over the KEGG Human 2019 [74] terms, which includes both functional and disease pathways. Some of the highest enriched clusters are shown in Table 3.4. Among the 2000 clusters, 559 have an adjusted P-value of less than 0.01, and 139 have an adjusted P-value of less than 0.001. Interestingly, the highest scoring gene sets often contain some lncRNAs not previously associated with these functional terms. It warrants additional experimental studies to verify the functional associations of these lncRNAs.

#### 3.5.5.3 Learned Projection of RNA Embeddings Demonstrates an Organized Distribution

We further visualize the learned 128-dimensional embedding to 2-dimensional space using t-SNE [90]. It can be expected that the RNA nodes in this manifold can preserve the local structure of the interactions and functional annotations, as well as exhibit good separation based on their transcript biotype classification. In Fig. 3.6, nodes are colored based on RNA biotype, and only the 5000 top-scoring interaction



(a) node2vec (b) HOPE (c) BioVec

(d) SDNE (e) LINE (f) rna2rna

Figure 3.6: **Visualization of the lncRNA-miRNA-mRNA regulatory interaction network across different methods**, where RNA nodes are mapped to a 2-D projection using t-SNE from the learned 128-D embeddings. Color of a node indicates the RNA locus type, and grayscale lines indicate the top-5000 interactions predicted by each method.

edges are shown to increase visibility. It is observed that the microRNAs are well separated from the rest of the nodes, but the mRNAs and lncRNAs may have some overlap, which is expected since the sequence structure of these two RNA classes is similar. In comparison to other methods, rna2rna can map a much higher number of lncRNAs and a more extensive variety of different RNA classes to the embedding representation.

### 3.5.6 Subnetwork of LncRNAs Shows Promising Novel Function Interactions

We visualize sub-networks of some well-studied lncRNAs including HOTAIR, GAS5, H19, CASC15, SNHG1, SNHG5, PINK1-AS, UCA1, XIST, and ZFAS1. Each of these subnetworks contains ground-truth interactions between the lncRNA and its

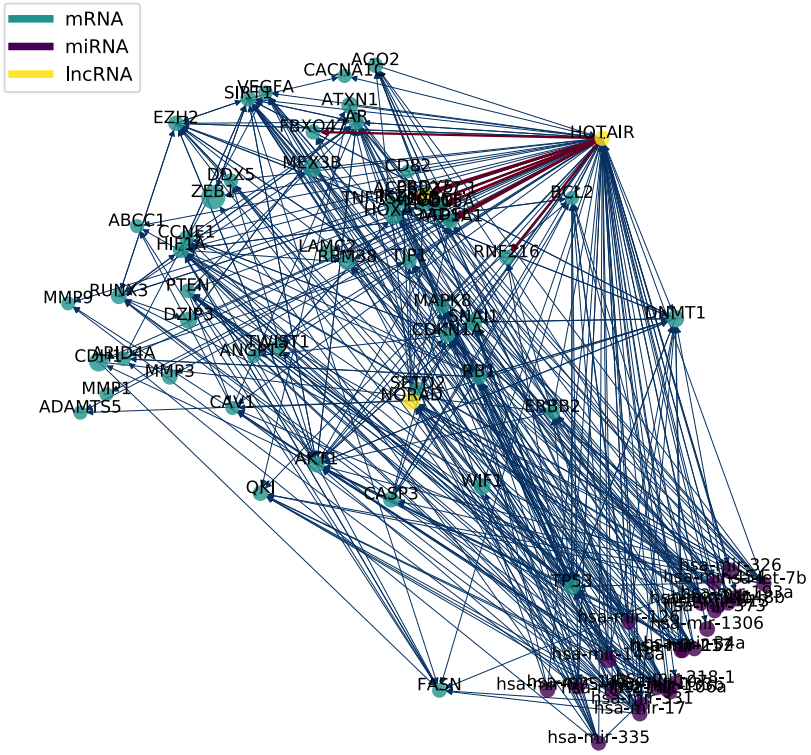



Figure 3.7: **HOTAIR predicted interaction subnetwork**. Nodes placement are determined based on the learned network embeddings. Blue lines represent ground-truth directed regulatory interactions. Red lines represent the top-25 predicted interactions.

miRNA and mRNA interacting neighbors, while predicted interactions are highlighted in red. An illustration of the HOTAIR subnetwork is shown in Fig. 3.7, while others are in the Supplementary Materials. For other well-studied lncRNAs such as H19, GAS5, and SNHG1, the number of interacting partners reached to nearly one thousand, so we selected only interactions supported by two or more databases for better visibility. In each visualization, the placement of the nodes is determined from the force-directed layout of the subnetwork, with the exception of the HOTAIR subgraph, which obtained node placements from the t-SNE transform. In SNHG5, CASC15, and HOTAIR, it was interesting that they are accompanied by another

lncRNA partner. This partner has an interaction to/from the main lncRNA, and also shares some of the same neighbors that the lncRNA is connected to.

## 3.6 Discussion

In this study, we have proposed a method to encode the heterogeneous lncRNA-miRNA-mRNA interaction network, being the union of lncRNA-miRNA, miRNA-lncRNA, lncRNA-mRNA, miRNA-mRNA, and mRNA-mRNA interactions databases. With the framework we have developed, existing annotation data as well as heterogeneous interactions are integrated to enable characterization of RNA sequences using an embedding representation. While this method of integrating different functional annotation sources is simple, its purpose is to allow for characterizing the functional affinities for an extensive number of RNAs, even among sparsely annotated ones. While very few lncRNAs have been annotated for all of its attributes, especially functional annotation or disease association attributes, most have already been annotated with the basic transcript biotype and a transcript sequence. Since we did not constrain the calculations to only RNAs that have all non-empty annotations, we can utilize a less stringent affinity scoring method where a similarity measure between sparsely annotated RNAs can be calculated.

To the best of our knowledge, utilizing the two-part source & target embeddings to model both directed interaction and undirected affinities is novel concept among the network embedding methods. It has a direct purpose at the task of modeling the directed regulatory interactions between biological entities. Considering the node2vec, LINE, and SDNE methods that model the first-order proximity without considering the direction of the edge, the directed regulatory interactions may produce the embeddings to represent functional similarity incorrectly. For instance, suppose there exists a directed edge to represent microRNA $i$ targeting mRNA $j$. If

we model undirected first-order proximity, the resulting embedding representation $y_i$ and $y_j$ would be selected to be similar. This would be misleading because although we know microRNA $i$ targets mRNA $j$, mRNA $j$ does not target microRNA $i$, they belong to different classes of RNA transcripts, and is unlikely to be involved in the same biological functions. By modeling each node's embedding representation with both $s_i$ and $t_i$ separately, we can conceptualize a representation for a biological entity by modeling its functional targeting information and receptive field information.

In the prospective evaluations of recalling the interactions from a future version of various databases, it was shown our method could achieve comparable, and in some cases, superior performance, with other state-of-the-art methods. Rna2rna was able to achieve this accuracy even when predicting more interactions over a more extensive range of RNA nodes since it can obtain embeddings for 32530 unique interacting lncRNA, miRNA, and mRNA nodes. In other methods, only the interactions among a subset of 17905 RNA nodes were considered for link prediction analysis. This is because most other network embedding methods typically only consider the nodes within the largest connected component of the network, while rna2rna can provide a functionally consistent embedding for all nodes in the network that's associated with an RNA sequence. Since it can also handle sequences of various length, rna2rna can provide this mapping for a wide range of RNA transcripts of different structures.

Additionally, since our method was able to map the functional affinity between RNA nodes belonging in disconnected components in the interaction topology, we hypothesize rna2rna could effectively map individual RNA's to a functional manifold in the embedding representation. It is observed in the t-SNE visualization of the embeddings in Fig. 3.6(f) that there is a clear separation between miRNAs, lncRNAs, and mRNAs, albeit overlaps between lncRNAs and mRNAs. Note that although no negative undirected edges between RNAs of different types (e.g., lncRNAs v.s. miRNAs)

were sampled to explicitly indicate different RNA types to have dissimilar embeddings, the network can still make a distinction between their functional roles. This shows that the source-target embedding representation that can effectively encode an RNA's biological function only by its given directed interactions.

## 3.7 Conclusion

Our main contribution proposes a highly versatile architecture aimed at predicting interactions between heterogeneous RNA transcripts while characterizing the functional landscape of non-coding RNAs. Although rna2rna have demonstrated promising performance at various interaction prediction and clustering tasks in experimental results, we believe further improvements to the framework can help it achieve even better performance and usability. Firstly, it cannot be easy to identify the specific binding region from the learned convolutional filters for a given RNA-RNA functional interaction. A future implementation of an attention-based network architecture [129] can provide more power to the framework. Additionally, rna2rna's calculation of the RNA-RNA functional affinity using the Dice distance can be improved, as it simply counts the number of matching functional terms a pair of RNA shares. In this aspect, we plan to apply a method that can calculate a semantic functional similarity, even between non-matching terms. Moreover, while rna2rna was designed to tackle the task of broadening the general knowledge in the human non-coding transcriptome, we also look forward to a modification of the model to allow analysis of the interactome within a specific biological context such as a tissue type or disease condition. Toward this end, we can integrate RNA expression data as additional node attributes to identify specific RNA-RNA interacting pairs within a sample cohort.

In conclusion, we intend this method to be the groundwork for further downstream analysis tasks, where various other downstream genomic prediction tasks such as prediction of gene annotation, gene-disease association, and discovery of unknown gene cluster families can be readily applicable by directly processing the learned embeddings. Further works to this framework can provide an invaluable tool to support significant discoveries in systems biology, especially for newly identified lncRNAs.

# CHAPTER 4

# OPENOMICS: TOOLS FOR INTEGRATING MULTI-OMICS, ANNOTATION, AND INTERACTION DATA

## 4.1 Abstract

Recent advances in sequencing technology and computational methods have generated a variety of heterogeneous genetic and phenotypic characterizations. Leveraging these large-scale multi-omics data is emerging as the primary approach for systemic research of human diseases and general biological processes. As data integration and feature engineering are the vital steps in these bioinformatics projects, there currently lacks a tool for standardized preprocessing of heterogeneous multi-omics and annotation data within the context of a clinical cohort. OpenOmics is a Python library for integrating heterogeneous multi-omics data and interfacing with popular public annotation databases, e.g., GENCODE, Ensembl, BioGRID. The library is designed to be highly flexible to allow the user to parameterize the construction of integrated datasets, interactive to assist complex data exploratory analyses, and scalable to facilitate working with large datasets on standard machines. OpenOmics is also designed to facilitate network-based and graph-theoretic analyses of DNA, RNA, and protein interactions in a high-throughput manner. We demonstrate the wide-ranging use cases of OpenOmics using the Galaxy interfaces to our tool with the goal of maximizing usability and reproducibility of the data integration framework.

**Availability and implementation**: OpenOmics is available in the Galaxy Tool Shed. The source code, example usage and datasets, and documentation are

made freely available under a MIT License at the repository: `https://github.com/BioMeCIS-Lab/OpenOmics`.

## 4.2 Introduction

Recent advances in sequencing technology and computational methods have enabled the means to generate large-scale, high-throughput multi-omics data [87], providing unprecedented research opportunities for cancer and other diseases. These methods have already been applied to a number of problems within bioinformatics, and indeed several integrative disease studies [155, 98, 110, 62]. In addition to the genome-wide measurements of different genetic characterizations, the growing public knowledge-base of functional annotations [34, 37], experimentally-verified interactions [31, 152, 32, 100], and gene-disease associations [68, 105, 26] also provides the prior-knowledge essential for system-level analyses. Leveraging these data sources allow for a systematic investigation of disease mechanisms at multiple molecular and regulatory layers; however, such task remains nontrivial due to the complexity of multi-omics data.

While researchers have developed several mature tools to access or analyze a particular single omic data type [140, 119], the current state of integrative data platforms for multi-omics data is lacking due to three reasons. First, pipelines for data integration carry out a sequential tasks that does not process multi-omics datasets holistically. Second, the vast size and heterogeneity of the data poses a challenge on the necessary data storage and computational processing. And third, implementations of data pipelines are close-ended for down-stream analysis or not conductive to data exploration use-cases. Additionally, there is currently a need for increased transparency in the process of multi-omics data integration, and a standardized data preprocessing strategy is important for the interpretation and exchange of bioin-

formatic projects. Currently, there exist very few systems that, on the one hand, supports standardized handling of multi-omics datasets but also allows to query the integrated dataset within the context of a clinical cohort.

We have developed OpenOmics for the systematic integration and processing of multi-omics datasets. The framework supports various data types, including patient's clinical data, gene/RNA expression, variants, copy number variation, DNA methylation, and even whole slide images. It can cross-reference IDs between different annotation systems to provide an interface that integrates with public interactions and annotations databases. Moreover, it provides an integrated data structure for network analysis to aid biomarker discovery and clinical outcome predictions. In addition to the accessible application programing interface (API), an interactive dashboard web interface is easily deployed by the user to perform exploratory analysis of the data while providing intuitive visualizations. To power the computational load, the backend system utilizes a distributed framework to efficiently parallelize data processing tasks and handle large data that does not fit in memory. To our knowledge, this is the first Python library for multi-omics data integration with a web dashboard interface. The source code and documentation for the package are hosted on GitHub, where it is actively maintained, tested, and deployed as open-source software.

## 4.3 Related Works

There are several existing platforms that aids in the integration of multi-omics data, such as Galaxy, Anduril, MixOmics and O-Miner. First, Galaxy [13] and Anduril [23] are mature platforms and has an established workflow framework for genomic and transcriptomic data analysis. Galaxy contains hundreds of state-of-the-art tools of these core domains for processing and assembling high-throughput sequencing data. Second, MixOmics [111] is an R library dedicated to the multivariate analysis

of biological data sets with a specific focus on data exploration, dimension reduction and visualisation. Third, O-Miner [113] is web tool that provides a pipeline for analysis of both transcriptomic and genomic data starting from raw image files through in-depth bioinformatics analysis. However, as large-scale multi-omic data analysis demands continue to grow, the technologies and data analysis needs continually change to adapt with "big data". For instance, the data manipulation required for multi-omics integration requires a multitude of complex operations, but the point and click interface given in existing Galaxy tools can be limiting or not computationally efficient. Although the MixOmics toolkit provides an R programming interface, it doesn't yet leverage high-performance distributed storage or computing resources. Finally, while O-Miner can perform end-to-end analysis in an integrated platform, its interim analysis results cannot be exported elsewhere for down-stream analysis.

Aside from integrated analysis platforms, several specialized tools exists for handling single omics data, such as AnnData [140] and Loom[1] files. These Python-based libraries provide an intuitive data structure for expression arrays and side annotations. Loom additionally provides an efficient hdf5-based data format that allows for out-of-memory data processing. While these data structures have been popular for general purpose single-omics analytics, they doesn't yet provide mechanisms for multi-omics data integration.

## 4.4 The OpenOmics Library

OpenOmics consists of five core modules: multi-omics integration, annotation interface, network integration, ad-hoc query, and visualization modules. An overview visualization of the OpenOmics system architecture is provided in Figure 4.1.

[1] https://loompy.org

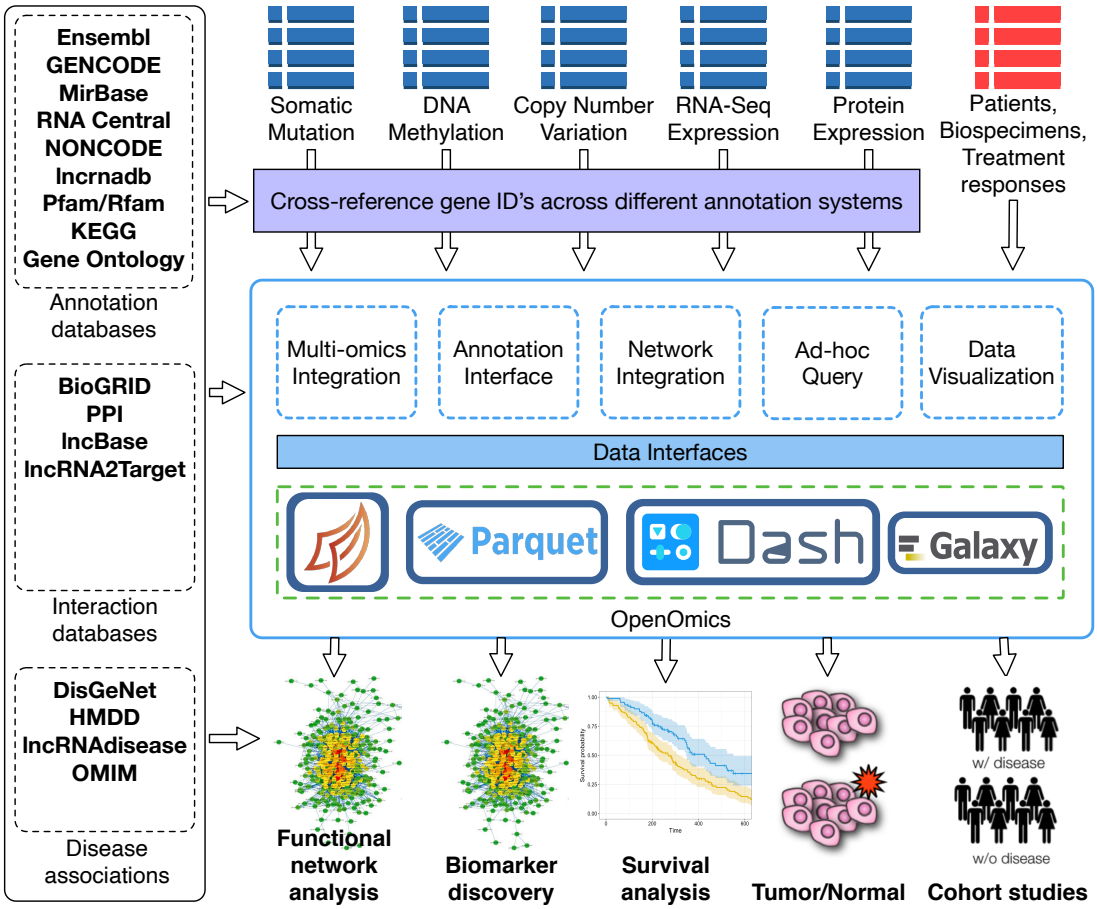



Figure 4.1: Overall OpenOmics System Architecture, Data Flow, and Use Cases.

### 4.4.1 Multi-omics Integration

Tabular data are everywhere in bioinformatics. To record expression quantifications, annotations, or variant calls, data are typically stored in various tabular-like formats, such as BED, GTF, MAF, and VCF, which can be preprocessed and normalized to row indexed formats. Given any processed single-omic dataset, the library generalizes the data as a tabular structure where rows correspond to observation samples and columns correspond to measurements of different biomolecules. The core functionality of the Multi-omics Integration module is to integrate the multiple single-omic datasets for the overlapping samples. By generating multi-omics data for

the same set of samples, our tool can provide the necessary data structure to develop insights into the flow of biological information across multiple genome, epigenome, transcriptome, proteome, metabolome and phenome levels. The user can import and integrate the following supported omic types:

- Genomics: single nucleotide variants (SNV), copy number variation (CNV)
- Epigenomics: DNA methylation
- Transcriptomics: RNA-Seq, single-cell RNA-Seq, miRNA expression, lncRNA expression, microarrays
- Proteomics: reverse phase protein array (RPPA), iTRAQ

After importing each single omics data, OpenOmics stores a Pandas Dataframe that is flexible for a wide range of tabular operations. For instance, the user is presented with several functions for preprocessing of the expression quantifications to normalize, filter outliers, or reduce noise.

Within a study cohort, the clinical characteristics are crucial for the study of a disease or biological phenomenon. The user can characterize the set of samples using the Clinical Data structure, which is comprised of two levels: Patient and Biospecimen. A Patient can have attribute fields on demographics, clinical diagnosis, disease progression, treatment responses, and survival outcomes. Typically, multi-omics data observations are captured at the Biospecimen level and each Patient can have multiple Biospecimens. OpenOmics tracks the ID's of biospecimens and the patient it belongs to, so the multi-omics data are organized in a hierarchical order to enable aggregated operations.

After integrating the multi-omics data with the clinical data, the Multi-omics Integration constructs a data structure that indexes all single-omics data associated with the samples clinical data. The data structure is computationally efficient to enable various data operations for down-stream data analysis tasks, rather than re-

| Data Repository | Annotation Data Available | Index | # entries |
|---|---|---|---|
| GENCODE | Genomic annotations, primary sequence | RNAs | 60660 |
| Ensembl | Genomic annotations | Genes | 232,186 |
| MiRBase | MicroRNA sequences and annotatinos | MicroRNAs | 38589 |
| RNA Central | ncRNA sequence and annotation collection | ncRNAs | 14,784,981 |
| NONCODE | lncRNA sequences | LncRNAs | 173,112 |
| lncrnadb | lncRNA functional annotations | LncRNAs | 100 |
| Pfam | Protein family annotation | Proteins | 18,259 |
| Rfam | RNA family annotations | ncRNAs | 2,600 |
| Gene Ontology | Functional, cellular, and molecular annotations | Genes | 44,117 |
| KEGG | High-level functional pathways | Genes | 22,409 |
| DisGeNet | gene-disease associations | Genes | 1,134,942 |
| HMDD | microRNA-disease associations | MicroRNAs | 35547 |
| lncRNAdisease | lncRNA-disease associations | LncRNAs | 3000 |
| OMIM | Ontology of human diseases | Diseases | 25,670 |

Table 4.1: Public annotation databases and availability of data in the Human genome.

stricting to predefined analyses. For instance, the user can select and group the associated multi-omics data based on customizable criteria on any clinical attributes at the Patient and Biospecimen levels. Finally, processed multi-omics data objects can then be exported as a collection of feature vectors and target labels for machine learning tasks, saved to disk as a compressed dataset, or exported to a compatible Galaxy [71] data structure for other downstream analysis.

### 4.4.2 Annotation Interface

After importing and integrating the multi-omic data, the user can supplement their dataset with various annotation attributes from public data repositories such as GENCODE, Ensembl, and RNA Central. With just a few operations, the user can easily download a data repository of choice, select relevant attributes, and efficiently join a variable number of annotation columns to their genomics, transcriptomics, and proteomics data. The full list of databases and the availability of annotation attributes is listed in Table 4.1.

For each public database, the Annotation Interface module provides a series of interfaces to perform specific importing, preprocessing, and annotation tasks. At the import step, the module can either fetch the database files via a file-transfer-protocol (ftp) URL or load a locally downloaded file. At this step, the user can specify the species, genome build, and version of the database by providing a ftp URL of choice. To streamline this process, the module automatically caches downloaded file to disk, uncompress them, and handle different file extensions, including FASTA, GTF, VCF, and other tabular formats. Then, at the preprocessing step, the module selects only the relevant attribute fields specified by the user and perform necessary data cleanings. Finally, the annotation data can be annotated to an omics dataset by performing a SQL-like join operation on a user-specified index of the biomolecule name or ID. If the user wishes to import an annotation database not yet included in OpenOmics, they can extend the Annotation Dataset API to specify their own importing, preprocessing, and annotation tasks in an object-oriented manner.

An innovative feature of our integration module is the ability to cross-reference the gene IDs between different annotation systems or data sources. When importing a dataset, the user can specify the level of genomic index, such as at the gene, transcript, protein, or peptide level, and whether it is a gene name or gene ID. Since multiple single-omics datasets can use different gene nomenclatures, the user is able to convert between the different gene indexing methods by reindexing the annotation data frame with an index column of choice. This not only allows the Annotation Interface to select and join the annotation data to the correct index level, but also allow the user to customize the selection and aggregation of biological measurements at different levels.

| Data Repository | Interactions Data | # entries |
|---|---|---|
| BioGRID v3.5 | DNA & protein interactions | 313,724 |
| lncRNA2Target v2.0 | lncRNA-mRNA interactions | 65,624 |
| miRTarBase 7.0 | miRNA-mRNA interactions | 377,318 |
| DIANA-lncBase v2 | miRNA-lncRNA interactions | 53,926 |
| NPInter v3.0 | ncRNA-RNA interactions | 123,054 |

Table 4.2: Public interactions databases accessible from OpenOmics.

### 4.4.3 Network Integration

Leveraging the interconnections between the multi-omics levels is necessary to have a holistic view of a biological system. After constructing an integrated multi-omic dataset and annotating the side information, the user can supplement their dataset with various DNA, RNA, and Protein interactions from experimentally-verified data repositories. A full list of interaction databases that is accessible from OpenOmics is listed in Table 4.2.

The primary goal of this feature is to assist users in downstream graph-theoretic analysis by providing an integrated network data structure. The Network Integrator module provides a series of functions to perform import, selection, and network construction tasks. Similar to the Annotation modules, the user can load a public interaction database via a ftp URL or a local file. Then, the user can filter the subset of interactions based on user-defined criteria, such as species, tissue-site, interaction type, and more. The Network Integrator module then constructs an interaction sub-network from the filtered list of interactions. The resulting output is a sparse network object, where "nodes" are individual biomolecules from one or more -omics data, and each "edges" are tagged with interaction type, directed-ness, database source, and any other relevant metadata.

With multiple subnetworks, the user also has the ability to combine them to form an integrated network data structure. This network integration forms a "hetero-

geneous network", where there are multiple types of interactions between biomolecules of different types. These data snapshots essentially contain sets of differently typed edges and the nodes attributes data, and are conductive any graph-theoretical or machine learning analysis tasks. To aid down-stream analysis on this complex data structure, OpenOmics can export the integrated data in multiple formats, such as NetworkX [58], DGL Heterogeneous Graphs [161], PyTorch Geometric Dataset [46], or saved to disk in a compressed format.

### 4.4.4 Ad-hoc Query

With an integrated data structure, OpenOmics provides a framework to perform in-memory tabular computations on the multi-omics dataset. Given the collection of single omics dataframes, the user is able to select and filter the subset of samples or genes which has matching values on any number of clinical or annotation attributes. For any selection queries, the module performs the data selection by constructing a SampleIndex and a GeneIndex. The SampleIndex selects the subset samples from all Biospecimen samples within the clinical cohort, while the GeneIndex selects the subset biomolecules from each of the multi-omics types. Given a specific attribute-matching query, the SampleIndex and the GeneIndex is computed and returns a Multi-Omics DataView, containing a subset of the expression and the annotation data table. As an example on a TCGA multi-omics cancer dataset, a user may filter the data with only samples from patients with a certain survival outcomes and filter gene expressions from a subset of genes, then export the data subset into a file.

Since all data tables utilize the Pandas dataframes, the queried data structure comes with a wide range of tabular computations at the user's disposal. Using the Pandas API on our MultiOmics DataView structure, the user can perform aggregation, sort, select, and operations on any numerical, string, or categorical datatypes.

They are designed to offer quick response time and useful diagnostic feedback on ad-hoc computing operations.

### 4.4.5 Data Visualization

Using the Ad-hoc Query API, OpenOmics contains several data visualization components aimed to provide an interactive data exploration dashboard using the Dash framework[2]. This component is a standalone web server which can be launched on the user's own server in one command line, i.e. `openomics web dataset.omics`, where `dataset.omics` is the path of the dataset saved on disk as in Section 4.5.0.2. Given a Multi-Omics dataset with integrated clinical data, genomics annotation, and network interaction, the following interactive visualization components are available:

- Pivot table: build interactive pivot tables on the Clinical data that allows selecting for a SampleIndex.
- DataTable: an interactive component designed for viewing, selecting, editing, and exploring large expression tables for each of the -omics data type. Selecting on the columns with a substring match allows for selecting on the GeneIndex.
- Network: plot a network containing genes, RNAs, and proteins, along their heterogeneous interactions within the integrated network, where the user can easily choose network layouts and move the view. Node selection with drag-and-click operations allows for selecting on the GeneIndex.
- Interactive Genome Viewer: visualize sequences and overlay feature highlights such as annotations, methylation patterns, variants and mutations. Selecting the genomic region allows for selecting on the GeneIndex.
- NGL MoleculeViewer: 3D visualization of biomolecules such as DNA/RNA and proteins from given sequence of annotation data.

[2]https://plotly.com/dash/

This data-driven interactive dashboard has a grid layout designed to create beautiful and functional visualization components that are draggable, resizable, and responsive. At any selection event on the SampleIndex or GeneIndex in a certain component, OpenOmics instantly queries the data subset and asynchronously update the visuals on all other components. This allows the user to have an interactive data exploration platform where they can perform step-by-step filter operations on different facets of the multi-omics dataset. After the final data selection, the user can export the data subset to another `.omics` file for downstream analysis.

#### 4.4.5.1 Galaxy Tool Interfaces

To increase usability for users with diverse programming backgrounds, we have also developed several interfaces of our toolset to the Galaxy platform. Administrators of a Galaxy server can install the suite of OpenOmics tools via the public Galaxy Tool Shed[3]. Users of a Galaxy instance can use the OpenOmics tools to perform the following use cases:

1. Import single-omics data tables into Parquet data structures.
2. Construct a multi-omics dataset by integrating and indexing multiple single-omics datasets.
3. Select, filter, and group samples by clinical attributes.
4. Select and download relevant public annotate or interaction databases.
5. Export integrated multi-omics dataset and interaction networks for down-stream analysis within Galaxy.
6. Automated provenance tracking saves all OpenOmics operations steps in a history for reproducibility.

[3]https://toolshed.g2.bx.psu.edu/

With an easy-to-use and robust Galaxy-based GUI interface to the primary functionalities of the package, users can reproduce the experiments and integrate them into their own workflow.

## 4.5 System Design

This chapter describes the various implementation details behind the scalable processing and efficient data storage, and the design choices in the development operations.

#### 4.5.0.1 Distributed and Scalable Processing

While the in-memory Pandas dataframes utilized in our data structures are fast, they have size and speed limitations when the dataset size approaches the system memory limit. When this is an issue, the user can enable out-of-memory distributed data processing on all OpenOmics operations, implemented by the Dask framework[4]. When memory resources is limited, data in a Dask dataframe can be read directly from disk and is only brought into memory when needed during computations (also called lazy evaluations). When performing data query operations on Dask dataframes, a task graph containing each operation is built and is only evaluated on command, in a process called lazy loading. Operations on Dask dataframes are the same as Pandas dataframes, but can utilize multiple workers and can scale up to clusters by connecting to a cluster client with minimal configuration. To enable this feature in OpenOmics, the user simply needs to explicitly enable an option when importing an omics dataset, importing an annotation/interaction database, or importing a MultiOmics file structure on disk.

[4]https://dask.org/

There is an argument that conventional data integration systems should instead be designed with a database, which stores all persistent data on disk and only brings data to memory during computations. OpenOmics takes the approach by handling all data in-memory, which allows for faster computations with a wide-range of data processing features. Since data that lives in memory can be computed faster, it allows researchers to perform more interactive and ad-hoc data explorations than traditional SQL-based systems.

#### 4.5.0.2 Data Storage

OpenOmics provides an efficient file format for large-scale integrated multi-omics datasets. It consists of multiple omics data-frames of variable sizes, clinical samples, annotations, and sparse graph objects. The data structure is packaged in a single folder, where OpenOmics can make read and write operations with a collection of highly optimized and compressed binary Parquet[5] files. The Parquet dataset structure for a OpenOmics dataset has the following schema:

- `dataset.omics` - A folder structure for the multi-omics data.
  - `clinical.parq` - Patients and Biospecimen data-frames, containing categorical data types.
  - For each omic type $X$:
    - *X*`_omics.parq` - A data-frame containing numerical data types for a single omics.
    - *X*`_annotations.parq` - Annotation data for the single-omic, containing categorical and string data types.
  - `network.parq` - Sparse graph data, containing a list of edges and the annotations for each edge.

[5]https://parquet.apache.org

A Parquet file structure can be created after the user constructs an integrated dataset either with the Python API or the Dash dashboard.

### 4.5.1 Software Requirements

OpenOmics is distributed as a readily installable Python package from the Python Package Index (PyPI) repository. For users to install OpenOmics in their own Python environment, several software dependencies are automatically downloaded to reproduce the computing environment. We list the primary package dependencies and describe their uses below:

- `pandas`: Core data manipulation operations such as select, filter, and join.
- `dask`: Distributed and out-of-core data manipulation.
- `dash`: Python-based web framework for data-driven visualization.
- `validators,typing,gtfparse`: Automated preprocessing and parsing for a variety of file types.
- `biopython`: Tools for biological computation.
- `astropy,bioservices,requests`: Offline caching of downloaded public datasets.
- `goatools,obonet`: Parsing of gene-ontology structures.
- `networkx`: Construction and manipulation interaction graphs.

OpenOmics is compatible with Python 3.6 or higher, and is operational on both Linux and Windows operating systems. The software requires as little as 4 GB of RAM and 2 CPU cores, and can computationally scale up to large-memory multi-worker distributed systems such as a compute cluster. To take advantage of increased computational resource, OpenOmics simply requires one line of code to activate parallel computing functionalities.

### 4.5.2 Open-source Development Operations

We developed OpenOmics following modern software best-practices and package publishing standards. For the version control of our source-code, we utilized a public GitHub repository which contains two branches, master and develop. The master branch contains stable and well-tested releases of the package, while the develop branch is used for building new features or software refactoring. Before each version is released, we utilizes Travis CI for continuous integration, building, and testing for version and dependency compatibility. Our automated test suite covers essential functions of the package and a reasonable range of inputs and conditions.

For increased visibility and quality of this scientific software, our package was reviewed according to the pyOpenSci [137] standards. Installation instructions, documentation and vignette with examples of the API's essential functions are provided via Read The Docs[6].

## 4.6 Budget Justification

### 4.6.1 Human Resources

While the core data manipulation functionalities of OpenOmics has been completed, several future works remains to be done to further enhance the usability of the library. The first item is to develop an interactive dashboard visualization where users without a programming background can access various data manipulation functions through a web-application interface. The web-based server utilizes the Dash framework [7] which operates with the OpenOmics functional interfaces to generate data-driven visualizations. The analytics interface, named the Ad-hoc Query Engine, will be a stand-alone tool with efficient data pipelines where users can experiment

[6]https://openomics.readthedocs.io/

[7]https://plotly.com/dash/

with various input parameters, data manipulation operations, and see live updates to the multi-omics data. We estimate the software development of the web interface design to cost 40 man-hours and the data analytics functionalities to cost 60 man-hours.

The second item in our list is to increase reproducibility and compatibility with other systems. When performing data manipulations OpenOmics primarily stores the multi-omics data structures as in-memory data-frames. When exporting the preprocessed data for down-stream analysis, it is desirable write the data to disk as a single file for data versioning and sharing. We plan to develop a memory-mappable file structure for the various Multi-Omics, Clinical, and Annotated data structures that is efficient for out-of-core data operations. We estimate the development of this feature to cost 30 man-hours.

### 4.6.2 Infrastructures

OpenOmics can trivially scale to multi-core parallel processing on a single workstation. However, to develop and test distributed operations for OpenOmics to scale to large-memory multi-worker computing clusters, we must have access to a cloud computing platform. To manage and connect to distributed computing instances, we plan to utilize the Dask framework on the Kubernetes platform. We estimate the ideal computational resource for testing to be a AWS EC2 Servers environment that contains at least 10 workers, each with at least 16 GB RAM, 128 GB hard-drive disk, and 4 CPU cores.

## 4.7 Conclusion

A standardized data preprocessing strategy is essential for the interpretation and exchange of bioinformatics research. OpenOmics provides researchers with the

means to consistently describe the processing and analysis of their experimental datasets. It equips the user, a bioinformatician, with the ability to preprocess, query, and analyze data with modern and scalable software technology. As the wide array of tools and methods available in the public domain are largely isolated, OpenOmics aims toward a uniform framework that can effectively process and analyze multi-omics data in an end-to-end manner along with biologist-friendly visualization and interpretation.

# CHAPTER 5

# LAYER-STACKED ATTENTION FOR HETEROGENEOUS NETWORK EMBEDDING

## 5.1 Abstract

The heterogeneous graph is a robust data abstraction that can model entities of different types interacting in various ways. Such heterogeneity brings rich semantic information but presents nontrivial challenges in aggregating the heterogeneous relationships between objects – especially those of higher-order indirect relations. Recent graph neural network approaches for representation learning on heterogeneous graphs typically employ the attention mechanism, which is often only optimized for predictions based on direct links. Furthermore, even though most deep learning methods can aggregate higher-order information by building deeper models, such a scheme can diminish the degree of interpretability by conflating relations with different semantics. To overcome these challenges, we explore an architecture, Layer-stacked ATTention Embedding (LATTE), designed to explore all possible higher-order meta relations at each layer to extract the relevant heterogeneous neighborhood structures for each node type. Additionally, by successively stacking layer representations, the learned node embedding offers a more interpretable aggregation scheme for nodes of different types at different neighborhood ranges. We conducted experiments on several benchmark heterogeneous graph datasets. In both transductive and inductive node classification tasks, LATTE can achieve state-of-the-art performance compared to existing approaches, all while offering a lightweight model. With extensive ex-

perimental analyses and visualizations, the architecture demonstrates the ability to extract informative insights on heterogeneous graphs.

## 5.2 Introduction

Heterogeneous graphs have been commonly used to model complex systems where there are multiple types of relationships among objects of different types. Such a rich semantic structure brings ripe graph mining opportunities for various real-world systems, including knowledge bases, academic graphs, social graphs, biomolecular interactomes, and other multimodal abstractions. Recently, a significant line of research has been explored for representation learning of heterogeneous graphs [42]. The basic principle behind these dimensionality-reduction approaches is to aggregate the high-dimensional information about a node's heterogeneous neighborhood to an embedding vector representation. These node embeddings can then aid in downstream machine learning tasks such as node classification, clustering, and link prediction.

Among the most effective approaches for representation learning on graphs, graph neural network (GNN) methods has gained a dramatic increase in popularity in recent years [80, 59, 130]. While these powerful methods were designed for homogeneous graphs, one can apply them to heterogeneous graphs by ignoring the link/node type distinction and assuming the graph structure to be homogeneous. However, this would be suboptimal, as it's been proven that neglecting the structural dependencies between relations by combining the multi-relations into a single graph will omit important topological properties of the system [9]. Therefore, the primary challenges for heterogeneous graph embedding are maintaining the semantic information and aggregating the multi-relations for respective node types.

There have been several attempts to adopt GNNs to learn multi-relational graphs [114, 156]. More recently, several GNN models designed for heterogeneous

graphs have introduced the attention mechanism for increased interpretation of the aggregation of heterogeneous structures [135, 153, 65]. However, these approaches for heterogeneous graphs face at least one of the following issues. First, some of them are only fitted to aggregate the multi-relations for a single primary node type; thus, they may require a manual design of meta paths. Second, they only optimize for prediction between directly interacting nodes, which is insufficient to capture the heterogeneous graph's global properties [**?**] and higher-order structures. Third, although GNNs with the message-passing paradigm can flexibly propagate high-order information across multiple layers, they do not explicitly preserve the semantics of higher-order meta relations. These shortcomings can often affect the model's scalability, hinder the its generalizability for inductive predictions, or limit the interpretability of the learned model parameters.

In consideration of these current limitations and challenges, we aim to design an approach for heterogeneous GNNs to extract higher-order structures by leveraging the semantic information of all relations and node types. To handle heterogeneity in the graph, we introduce a relation-specific attention mechanism, i.e., depending on the types and direction of a link. As each node type is involved in a subset of all relations, only the relevant relations are aggregated. The mechanism can then capture individual node heterogeneity, where each node is allowed to selectively determine which of its relation-specific neighborhoods contain a more salient signal for a given task.

To generate higher-order meta path connections between nodes of different types, we propose a novel scheme that combines transitive meta relations at each layer successively. As a result, all meta relation sequences of arbitrary length can be enumerated while retaining their semantic context. This process allows the model to distill the unique global structure of each node type by decomposing its hetero-

geneous neighborhoods at different ranges. With a combination of the mechanisms proposed generate higher-order meta paths, our approach can infer the most effective meta paths for inductive prediction even when the full graph data is not available.

Our main contributions with the proposed Layer-stacked ATTention Embedding (LATTE) method for heterogeneous graphs are as followed:

- Propose an architecture that include both node-level and relation-level attentions to effectively capture the heterogeneity among various node types and relation types in the graph.
- Through an efficient mechanism of stacking higher-order attention-based layers, LATTE can compute distant proximity between nodes connected through $n$-hop metapaths and can weigh the importance of various metapaths into consideration.
- Formalizes a learning scheme that can simultaneously infer proximity-based pairwise link prediction and predict heterogeneous node representations for down-stream tasks.

To the best of our knowledge, the proposed approach is the only to introduce a GNN architecture that can both considers node- and relation-type dependent aggregations while efficiently considering all possible high-order meta relations.

## 5.3 Related Work

### 5.3.1 Graph Neural Networks

In recent years, many classes of GNN methods [114, 135, 156, 160, 164, 65] have been developed to handle graph heterogeneity by designing node- and relation-type dependent encoders and aggregators. Although these types of GNNs are flexible for end-to-end supervised prediction tasks, they would only optimize for predictions

between direct interactions. Compared to conventional graph embedding methods [56, 125], standard GNNs generally do not take advantage of second-order relationships between indirect neighboring nodes. Recently, a paper by Huang et al. [67] applied a fusion technique to combine first-order and second-order embeddings at alternating steps. Additionally, the Jumping Knowledge architecture [144] and the GraphSAGE sampling and aggregation [59] has proposed to extend the neighborhood ranges; however, there has yet to be an extension of such techniques to extract higher-order heterogeneous relations.

Only few works have been devoted to mine higher-order relations in heterogeneous structured graphs [147, 153]. Notably, GTN [153] was proposed to enable learning on higher-order meta paths in heterogeneous graphs. It proposes a mechanism that soft-selects a convex combination of the meta relations using attention weights, then applies multiplication of adjacency matrices successively to reveal arbitrary-length transitive meta paths. Similar to GTN, in this paper we focuses on an attention mechanism that infer attention weights not only on the given relations, but also on higher-order relations generated by deeper layers, a feature that existing GNN methods often neglect.

### 5.3.2 Multiplex graph Embedding

Another set of approaches designed for a subclass of the heterogeneous graph are methods for multiplex graphs or multi-relational graphs. Many of the current multiplex or multiview graph embedding methods [47, 158, 92, 107, 121, 116, 48] have proposed strategies for aggregating the learned embeddings of multiple graph "layers" into a single unified embedding. This class of methods typically specify separate objectives for each of the layers to estimate the node features independently, then apply another objective to aggregate the information from all layers together.

Another paradigm is to use random-walk of meta paths to model heterogeneous structures, as proposed in [104, 41, 47]. This class of approaches can learn graph representations without supervised training for a specific task. However, they only learn representations for the primary node type, which consequently requires the customized design of meta paths. Also, they can be sensitive to the random walk's hyper-parameter settings, which may introduce unwanted biases or is computationally costly, thus can lead to lacking performance. Another class of algorithm utilizing embedding translations can also be applied for embedding heterogeneous graphs. For instance, [15] learned linear transformations for each relation to model semantic relationships between entities. While embedding translations can effectively model heterogeneous graphs, they are mainly fitted for link prediction tasks.

## 5.4 Method

### 5.4.1 Preliminary

We consider a heterogeneous graph as a complex system involving multiple types of links between nodes of various types. To effectively represent the complex structure of the system, it is important to define separate adjacency matrices to distinguish the nature of relationships. In this section, we define coherent notations to study the class of heterogeneous information graphs.

**5.4.1.0.1 Definition 3.1: Heterogeneous Information graph,** is defined as a graph $G = (\mathcal{V}, \mathcal{E}, \mathcal{T})$ in which each node $i \in \mathcal{V}$ and each link $e_{ij} \in \mathcal{E}$ are associated with their mapping function $\phi(i) : \mathcal{V} \to \mathcal{T}_{\mathcal{V}}$ and $\phi(e_{ij}) : \mathcal{E} \to \mathcal{T}_{\mathcal{E}}$. $\mathcal{T}_{\mathcal{V}}$ and $\mathcal{T}_{\mathcal{E}}$ denote the sets of node and relation types, where $|\mathcal{T}_{\mathcal{V}}| + |\mathcal{T}_{\mathcal{E}}| > 2$. Since node types can have different feature distributions, the node features representation is given by

$\Phi(i) = \mathbf{x_i} \in \mathbb{R}^{D_m}$, which maps node $i$ of node type $m \in \mathcal{T}_\mathcal{V}$ to its corresponding feature vector $\mathbf{x_i}$ of dimension $D_m$.

We represent the heterogeneous link types as a set of biadjacency matrices $\mathcal{A} = \{\mathbf{A}^{(m,n)} \mid \exists m, n \in \mathcal{T}_\mathcal{V}\}$ where $|\mathcal{A}| = |\mathcal{T}_\mathcal{E}|$. Each meta relation $(m, n)$ specifies a link type between source node type $m$ and target node type $n$, such that $\mathbf{A}^{(m,n)} = \{e_{ij}^{(m,n)} \mid i \in \mathcal{V}_m, j \in \mathcal{V}_n\}$. The biadjacency matrix may consist of weighted links, where $e_{ij}^{(m,n)} > 0$ if there exists a link, otherwise, $e_{ij}^{(m,n)} = 0$. For a $\mathbf{A}^{(m,n)}$ subgraph, we define node $i$'s neighbors set as $\mathcal{N}_i^{(m,n)} = \{j \mid \forall j \in \mathcal{V}_n \; s.t. \; e_{ij}^{(m,n)} > 0\}$. Note that $\mathbf{A}^{(m,n)} \in \mathbb{R}^{|\mathcal{V}_m| \times |\mathcal{V}_n|}$'s size is non-quadratic, and thus does not have a diagonal. Furthermore, this definition assumes relations of directed links, but for a relation $\mathbf{A}^{(m,n)}$ with inherently undirected links, we can inject a reverse relation $\mathbf{A}^{(n,m)} = \{e_{ji} | \forall e_{ij} \in \mathbf{A}^{(m,n)}\}$ into the $\mathcal{A}$ set.

**5.4.1.0.2 Definition 3.2: Meta Relation** To represent higher-order relationships, we denote $(m \xrightarrow{r} p)$ as any length-$r$ sequence of meta relations with source type $m$ and target type $p$. For instance, when $r = 2$, we can connect a relation $\mathbf{A}^{(m,n)} \in \mathcal{A}$ with target type $n$ to another relation $\mathbf{A}^{(n,p)} \in \mathcal{A}$ with matching source type $n$ to yield a second-order relation $\mathbf{A}^{(m \xrightarrow{2} p)}$. Throughout this paper, the meta relations $(m, n)$ notation is overloaded for brevity. In fact, the proposed architecture can handle multiple meta relation types with the same source type and target type, i.e. $\phi(e_{ij}) = \langle \phi(i), \phi(e), \phi(j) \rangle$, without loss of generalization.

### 5.4.2 LATTE: Higher-order Heterogeneous Graph Embedding

In this section, we start by describing the attention-based layers used in the LATTE heterogeneous graph embedding architecture. The attention mechanism utilized in our method extends GAT [130] to infer higher-order link proximity scores for

nodes and links of heterogeneous types. We also introduce the layer building blocks where each layer has the roles of inferring node embeddings from heterogeneous node content while preserving higher-order link proximities.

The input to our model is the set of heterogeneous adjacency matrices $\mathcal{A}$ and the heterogeneous node features $\mathcal{X} = \{\mathbf{X}_m | \exists m \in \mathcal{T}_\mathcal{V}\}$, where $\mathbf{X}_m = \{\mathbf{x}_i \in \mathbb{R}^{D_m} | \forall i \in \mathcal{V}_m\}$. At each $r^{\text{th}}$ layer, the node embeddings output is $\mathbf{h}^r \in \mathbb{R}^{|\mathcal{V}| \times F}$, where $F$ is the embedding dimension, as

$$\mathbf{h}^r = f(\mathbf{h}^{r-1}, \mathcal{A}^r)$$

where $\mathbf{h}_i^0 = \mathbf{x}_i$ and $\mathcal{A}^r$ is the heterogeneous link adjacency matrices in the $r^{\text{th}}$-order.

#### 5.4.2.1 Generating Higher-order Structures

The first-order proximity refers to direct links between any two nodes in the graph among the heterogeneous relations in $\mathcal{A}$. The $\text{r}^{\text{th}}$-order proximity refers to indirect $r$-hop graph structures achieved by combining two matching meta relations. Then, by computing the Adamic-Adar [2] as

$$\begin{aligned} \mathbf{A}^{(m,n,p)} &= \mathbf{A}^{(m,n)} \mathbf{D}^{-1} \mathbf{A}^{(n,p)} \\ \mathbf{D}_{jj} &= \sum_{i \in \mathcal{V}_m} e_{ij}^{(m,n)} + \sum_{k \in \mathcal{V}_p} e_{jk}^{(n,p)} \end{aligned} \tag{5.1}$$

yields $\mathbf{A}^{(m,n,p)}$ as the degree-normalized biadjacency matrix consisting of 2-hop meta-paths from $\mathcal{V}_m$ nodes to $\mathcal{V}_p$ nodes. We define the set of meta relations containing all $r^{\text{th}}$-order relations as the composition between $\mathcal{A}^{r-1}$ and $\mathcal{A}$ meta relation sets,

$$\mathcal{A}^r = \mathcal{A}^{r-1} \times \mathcal{A} \tag{5.2}$$

where $\times$ behaves as a cartesian product that yields the Adamic-Adar only for source-target matching pairs of relations. Note that this is directly applicable to the classical

metapath paradigm [122], where all possible $r$-length metapaths are contained in each separate relation in $\mathcal{A}^r$.

#### 5.4.2.2 Heterogeneous Higher-order Proximities

In order to model the different distribution of links in each relation type $\mathbf{A}^{(m \overset{r}{\rightarrow} p)} \in \mathcal{A}^r$, we utilize a relation-type dependent attention kernel to score every pairwise link. Given any node $i$ of type $m$ and node $k$ of type $p$ in a relation $(m \overset{r}{\rightarrow} p) \in \mathcal{A}^r$, the respective attention kernel $\mathbf{q}^r_{(m \overset{r}{\rightarrow} p)} \in \mathbb{R}^{2F}$ is utilized to compute the scoring mechanism,

$$a_{ik}^{(m \overset{r}{\rightarrow} p)} = \mathbf{q}_{(m \overset{r}{\rightarrow} p)}{}^{\top} [\mathbf{U}^r_m \mathbf{h}_i^{r-1} || \mathbf{V}^r_p \mathbf{h}_k^{r-1}] \tag{5.3}$$

where $\cdot^{\top}$ is the transposition and $||$ is the concatenation operation. The two weight matrices $\mathbf{U}^r_m \in \mathbb{R}^{F \times F}$ and $\mathbf{V}^r_p \in \mathbb{R}^{F \times F}$ encode node features for a pair of nodes and obtain the "source" context and the "target" context, respectively, depending on the node types and the direction of the link. Note that the attention-based proximity score $a_{ij}$ is asymmetric, hence capable of modeling directed relationships where $e_{ij} \neq e_{ji}$.

#### 5.4.2.3 Inferring Node-level Attention Coefficients

Next, our goal is to infer the importance of each neighbor node in the neighborhood around node $i$ for a given relation. Similar to GAT, we compute masked attention on existing links, such that $a_{ik}$ is only computed for first-order neighbor nodes $k \in \mathcal{N}_i^{(m \overset{r}{\rightarrow} p)}$. The attention coefficients are computed by softmax normalization of the scores across all $j$, as:

$$\alpha_{ik}^{(m \overset{r}{\rightarrow} p)} = \frac{exp(\tau_{(m \overset{r}{\rightarrow} p)} a_{ik}^{(m \overset{r}{\rightarrow} p)})}{\sum_{k' \in \mathcal{N}_i^{(m \overset{r}{\rightarrow} p)}} exp(\tau_{(m \overset{r}{\rightarrow} p)} a_{ik'}^{(m \overset{r}{\rightarrow} p)})} \tag{5.4}$$

where $\tau_{(m \overset{r}{\rightarrow} k)}$ is a learnable "temperature" variable initialized at 1 that have the role of "sharpening" the attention scores [30] across the links distribution in a $(m \overset{r}{\rightarrow} k)$

relation. It is expected that $\tau_{(m \xrightarrow{r} k)} > 1$ when the particular link distribution is dense or noisy, thus, integrating this technique allows the attention mechanism to focus on fewer neighbors. Once obtained, the normalized attention coefficients are used to compute the features distribution of a node's by a linear combination of its neighbors for each relation.

#### 5.4.2.4 Inferring Relation Weighing Coefficients

Since a node type $m$ is assumed to be involved in multiple types of relations, we must aggregate the relation-specific representations for each node. Previous works [135, 160, 153] have proposed to measure the importance of each relation type using a set of semantic-level attention coefficients shared by all nodes. Instead, our method chooses to assign the relation attention coefficients individually for each node among only associated relation types, which enables the capacity to capture individual node heterogeneity in the graph.

We denote $\mathcal{A}_{(m \xrightarrow{r})} \subset \mathcal{A}^r$ as the subset of meta relations with source type $m$. Since the number of relations involved in each node type can be different, each node of type $m$ only needs to soft-select from the subset of relevant relations. We utilize another linear transformation directly on node features to predict a normalized coefficient vector of size $|\mathcal{A}_{(m \xrightarrow{r})}| + 1$ that soft-selects among the set of associated relations $\mathcal{A}_{(m \xrightarrow{r})}$ or itself. This operation is computed by:

$$\boldsymbol{\beta}^{r,i} = softmax(\mathbf{W}_m^r \mathbf{h}_i^{r-1} + \mathbf{b}_m^r) \tag{5.5}$$

where $\boldsymbol{\beta}^{r,i} \in \mathbb{R}^{|\mathcal{A}_{(m \xrightarrow{r})}|+1}$ is parameterized by weights $\mathbf{W}_m^r \in \mathbb{R}^{1+|\mathcal{A}_{(m \xrightarrow{r})}| \times F}$ and bias $\mathbf{b}_m^r$ for each node type $m \in \mathcal{T}_{\mathcal{V}}$. Since $\boldsymbol{\beta}^{r,i}$ is softmax normalized, $\beta_0^{r,i} + \sum_{(m,n)}^{\mathcal{A}_{(m \rightarrow)}} \beta_{(m,n)}^{r,i} = 1$, where $\beta_0^{r,i}$ is the coefficient indexed for the "self" choice.

#### 5.4.2.5 Aggregating Layer-wise Embeddings

It is important to not only capture the local neighborhood of a node in a single relation but also to aggregate the neighborhoods among multiple relations and to integrate the node's own features representation. While the first-order embedding represents the local neighborhood among the multiple relations, its $r^{\text{th}}$-order embedding aggregates a larger vicinity by traversing among higher-order meta paths. Along with relation-type attention, LATTE can automatically identify important meta relations of any arbitrary $r$-length by learning an adaptive relation weighing mechanism.

First, we gather information obtained from each relation's local neighborhoods, then combine their relation-specific embeddings. We apply both the node-level and relation-level attention coefficients to a weighted-average aggregation scheme:

$$\mathbf{h}_i^r = \sigma\left(\beta_0^{r,i}\mathbf{U}_m^r\mathbf{h}_i^{r-1} + \sum_{(m\xrightarrow{r}p)}^{\mathcal{A}^r_{(m\rightarrow)}} \beta_{(m\xrightarrow{r}p)}^{r,i} \sum_{k\in\mathcal{N}_i^{(m\xrightarrow{r}p)}} \alpha_{ik}^{(m\xrightarrow{r}p)}\mathbf{V}_p^r\mathbf{h}_k^{r-1}\right) \tag{5.6}$$

where $\sigma$ is a nonlinear function such as ReLU.

Next, we show that multiple LATTE layers can be stacked successively in a manner that allows the attention mechanism to capture higher-order relationships. With this framework, the receptive field of $r^{\text{th}}$-order relations is contained within each $r^{\text{th}}$-order context embedding. Furthermore, as $\boldsymbol{\beta}^{r,i}$ encapsulates each relation in $\mathcal{A}^r$ separately, it is possible to identify the specific relation types that are involved the composite representation. Given the layer-wise representations $\mathbf{h}_i^1, ..., \mathbf{h}_i^r$ of node $i$, we obtain the final embedding output by concatenating all the $R$-order context embeddings, as

$$\mathbf{h}_i = \Big\|_{r=1}^{R} \mathbf{h}_i^r \tag{5.7}$$

where $\mathbf{h}_i \in \mathbb{R}^{RF}, \forall i \in \mathcal{V}$ with $R * F$ as the unified embedding dimension size for all node types.

| Dataset | *Relations (A-B)* | *# nodes (A)* | *# nodes (B)* | *# links* | *# features* | *Training* | *Testing* |
|---|---|---|---|---|---|---|---|
| DBLP | Paper-Author (PA) | 14328 | 4057 | 19645 | 334 | 20% | 70% |
| | Paper-Conference (PC) | 14328 | 20 | 14328 | | | |
| | Paper-Term (PT) | 14328 | 4057 | 88420 | | | |
| ACM | Paper-Author (PA) | 2464 | 5835 | 9744 | 1830 | 20% | 70% |
| | Paper-Subject (PS) | 3025 | 56 | 3025 | | | |
| IMDB | Movie-Actor (MA) | 4780 | 5841 | 9744 | 1232 | 10% | 80% |
| | Movie-Director (MD) | 4780 | 2269 | 3025 | | | |

Table 5.1: Sample characteristics for the heterogeneous graph datasets.

### 5.4.3 Preserving Proximities with Attention Scores

We repurpose the computed attention scores to estimate the heterogeneous pairwise proximities in the graph explicitly. Incorporating this objective not only enables our model for unsupervised learning but also allows the node-level attention mechanism to reinforce highly connected node pairs by taking advantage of weighted links. To preserve pairwise r$^{\text{th}}$-order proximities for all links in each $(m \xrightarrow{r} p)$ relation, we apply the Noise Contrastive Estimation with negative sampling [93] objective as

$$\begin{aligned} L_r(\mathbf{A}^{(m \xrightarrow{r} p)}) = &- \frac{1}{|\mathbf{A}^{(m \xrightarrow{r} p)}|} \sum_{a_{ik}}^{\mathbf{A}^{(m \xrightarrow{r} p)}} a_{ik} \log(\rho(e^r_{ik})) \\ &- \frac{1}{K} \sum_{k}^{K} E_{a_{uv} \sim P(\mathbf{A}^{(m \xrightarrow{r} p)})}[\log \rho(-e^r_{uv})] \end{aligned} \tag{5.8}$$

where $\rho$ denotes the sigmoid function applied to the attention score to infer a probability value. The first term models the observed links, the second term models the negative links drawn from the noise distribution in $(m \xrightarrow{r} p)$, and $K$ is the number of sampled negative links. Typically, $K$ is chosen to be between 2 to 5 times the number of positive links.

### 5.4.4 Model Optimization

To learn from both the heterogeneous graph's attributes and topology, we optimize the proximity-preserving objectives and the downstream objective of the embedding outputs with the standard back-propagation algorithm. For semi-supervised

node classification, a multi-layer perceptron $g(\mathbf{h}_i) = \widetilde{\mathbf{y}}_i \in [0,1]^G$ follows the LATTE layers in order to predicts $G$ labels given the node embedding. The cross-entropy minimization objectives are defined as:

$$L(\mathcal{X}, \mathcal{A}) = -\sum_{i \in \mathcal{V}_Y} \mathbf{y}_i log(g(\mathbf{h}_i)) + \sum_{r=1}^{R} \sum_{\mathbf{A}^{(m \overset{r}{\to} n)} \in \mathcal{A}^r} L_r(\mathbf{A}^{(m \overset{r}{\to} n)}) \tag{5.9}$$

where $\mathcal{V}_Y$ is the set of nodes that have labels, and $\mathbf{y}_i$ is the true label. The first term aims to encode the node embedding representations with attention mechanisms, while the second term reinforces the attention scores by iterating through weighted positive and sampled negative links.

### 5.4.5 Analysis of the Proposed Model

Our model allows for computing embeddings for a subgraph each iteration; thus, it does not require computations involving the global graph structure of all nodes at once. To perform online training at each iteration, an input batch is generated by recursively sampling a fixed number of neighbor nodes [59]. Then, LATTE can yield embedding outputs for a sampled subgraph given the local links and node attributes.

A key observation is that the matrix products $\mathcal{A}^r$ in equation (5.2) do not depend on the model parameters, can thus can be precomputed. In practice, we utilize a sparse matrix multiplication subroutine which yields a time complexity of $\mathcal{O}((\frac{|\mathcal{T}_\mathcal{E}|}{2})^R \times |\mathcal{V}| \times |\mathcal{E}|)$ when generating up to $R$-order heterogeneous structures. For large graphs, distributed computing infrastructures such as Apache Spark can effectively speed up computations.

## 5.5 Experiments

An effective graph representation learning method can generalize to an unseen node by accurately encoding its links and attributes and then "aligning" them to

the embedding space learned from seen (trained) nodes. In this section, we evaluate our method's effectiveness on several node classification and clustering experiments, where the task is to predict node labels for a portion of the graph hidden during training.

### 5.5.1 Datasets

We conduct performance comparison experiments over several benchmark heterogeneous graph datasets. In Table 5.1, a summary of the graph statistics is provided for each of the following datasets:

1. **DBLP**[1]: a heterogenous graph extracted from a bibliography dataset on major computer science journals and proceedings. The dataset have been preprocessed to contain 14328 *papers*, 4057 *authors*, 20 *conferences*, and 8789 *terms*. There are 3 relations types *paper-author*, *paper-conference* and *paper-term* considered. The *author*'s attributes are a bag-of-word representation of publication keywords. The classification task is to predict the label for each author among four domain areas: database, data mining, machine learning, and information retrieval.
2. **ACM**[2]: A small citation graph dataset containing *paper-author* and *paper-subject* relation types among 3025 *papers*, 5835 *authors*, and 56 *subjects* node types. *Paper* nodes are associated with a bag-of-words presentation of keywords as features. The task is to label the conference each paper is published in, among the KDD, SIGMOD, SIGCOM, MobiCOMM, and VLDB venues.
3. **IMDB** [21]: A movie database graph containing *movie-actor* and *movie-director* relations among 4780 *movies*, 5841 *actors*, and 2269 *directors*. Each movie con-

[1]https://dblp.uni-trier.de

[2]https://dl.acm.org

| Dataset | Metric | *metapath2vec* | *HIN2Vec* | *HAN* | *GTN* | *HGT* | *LATTE-1* | *LATTE-2* | *LATTE-2$_{prox}$* |
|---|---|---|---|---|---|---|---|---|---|
| DBLP | $F1_{trans}$ | 0.7518 | 0.7431 | 0.9121 | 0.9203 | 0.8246 | 0.8911±0.003 | **0.9240**±0.003 | 0.9156±0.003 |
| | $F1_{induc}$ | – | – | 0.8666 | 0.8721 | 0.8411 | 0.8620±0.004 | 0.8631±0.003 | **0.8822**±0.032 |
| | # params | 2.3M | 2.3M | 240K | 125K | 217K | 78K | 111K | 111K |
| ACM | $F1_{trans}$ | 0.8879 | 0.8466 | 0.8725 | 0.9085 | 0.8460 | 0.9118±0.005 | 0.9134±0.005 | **0.9153**±0.003 |
| | $F1_{induc}$ | – | – | 0.7909 | 0.8860 | 0.8495 | 0.8988±0.003 | 0.9007±0.003 | **0.9156**±0.003 |
| | # params | 387K | 1.1M | 1.5M | 326K | 458K | 250K | 273K | 273K |
| IMDB | $F1_{trans}$ | 0.4310 | 0.4404 | 0.5394 | 0.5924 | 0.4923 | 0.6066±0.018 | 0.6135±0.014 | **0.6363**±0.007 |
| | $F1_{induc}$ | – | – | 0.3877 | 0.5810 | 0.4836 | 0.6036±0.009 | 0.6117±0.038 | **0.6355**±0.004 |
| | # params | 611K | 1.6M | 1.4M | 243K | 343K | 170K | 196K | 196K |

± denotes the mean and standard deviation over 10 trials.

Table 5.2: Performance comparison of Macro F1 over *trans*-ductive and *induc*-tive node classifications of the test dataset.

tain bag-of-words features of the plot, and the prediction task is to label the movie's genre among Action, Comedy, and Drama.

In each of the datasets, all directed relation have a reverse relation included. All self-loop links have been removed, unless if required for a certain algorithm.

### 5.5.2 Experimental Setup

To provide a consistent and reproducible experimental setup, the preprocessed graphs were obtained from the CogDL Toolkit [22] benchmark datasets. Each of the datasets has been provided with a standard separation of train, validation, and test sets, as well as the full input features and labels set. Since our model evaluates these datasets based on their standard environment, the result from different experiments can be directly compared.

#### 5.5.2.1 Baselines

We verify the effectiveness of our framework by testing multiple variants of LATTE along with the existing approaches. For comparison with some of the state-of-the-art baselines, we consider two main approaches of heterogeneous graph embedding and GNN methods:

- **Metapath-based** methods which requires manual design of metapaths that are limited to the same source and target node type.
  - *Metapath2Vec* [41]: An unsupervised random walk method that utilizes the skip-gram along with negative sampling on meta paths to embed heterogeneous nodes. It has been shown to achieve prominent performance among random walk based approaches.
  - *HIN2Vec* [47]: a state-of-the-art deep neural network that learns embedding by considering the meta paths in an attributed heterogeneous graph. It utilizes a random walk preprocessing, and it does not consider weighing of different meta paths.
- **Heterogeneous GNN** methods that either only considers node- and relation-type dependent encoders, or only considers high-order metapaths, but not both.
  - *HAN* [135]: Employs a GAT-based node-level attention mechanism for heterogeneous graphs. It proposes a hierarchical attention procedure that weighs the importance for each meta path, however only among pre-defined hand-crafted meta paths.
  - *GTN* [153]: Utilizes an attention mechanism that weighs and combines heterogeneous metapaths successively into higher-order structures, then performs graph convolution on the resulting adjacency matrix.
  - *HGT* [65]: Proposes a heterogenous mutual attention mechanism that aggregates from heterogeneous relation types while capturing features and representation space of distinct node types.
- **Proposed method**.
  - *LATTE-1*: The proposed LATTE model with one layer that only considers first-order meta relations. The pairwise proximity preserving objectives is excluded.

- *LATTE-2*: LATTE with two layers that considers both first-order and second-order meta relations. The pairwise proximity preserving objectives is excluded.
- *LATTE-2*$_{prox}$: *LATTE-2* but additionally optimizes the higher-order proximity preserving objectives.

Every method was evaluated on the identical split of training, validation, and testing sets for fairness and reproducibility. The final model is trained only on the training set until the early stopping criteria on the validation set is met, then evaluated on the test set. Additionally, each method must exploit all relations and the available node attributes in the dataset, except for *metapath2vec* due to its limitation. If a particular node type in the heterogeneous graph is not attributed, we instantiate a set of learnable embeddings to replace $\mathcal{X}$ as node features.

#### 5.5.2.2 Implementation Details

We set the following hyper-parameters identically for all methods: embedding dimension size at 128, learning rate at 0.001, mini-batch size at 2048, and early stopping if the validation loss doesn't decrease after five epochs. For HAN, GTN and HGT, the number of GNN hidden layers is 2, followed by an MLP that predicts node labels given the embedding outputs in an end-to-end manner. For random walk-based methods, a separate logistic classifier is employed to perform node classification given the learned node embeddings. The hyper-parameters for metapath2vec and HIN2Vec are walk length at 100, window size at 5, walks per node at 40, and the number of negative samples at 5. Among GNN-based methods, the batch sampling procedure that recursively samples a fixed number of neighbor nodes [59] is utilized, with neighborhood sample sizes 25 and 20. Where possible, the standard implementation of baseline methods has been provided by the CogDL Toolkit.

For all LATTE variants, the best performing hyper-parameters selected ReLU as the embedding activation function, drop-out at 30% on the embedding outputs, and weight decay regularization (excluding biases) at 0.01. In LATTE-$2_{prox}$, the negative sampling ratio is set to 5.0. The models have been implemented with Pytorch Geometric (PyG), and the experiments have been conducted on a GeForce RTX 2080 Ti with 11 GB of GPU memory. The hyper-parameter tuning were conducted by Weight and Biases [10], and the parameter ranges tested were reported in the supplements.

### 5.5.3 Node Classification Experiment Results

We consider the semi-supervised classification tasks in both inductive and transductive settings to perform thorough evaluations of representation learning in heterogeneous graphs. In the transductive setting, models can traverse on the subgraph containing nodes in the test set during training. In contrast, the inductive setting requires the models never to encounter the test subgraph during the training phase and must predict testing nodes' labels on the novel subgraph at the testing phase. We train and evaluate all baseline methods to predict test nodes for each transductive and inductive setting over ten trials.

To measure the classification performance of the prediction outputs, we record the precision and recall for each class label to compute the F1 score. Due to the apparent class imbalance in the three datasets, we report only the averaged Macro-F1 score, which was the more challenging metric in similar experiments [135]. The performance comparisons are reported in Table 5.2. For metapath2vec, HIN2Vec, HAN, and GTN, the benchmark Macro F1 scores in the transductive setting has been provided by the CogDL Toolkit, while the Macro F1 in the inductive setting are averaged scores over 10 experiment runs.

The top performance by LATTE-$2_{prox}$ indicates its effectiveness at learning node representations on the high-order meta relation structures, especially with 80-90% of the graph set aside for testing. Compared to HAN and HGT, which does not consider higher-order relations, GTN and LATTE-2 have a significant edge in inductive prediction because both can capture global properties. Compared to GTN, which does not maintain the semantic space of individual meta path, LATTE-$2_{prox}$ outperforms with explicit proximity-preserving objectives for each of the decomposed higher-order meta relations. Additionally, since GTN necessarily assumes the feature distribution and representation space of different node and link types to be the same, thus it cannot weigh the importance of each meta path separately for each node type. It can also be observed from the total model parameters size, that LATTE's model complexity is comparably less than the GNN baselines. While LATTE must allocate an exponential number of relation-specific attention kernels as $r$ increases, each kernel is only a 1-D vector of the embedding size.

### 5.5.4 Clustering Experiment Results

To show the robustness of the proposed method, we also conduct clustering comparison analysis with the baseline approaches. For each dataset, we train the methods with only the training subgraph, then predict full-graph node embeddings in one batch via feed-forward. We used K-Means to perform node clustering on the embeddings with the same number of clusters as the number of classes. To measure the quality of clusters, we compared them with the ground-truth node labels to compute the NMI score. Since K-Means' performance is affected by its random initial centroid, at each run K-Means is repeated 10 times to yield the average NMI score.

In figure 5.4, we observed LATTE-2 can consistently perform better HAN and GTN, however was only outperformed by HGT in the DBLP dataset. This result shows that in datasets for which has a significant portion of the graph hidden during training, methods that defined separate node- and relation-type dependent encoders can yield more meaningful representation for unseen nodes with a known type. Based on the clustering analysis, we can find that the proposed LATTE can achieve a significant improvement and give a better node representation of the full heterogeneous graph despite missing data.

### 5.5.5 Interpretation of the Attention Mechanism

LATTE's fundamental properties are the construction of higher-order meta relations and the attention mechanism that weighs the importance of those relations. To demonstrate these features' benefits, we interpret the importance levels chosen for each meta relations and verify whether they reflect the structural topology in the heterogeneous graph. Given the learned weights $\boldsymbol{\beta}^{r,i}$ for each node $i$ at a layer $r$, we can assess not only the averaged meta relation weights for a node type, but also the individual meta relation weights for each node. In Fig. 5.3, we report the average and standard deviation of the meta relation attention weights for IMDB, DLBP, and ACM. The correlation between those weights and the node degrees for each relation.

For IMDB movies, it can be observed that on average, the *MA*, *MD*, *MDM*, and *MAM* meta relations have the highest attention weights. This indicates that information from the *movie-actor* neighborhoods, *movie-director* neighborhoods, and node's features are relatively more represented in each *movie*'s first-order embedding. This selection also persists in the second-order embeddings, where *MDM* and *MAM* have higher weights. Additionally, when looking at the correlation between *MA*'s weights and the degree of *MA* links over all nodes, there is a 0.73 correlation, which

indicates the attention mechanism can adaptively weigh the relation based on the number connections present in the node. Interestingly, there is a substantial negative correlation of $-0.88$ between the $M$ "self" relation weights and the node degree. This fact indicates that nodes with fewer or no links will choose a higher weight for its own features, since little information can be gained from other modalities. As individual nodes may have varying levels of participation among the various relations, this result demonstrates that LATTE can select the most effective meta relation for individual nodes depending on its local and global properties in the heterogeneous topology.

## 5.6 Ablation Study

The core components in LATTE are the mechanisms for meta relation weighing, node-level attention weighing, and concatenation of lower- and higher-order embeddings. To assess the effectiveness of these components, we perform an ablation study to disable each single component with these variants:

- *LATTE-2* ($-\alpha$): *LATTE-2* with node-level attention disabled such that all neighbors have the same weight.
- *LATTE-2* ($-\beta$): *LATTE-2* with adaptive relation weighing disabled, such that relations have the same weight.
- *LATTE-2* ($-\tau$): *LATTE-2* without adaptive "sharpening" coefficient in node-level attention, and LeakyReLU activation is instead used as in GAT.
- *LATTE-2* ($-concat$): *LATTE-2* returns only the highest-order embeddings without stacking the $1, ..., R$-order embeddings in equation (7).

We conducted inductive node classification experiments five times for each variant, repeated for each of the three datasets. Shown in figure 5.5, we observed that up to 2%, 7%, or 19% of macro F1 is reduced on average when node-level attention, relation weighing or attention score "sharpening" are disabled, respectively. Most

noticeably, if higher-order embeddings are not stacked (i.e. concatenated) with lower-order embeddings, but rather passed directly to downstream tasks as commonly used in deep GNNs, it suffers a drastic 38% reduction in accuracy. These results demonstrate the effectiveness of the combination of the attention-based components, but also highlights the importance of layer-stacking embedding for higher-order relations.

## 5.7 Discussion

The task of aggregating heterogeneous relations remains a fundamental challenge in designing a representation learning method for heterogeneous graphs. As multiple relations can represent different semantics, their link distributions can be overlapping, interconnected, and/or non-complementary. Therefore, we argue it is an appropriate first step to consider them as separate components of the graph to unravel their structural dependencies.

One of the key differences between existing GNN methods and the proposed LATTE is that the latter exploits the semantic information in the meta relation to reduce the computation complexity of aggregating multi-relations. Instead of conflating heterogeneous relations for all node types as in HAN and GTN, LATTE aggregates only the relevant relations for each node type. Furthermore, by considering the source type and target type of each meta relation, only relevant pairs of relations can be joined when generating higher-order meta paths. A significant benefit to this approach is that it relieves the computational burden of multiplying adjacency matrices for all nodes while allowing distinct representation spaces for the different node types. On the other hand, GTN can be computationally expensive, since it requires computations involving the adjacency structure of all node types at once.

## 5.8 Conclusion

This work has proposed an architecture for heterogeneous graph embedding, which can generate higher-order meta relations. The benefits of the mechanism proposed are not only to improve inductive node classification performance but also to improve interpretation of deep GNN models. In the future, we will explore the possibility to incorporate a self-attention mechanism to learn the structural dependencies between relations by propagating information between lower- and higher-order meta relations. Other interesting future developments are to enable LATTE to pre-train without supervision, to efficiently generate higher-order graph structures during the graph sampling procedure, and to extend LATTE to link prediction tasks.

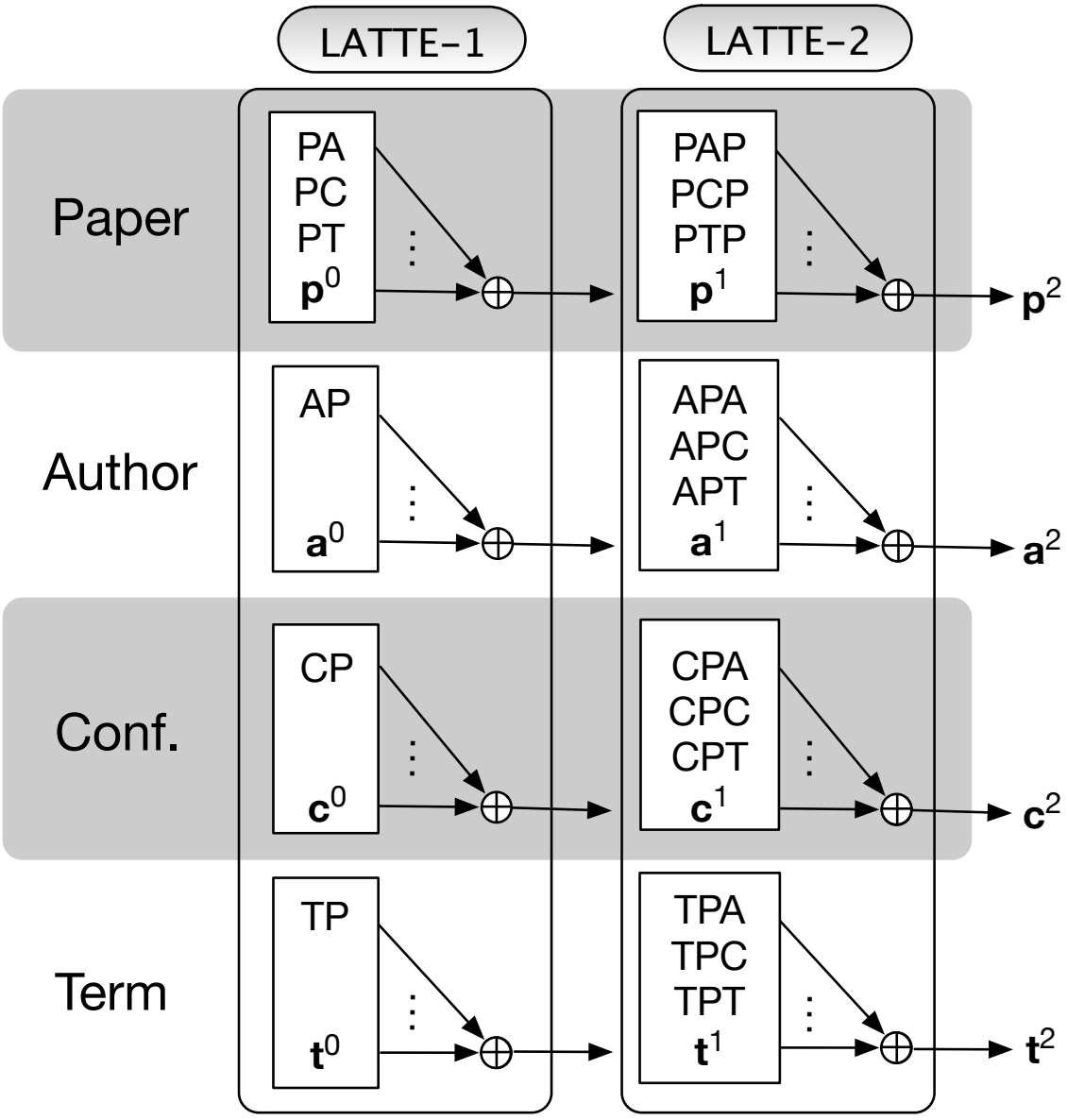
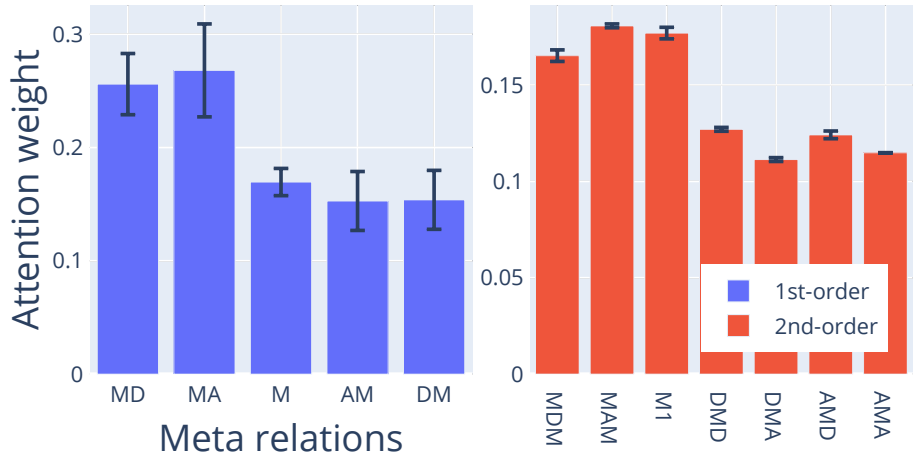
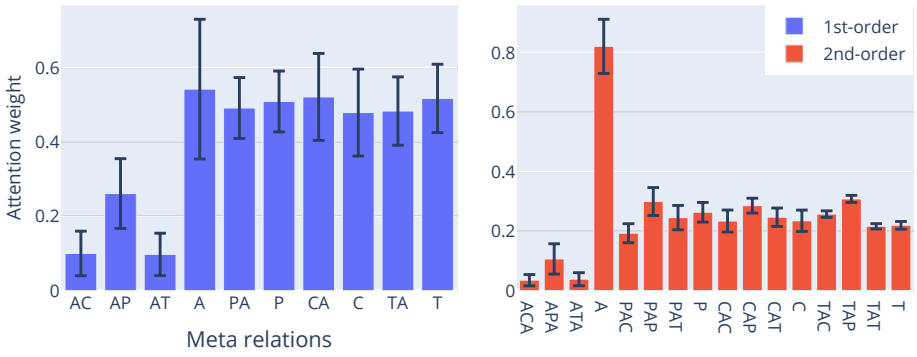


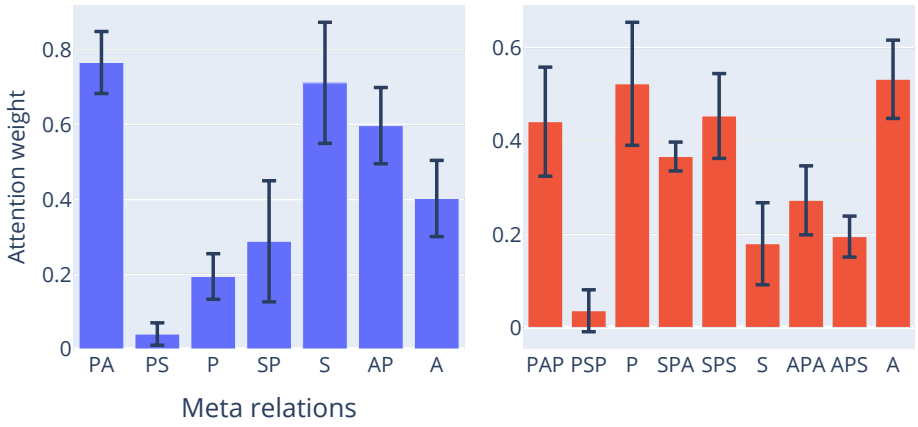

Figure 5.1: Conceptual illustration of the LATTE architecture demonstrating the layer-stacking operations that aggregates first-order and second-order meta relations. The heterogeneous graph contains Paper-Author (PA), Paper-Conference (PC) and Paper-Term (PT) relations and their reverse relations (i.e. AP, CP, TP). The node feature inputs for each node types are $\mathbf{p}^0$, $\mathbf{a}^0$, $\mathbf{c}^0$, and $\mathbf{t}^0$, and the LATTE-$t$ embedding outputs for each respective node types are $\mathbf{p}^r$, $\mathbf{a}^r$, $\mathbf{c}^r$, and $\mathbf{t}^r$.




(a) IMDB

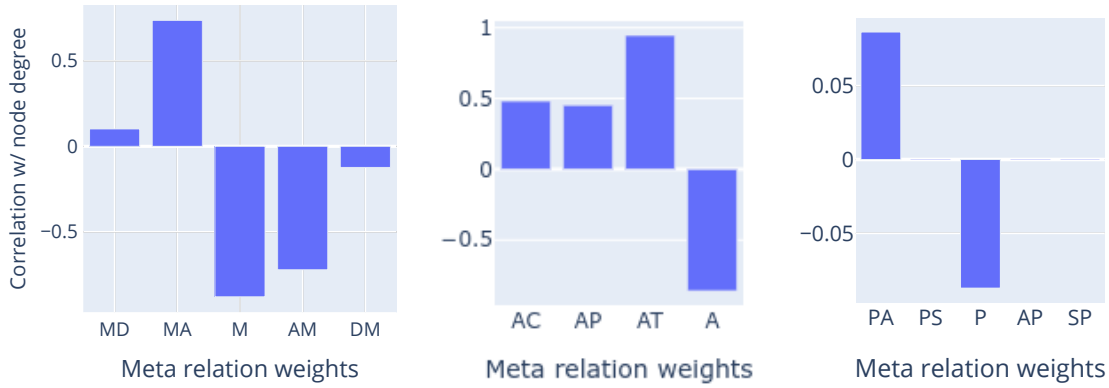



(b) DBLP



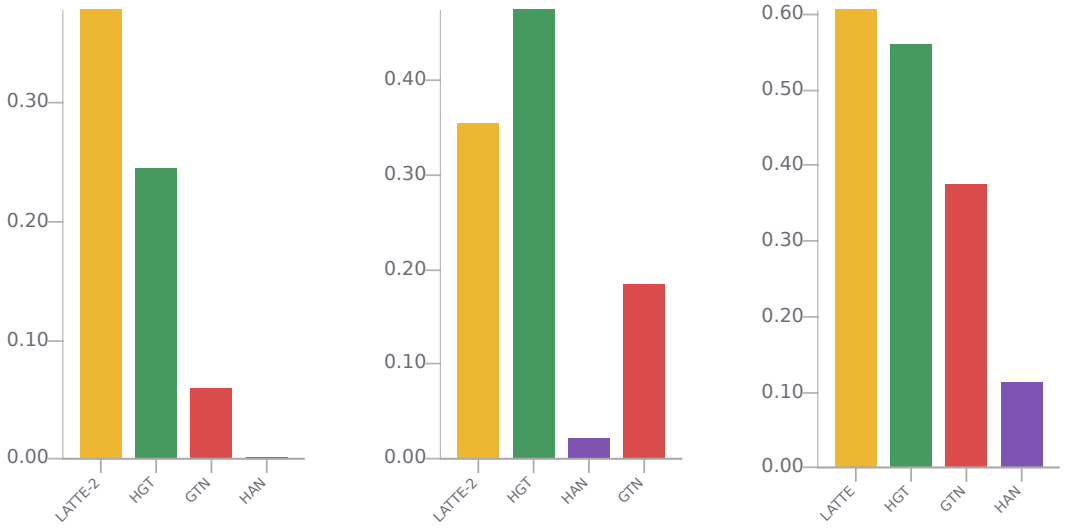

(c) ACM

Figure 5.2: Average and standard deviation of the 1st and 2nd-order meta relation attention weights over each node types. A single-letter relation (e.g. *M*, *M1*) denotes the "self" choice.




(a) IMDB (b) DBLP (c) ACM

Figure 5.3: Correlation between nodes degrees and relation weights for each first-order meta relationLA the three datasets.




(a) NMI on IMDB (b) NMI on DBLP (c) NMI on ACM

Figure 5.4: Clustering results showing the normalized mutual information score across three datasets in the inductive setting.

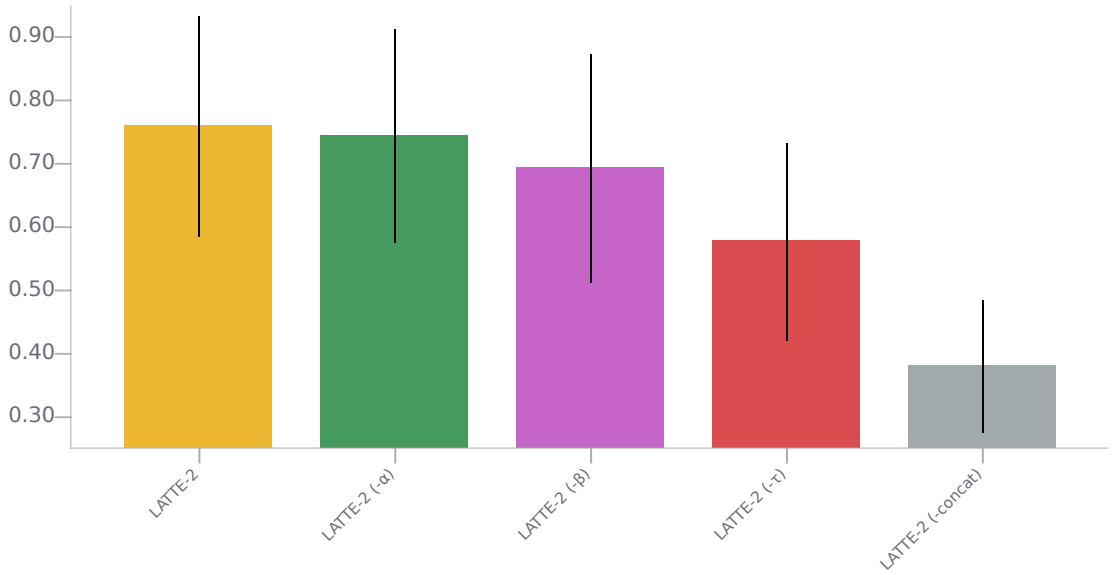



Figure 5.5: Ablation study measuring across 3 datasets. Each bar measures the average and standard deviation of Macro F1 (test) scores across a total of 15 runs.

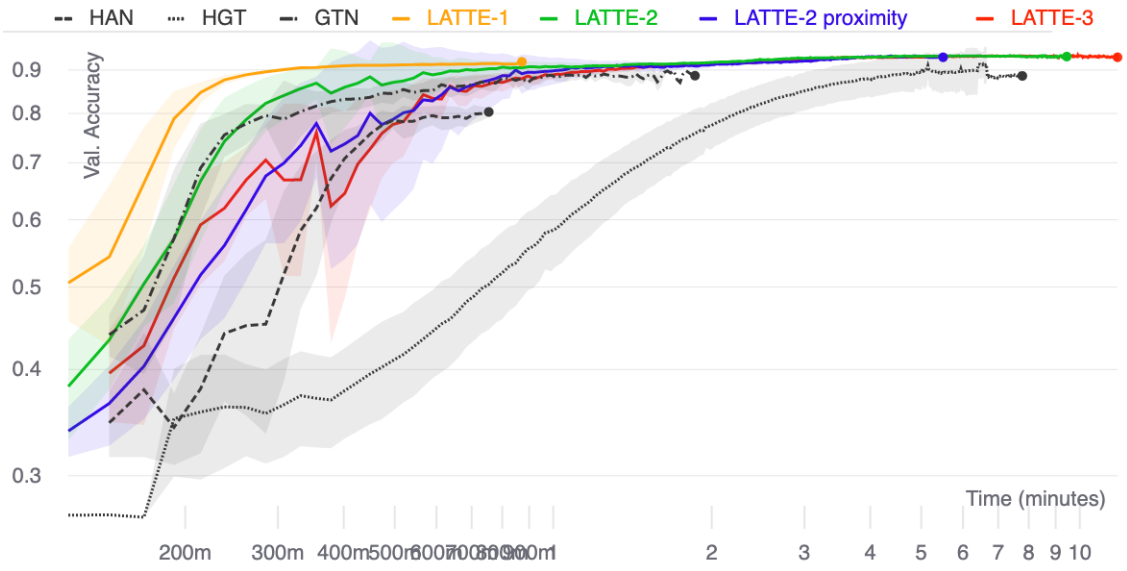



Figure 5.6: Accuracy v. training time on the ACM inductive dataset. Each line shows the mean and its surrounding area shows the standard deviation over 10 runs. Runs were stopped early when the accuracy on the validation set doesn't improve after 10 epochs.

# CHAPTER 6

# PROTEIN FUNCTION PREDICTION BY INCORPORATING KNOWLEDGE GRAPH REPRESENTATION OF HETEROGENEOUS RNA AND PROTEIN INTERACTIONS WITH GENE ONTOLOGY

## 6.1 Abstract

Protein Automatic Function Prediction (AFP) is a large-scale computational prediction problem between proteins and Gene Ontology (GO) terms where most of the identified protein sequences are not fully annotated. Many of the current approaches have resulted in higher AFP accuracy by incorporating protein-protein relationships to computationally infer functions given both sequence- and network-based features. Although a variety of methods have been developed to incorporate homogeneous protein-protein interactions (PPI), none have explored the integration of multi-omics interactions between genes, transcripts, proteins, as well as the Gene Ontology as an integrated heterogeneous graph. By learning representations for both proteins and GO terms in the same model, we developed LATTE2GO, a heterogeneous graph neural network designed to extract higher-order relationships from heterogeneous neighborhood structures. We trained the message-passing neural network model with DistMult to score and rank positive protein-GO term annotations higher than non-existent annotations, which is shown to be as effective as the node classification scheme typically employed for AFP. Experiments were conducted on benchmark datasets according to the CAFA4 protocol for multi-species proteins with the time-based splitting of experimentally-validated annotations. LATTE2GO achieved state-of-the-art performance in Fmax and AUPR metrics compared to recent graph

deep learning AFP methods, with a significant gain in the larger biological processes ontology. With extensive experimental analyses and visualizations, this architecture demonstrates the attention mechanisms that may uncover clues into the effect of specific protein-protein relationships in gene functions.

## 6.2 Introduction

Proteins are responsible for nearly all molecular functions as the build blocks of life [51]. To achieve a more comprehensive understanding of biology, elucidating protein functions are among the most important biological problems. Despite the tremendous growth of identified protein sequences due to the advent of next-generation sequencing technologies, functional annotations for the vast majority of proteins still remain partly or completely unknown. Therefore, *in silico* prediction of protein functions, known as automatic function prediction (AFP), have been widely considered to be promising in the task of predicting or inferring missing functional annotations, where biochemistry experiments are in short supply due to time, cost, and expertise [109].

The AFP task was formulated by a systematic blind prediction challenge named the Critical Assessment of Functional Annotation (CAFA) [72, 163], which provide benchmark datasets for experimentally-validated protein-functional associations. The functions are standardized by the Gene Ontology (GO) [8], which classifies protein and gene functions into hierarchically related functional classes[1] organized into three ontologies: Molecular Function (MF), Biological Process (BP), and Cellular Component (CC). The AFP prediction task is an imbalanced multi-label classification problem on these sets of terms. Effective representation of individual GO terms remains challenging, as there are over 44,000 unique GO functions with "is_a", "part_of",

[1]We interchangeably use the terms protein functions, GO terms, and function classes.

"up_regulates", "down_regulates", etc., relationships between the MF, BP, and CC terms [91].

Aside from extracting effective representations for GO terms, another challenge is learning protein representations to encapsulate their function. Protein interactomics is shedding new light on how protein functions through protein complexes involved in biochemical pathways [124]. Previously, the classical view of protein function focused entirely on the structure and action of a single protein molecule [14]. The more holistic view in protein interactomics considers that each protein plays a role in an extended network of interacting molecules, where a protein's function is the context of its interactions with other proteins. The effectiveness of this approach is signified in recent works that have shown frameworks that integrated protein features from multi-modal data types (e.g., sequence, structure, PPI, etc.) [57, 157] and interaction networks [136] are more likely to outperform the ones that rely on a single datatype. Furthermore, we hypothesize that the consideration of multi-omics interactome in the context of proteins would also better reveal associations to protein functions, as complex biological events usually involve the interplay of genes, transcripts, and proteins [95]. Since there are no direct interactions between certain types of RNAs and proteins, it is imperative to consider indirect multi-hop relationships to sufficiently characterize the RNA-protein interactome.

To address the complex hierarchical structures of GO terms and the multi-omics interactions around proteins, we aim to aggregate the GO structure and the multiple genes-transcripts-proteins interactions as a knowledge graph containing heterogeneous relationships. We propose a method based on graph neural networks [80, 114, 130, 135], which provides a message-passing approach to extract information from the graph structures among RNAs and proteins, and the hierarchical relationships among GO terms. Our method combined multiple data sources, including

sequence features and interaction networks. More specifically, we built a network of Protein, MessengerRNA (mRNA), MicroRNA (miRNA), and Long non-coding RNA (LncRNA) heterogeneous interactions, where each protein or RNA is associated with a sequence. The our method, LATTE2GO, provides the following contributions:

- Extracting higher-order multi-omics relationships from RNA-protein interactions as well as multi-relational protein-protein associations.
- Representation learning of protein functions from multiple relationships in the hierarchical Gene Ontology within the same message-passing framework.
- Exploring attention graph neural networks to effectively aggregate heterogeneous protein-protein interactions and GO term relationships.

Our method allows functional properties of proteins to be inferred by extracting information from complex, large-scale heterogeneous interaction networks.

## 6.3 Related work

Several graph-based methods incorporating protein network data have been dedicated to the AFP problem [96, 134]. Notably, You et al. [150] proposed DeepGraphGO, an end-to-end model consisting of two GCN layers [80], which incorporates sequence-based protein features and the protein-protein network from STRING database [124]. Additionally, DeepFunc [157] proposed to use protein sequence data and DeepWalk to learn protein representations from the combined PPI network from STRING and BioGRID [25]. Several shortcomings of these methods are that: (1) STRING PPI are treated as homogeneous interactions and does not differentiate between the types of protein-protein associations, and (2) physical interactions and genetic interactions are conflated into one protein-protein graph where it may be more beneficial to represent the latter among MessengerRNAs. We hypothesize that more information can be extracted from protein networks by retaining the semantic infor-

mation of specific interaction mechanisms through heterogeneous graph structures. The recent development of multi-relational and heterogeneous graph neural networks [114, 135, 65] can provide message-passing aggregations for the multi-relations among GO term and proteins. However, they have not been extensively explored in the AFP literature, nor have capabilities to generate higher-order relations in multi-omics data.

There are also many proposed AFP methods to extract information from the hierarchical structures of GO terms to improve protein function prediction. DeepGOZero [86] was recently proposed to learn GO term representations through ontology-derived axiom constraints in the n-ball space to enable zero-shot predictions. Yu et al. [151] propose a method called HashGO to explore the underlying structure between GO terms to predict the association between massive GO terms and proteins accurately. Zhou et al. [162] developed DeepGOA with a GCN that can employ the knowledge graph of GO to boost the performance of protein function prediction. While these methods can effectively extract information from the GO knowledge graph, our method has the advantage of learning features directly from the complete hierarchical ontology and connecting with protein network relations in an end-to-end manner.

## 6.4 Materials and methods

### 6.4.1 Data integration

#### 6.4.1.1 Heterogeneous RNA-RNA and protein-protein networks

We construct an integrated graph containing multiple interactions and relationship types between various biomolecule types. We collect multiple networks from experimentally-validated public interaction databases to consider the interaction and relationship networks between various RNA types and proteins. In this section, we list

the databases utilized for training and evaluating our model and the criteria to select specific interaction/relationship types to be integrated. In all databases, we harmonized all miRNA, lncRNA, and mRNA transcript names[2] to standardized MirBase v22 transcript name, Ensembl transcript ID, and HGNC gene names, respectively. For proteins, we index the sequences by the UniProt protein ID.

- **microRNA-mRNA interactions** obtained from miRTarBase version 9.0[66] database, which has a total of 414,828 directed interactions matched between 4,115 microRNAs and 21,943 target mRNAs. We also include microRNA-mRNA interactions from TarBase [76], which includes 966,000 interactions between 1,729 microRNAs and 34039 mRNAs.
- **microRNA-lncRNA interactions** obtained from DIANA-lncBase Experimental v3 [77], containing a total of 64,943 matched directed interactions between 1411 miRNAs and 7103 lncRNAs. We also include RNAInter's miRNA-lncRNA interactions [75], which resulted in 72,261 interactions between 1532 matching lncRNAs and 2701 mRNAs.
- **lncRNA-protein interactions** containing lncRNA-protein directed interaction from RNAInter [75], which contain a total of 12,082,426 interactions between 1037 lncRNAs and 326914 proteins.
- **mRNA-Protein relationships** to represent the one-to-many mapping between mRNAs and proteins. We use the "gene_name" attribute of UniProtKB/Swiss-prot annotation, which results in 227,972 directed relationships between 199,025 mRNAs and 239,987 proteins.

[2]A subset of transcripts and genes selected with matching species in the protein annotation dataset.

- **mRNA-mRNA interactions** obtained from the BioGRID v3.4 database [25] filtered by genetic interactions, which included more than 313,724 matched undirected interactions among 19,429 mRNAs.
- **Protein-protein networks**: We used version 11.0 of STRING database [124], which covers 24.6 million proteins from 5090 organisms totaling more than two billion interactions, which was generated before Jan. 2019. Here, rather than combining all STRING PPI with a non-zero "combined_score", we separate the different types of protein-protein associations into multiple sub-graphs where an edge exists between two proteins if there is a non-zero score in the respective edge type. Specifically, we obtained 23818564 associations for "co-expression", 724806 associations for "co-occurrence", 11352421 associations for "database", 22013391 associations for "experimental", 61520 associations for "fusion", 2889167 associations for "neighborhood", and 28532031 associations for "textmining" to create seven protein-protein undirected networks.

We used our package OpenOmics [128] to combine all nodes and edges into an integrated graph which contains a total of 86927 lncRNAs, 199025 mRNAs, 98444 microRNAs, and 239987 proteins. When two databases are integrated for the same interaction type, the overlapping edges are counted once.

#### 6.4.1.2 Gene Ontology representation

To holistically represent the Gene Ontology structure and the multiple types of relationships between GO terms, we also construct a heterogeneous graph among the GO terms to integrate with the RNA-protein heterogeneous graph. We downloaded the Gene Ontology [8, 1] in OBO format to extract the relationships and reverse the edge directionality. Depending on whether we are predicting biological process (BP), molecular function (MF), or cellular component (CC) functions, we consider all the

terms in the respective ontology, not just the target functions. The relationship types we selected are "is_a", "part_of", "has_part", "regulates", "positively_regulates", and "negative_regulates", where each type encodes directed interactions, e.g., an edge $i \overset{is\_a}{\rightarrow} j$ where $i$ is a parent term, and $j$ is a child term.

#### 6.4.1.3 Protein features

For each protein $i$, we generate its feature vector $\mathbf{x}_i$ from InterProScan [94] by extracting the count of InterPro signature matches in the sequence. Specifically, $\mathbf{x}_i \in \mathbb{Z}^m$ is a sparse vector where $m$ is the number of unique family, domain, and motif entries totaling 40,597 as of InterPro Release 90.0 [103]. We apply a memory-efficient row-sparse matrix multiplication to obtain a low-dimensional vector representation $\mathbf{h}_i^{(0)} \in \mathbb{R}^d$ with:

$$\mathbf{h}_i^{(0)} = \textsc{ReLU}\left(\mathbf{W}^{(0)}\mathbf{x}_i + \mathbf{b}^{(0)}\right) \tag{6.1}$$

where $\mathbf{W}^{(0)} \in \mathbb{R}^{d\times m}$ and $\mathbf{b}^{(0)} \in \mathbb{R}^d$ are learnable weights and biases, and $\textsc{ReLU}$ is the Rectifier Linear Unit activation function. As inputs to our model, only the proteins' vector representations are extracted from feature attributes, whereas other RNA types and GO terms use randomly initialized learnable embeddings of the same dimension size $d$.

### 6.4.2 LATTE2GO GNN architecture

The overall architecture of Layer-stacked ATTention Embedding to Gene Ontology (LATTE2GO) model is illustrated in Figure 6.1. Given pairs of protein and GO term nodes in the integrated knowledge graph, the model aggregates all heterogeneous relations up to $k$-hops around these "seed nodes". The goal of LATTE2GO is to aggregate information from each meta relation-specific neighborhood around the

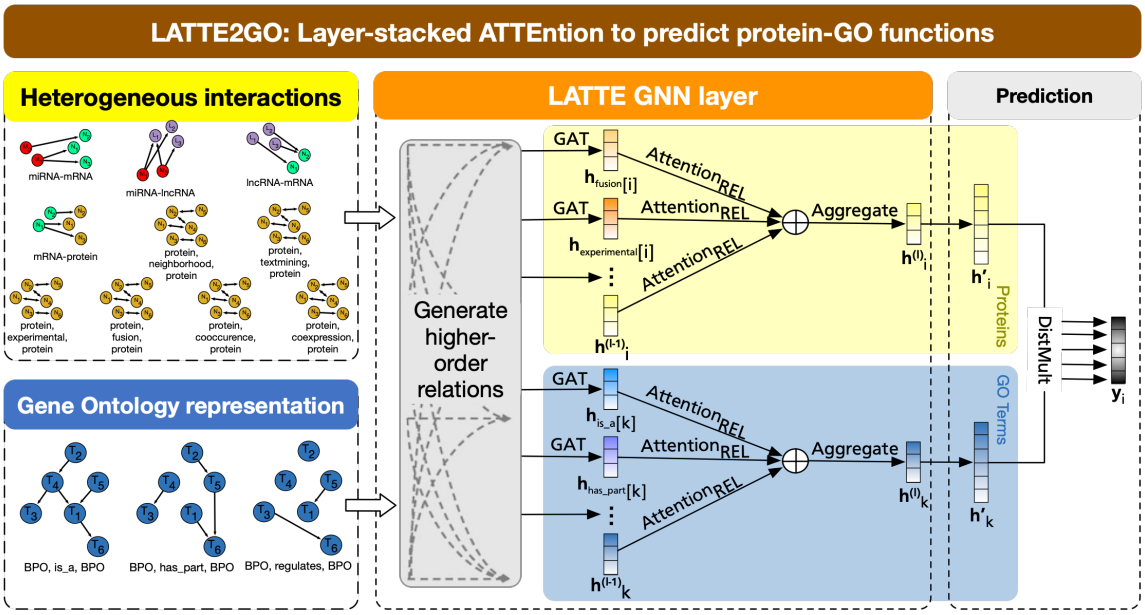



Figure 6.1: LATTE2GO architecture diagram.

seed nodes, then aggregate their contextualized representations with attention. Given the node representations of the gene and GO term nodes, a scoring function is used to determine the strength of the connection between the protein-function pair.

### 6.4.2.1 Heterogeneous graph representation

We encapsulate the various entities and relationships into a heterogeneous directed graph $G = (\mathcal{V}, \mathcal{E}, \mathcal{T}, \mathcal{A})$ in which each node $i \in \mathcal{V}$ and each edge $e_{ij} \in \mathcal{E}$ are associated with their entity and relation type mapping function $\tau(i) : \mathcal{V} \rightarrow \mathcal{T}$ and $\phi(e_{ij}) : \mathcal{E} \rightarrow \mathcal{A}$, respectively. Generally $|\mathcal{T}| + |\mathcal{A}| \geq 2$, where $\mathcal{T}$ and $\mathcal{A}$ denote the sets of node and relation types.

**6.4.2.1.1 Meta relations.** For a directed edge $e_{ij}$ linking source node $i$ to target node $j$, its meta relation is denoted as $\langle \tau(i), \phi(e_{ij}), \tau(j) \rangle$. Thus, the set of all heterogeneous meta relation types is defined as $\mathcal{A} = \{\langle s, r, t \rangle \mid s, t \in \mathcal{T}\}$, where $r$ can

denote $\langle s, r, t \rangle$ interchangeably for brevity. Note that there may exists more than one unique meta relation between the same source node type and target node type. We additionally denote the subset of edges with relation type $r$ as $\mathcal{E}_r = \{e_{ij} \mid \phi(e_{ij}) = r\}$.

**6.4.2.1.2 Higher-order meta relations.** Given $\mathcal{A}$, we additionally define the higher-order meta-relations set $\mathcal{A}^{(l)}, l \geq 1$, which contain $l$-hop metapaths as a sequence of $l$ meta relations, as follows:

$$\mathcal{A}^{(l)} = \{\langle u, w \circ r, t \rangle \mid v = s, \langle u, w, v \rangle \in \mathcal{A}^{(l-1)}, \langle s, r, t \rangle \in \mathcal{A}\} \tag{6.2}$$

where $\mathcal{A}^{(1)} = \mathcal{A}$ and $\circ$ denotes composition operator. Thus, the new edge set induced by a composed meta relation $r_c = r_a \circ r_b$ is defined as $\mathcal{E}_{r_c} = \{e_{ik} \mid \phi(e_{ik}) = r_c, e_{ij} \in \mathcal{E}_{r_a}, e_{jk} \in \mathcal{E}_{r_b}\}$.

**6.4.2.1.3 Knowledge graph preprocessing.** To model the ground-truth heterogeneous graph structure consisting of LncRNA, MicroRNA, MesssengerRNA, protein and GO term node types, we process various undirected and directed edges contained in the aforementioned databases as either undirected or directed meta relations. For undirected meta relations $r_u \in \mathcal{A}$, such as $\langle \textit{Protein}, \textit{experimental}, \textit{Protein} \rangle$, we ensure $e_{ji} \in \mathcal{E}$, $\forall e_{ij} \in \mathcal{E}$ where $\phi(e_{ij}) = \phi(e_{ji}) = r_u$. Additionally, for each directed meta relations $r_d \in \mathcal{A}$, we inject a separate "reverse" relation $r_d^{-1}$ into $\mathcal{A}$ and its reverse edges $\{e_{ji} \mid \forall e_{ij}$ where $\phi(e_{ij}) = r_d$ and $\phi(e_{ji}) = r_d^{-1}\}$ into $\mathcal{E}$, e.g. $\langle \textit{BP}, \textit{is\_a}, \textit{BP} \rangle^{-1} = \langle \textit{BP}, \textit{rev\_is\_a}, \textit{BP} \rangle$. This preprocessing step ensure messages can be propagated between every node types, while still preserving the meta-relation's directed/undirected semantics.

#### 6.4.2.2 Layer-stacked attention on meta-relations

Since there exist multiple relations connected to proteins and GO terms, we experiment with the idea that attention mechanisms are suitable to identify salient relations which contains the necessary information for classifying protein-function relationships. We apply the message-passing GNN framework [50], and propose our model LATTE2GO extending on the work of [135], with the goal of organizing messages from relation-specific neighborhoods into separate contextualized embeddings. At the $(l)$-th LATTE layer where $1 \leq l \leq L$, each node $i$'s representation $\mathbf{h}_i^{(l)} \in \mathbb{R}^d$ updates its state by aggregating context embeddings from multiple relations with its own representation, as follows:

$$\begin{aligned} \mathbf{h}_i^{(l)} &= \underset{\forall r \in \mathcal{A}_{\tau(i)}^{(l)}}{\textbf{Aggregate}} \left( \textbf{Attention}_{\text{REL}}\left(r, i\right) \cdot \mathbf{h}_{\mathbf{r}i}^{(l)} \right) \\ \mathcal{A}_{\tau(i)}^{(l)} &= \{\langle s, r, t \rangle \in \mathcal{A}^{(l)} \mid t = \tau(i)\} \cup \{\langle \tau(i) \rangle\} \end{aligned} \tag{6.3}$$

where $\mathbf{h}_{\mathbf{r}i}^{(l)} \in \mathbb{R}^d$ represents node $i$'s context embedding from relation $r$, and $\mathcal{A}_{\tau(i)}^{(l)}$ contains all $l$-hop meta relations with the target node type $\tau(i)$, including the "self" node type $\langle \tau(i) \rangle$ to represents the self-connection. Note that we're able to aggregate meta-relations from multiple source types to each target type, thus not constrained by predefined metapaths where $s = t$ [135].

Since the target node type $t$ is assumed to be involved in multiple relations types, multiple relation-specific representations are aggregated for each node. More specifically, our self-attentional $\textbf{Attention}_{\text{REL}}$ function adaptively infer the relation at-

tention coefficients individually for each target node $i$, which enables the capacity to capture individual node heterogeneity in the graph. It is defined as:

$$\begin{aligned}
\textbf{Attention}_{\text{REL}}\left(r,i\right) &= \underset{\forall r \in \mathcal{A}^{(l)}_{\tau(i)}}{\text{Softmax}}\left(\beta(r,l,i) + \mu_r\right) \\
\beta(r,l,i) &= {\mathbf{b}^{(l)}_r}^{\top} f\left(\left[\mathbf{h}^{(l)}_{\mathbf{r}_i} \| \mathbf{h}^{(l)}_{\langle \tau(i) \rangle_i}\right]\right) \\
\mathbf{h}^{(l)}_{\langle \tau(i) \rangle_i} &= \mathbf{W}^{(l)}_{\tau(i)} \mathbf{h}^{(l-1)}_i
\end{aligned} \tag{6.4}$$

where $\mathbf{b}^{(l)}_r \in \mathbb{R}^{2d}$ and $\mu_r$ is the trainable attention vector and bias scalar for relation $r$, $\mathbf{W}^{(l)}_{\tau(i)} \in \mathbb{R}^{d \times d}$ is the trainable weight matrix for node type $\tau(i)$, $\|$ denotes the concatenation operator, and $f$ is the $\text{LeakyReLU}_{\alpha=0.2}$ activation function.

Given the context embeddings and their predicted relation attention scores which sums up to 1, the aggregation step combines them with a weighted summation. Since Velickovic et al. [131] have shown attention learning is more stable with multi-head attention [129], we employ $H$ separate attention heads to concatenate their outputs, as follows:

$$\underset{\forall r \in \mathcal{A}^{(l)}_{\tau(i)}}{\textbf{Aggregate}}(\cdot) = \text{LayerNorm}\left(\text{ReLU}\left(\Big\|_{h=1}^{H} \sum_{\forall r \in \mathcal{A}^{(l)}_{\tau(i)}} (\cdot)\right)\right) \tag{6.5}$$

where if $H > 1$, then $\mathbf{h}^{(l)}_i, \mathbf{h}^{(l)}_{\mathbf{r}_i}$, and all parameters are separate for each head $h$ and have its hidden dimension size divided by $H$.

Given that each layer $l$ output node representations that contain context information aggregated only from $l$-hop meta relations, we use several layers to compute to up $L$-hop meta relations. The final embedding for node $i$ is obtained by stacking $\mathbf{h}^{(l)}_i$ from the outputs of $L$ layers, as follows:

$$\mathbf{h}'_i = \Big\|_{l=1}^{L} \mathbf{h}^{(l)}_i \tag{6.6}$$

where $\mathbf{h}'_i \in \mathbb{R}^{dL}$, which can be used for end-to-end training with downstream tasks.

#### 6.4.2.3 Graph attention network

GATs have shown to be powerful for inductive protein function classification from protein-protein interaction network data [131]. Here, we leverage the masked self-attention proposed in GAT and compute the relation-specific context embedding $\mathbf{h}_{\mathbf{r}i}^{(l)}$ of node $i$ with:

$$\mathbf{h}_{\mathbf{r}i}^{(l)} = \sum_{j \in \mathcal{N}_r(i)} \textbf{Attention}_{\text{GAT}}\left(j, r, i\right) \cdot \textbf{Message}\left(j\right) \tag{6.7}$$

where $\mathcal{N}_r(i) = \{j \mid e_{ji} \in \mathcal{E}_r\}$ contain incoming neighbors of target node $i$ in relation $r$, without the self-loop. Additionally, we apply the modification proposed in GATv2 [17] with improved expressiveness of the edge-level attention function, defined as:

$$\begin{aligned} \textbf{Attention}_{\text{GAT}}(j, r, i) &= \underset{\forall j \in \mathcal{N}_r(i)}{\text{Softmax}}\left(\alpha(j, r, i)\right) \\ \alpha(j, r, i) &= \mathbf{a}_r^{(l)\top} f\left(\left[\mathbf{W}_{\tau(j)}^{(l)} \mathbf{h}_j^{(l-1)} \| \mathbf{W}_{\tau(i)}^{(l)} \mathbf{h}_i^{(l-1)}\right]\right) \\ \textbf{Message}(j) &= \mathbf{W}_{\tau(j)}^{(l)} \mathbf{h}_j^{(l-1)} \end{aligned} \tag{6.8}$$

where $\mathbf{a}_r^{(l)} \in \mathbb{R}^{2d}$ is the edge-level attention vector for relation $r$, and $f$ is the $\text{LeakyReLU}_{\alpha=0.2}$ activation function. Note that in LATTE2GO, the edge weights that represent interaction strength or confidence level is not utilized, and instead allow GAT to infer edge weights via the attention mechanism.

#### 6.4.2.4 Computing classification scores between proteins and GO terms

To predict functions for protein $i$, a sampled subgraph of the heterogeneous graph is obtained from up-to-$L$-hops neighborhood expansions [59], starting from a "seed nodes" set that includes both protein $i$ and the set of target classes denoted as $\mathcal{V}_{GO}$. Although there are no relations between $i$ and $\mathcal{V}_{GO}$, node representations for proteins and GO terms can be computed simultaneously in the same feed-forward pass of the LATTE layers. Rather than a final linear transform layer to score protein

$i$'s class probabilities in the typical node classification setting, we apply DistMult [146] to score the probability for class $k \in \mathcal{V}_{GO}$, as:

$$\hat{y}_{ik} = \sigma\left(\mathbf{h}_i'^{\top} \mathbf{M} \mathbf{h}_k'\right) \tag{6.9}$$

where $\sigma$ is the sigmoid function and $\mathbf{M} \in \mathbb{R}^{dL \times dL}$ is a trainable diagonal matrix. For semi-supervised node classification learning, we use the binary cross-entropy loss function:

$$\mathcal{L}(\mathbf{\Theta}) = -\frac{1}{|\mathcal{V}_P||\mathcal{V}_{GO}|} \sum_{i \in \mathcal{V}_P} \sum_{k \in \mathcal{V}_{GO}} y_{ik} \log(\hat{y}_{ik}) + (1 - y_{ik}) \log(1 - \hat{y}_{ik}) \tag{6.10}$$

where $\mathbf{\Theta}$ is the set of parameters in all layers to be learned, $\mathcal{V}_P$ are the subset of protein nodes associated with ground-truth labels, and $y_{ik} \in \{0, 1\}$ is the true binary indicator for protein $i$ and function $k$.

### 6.4.3 Model training and implementation details

The LATTE2GO GNN model was implemented with PyTorch-Geometric [46] and can run on a single CUDA GPU with at least 10GB of RAM. To tractably train a graph of 6.6M nodes and 71M edges, we use mini-batch SGD with subgraph sampling [59], along with the heterogeneous nodes subsampling technique HGSampling [65] where the node budget per layer is the same as the batch size. To alleviate the exponentially increasing size of the higher-order relations set $\mathcal{A}^{(l)}$ due to the cartesian product in Eq. 6.2, we enforce: (1) meta relations with identical source and target node types can only compose if they are of the same edge type, and (2) to filter meta-relations at the last layer to have the target node type as the "seed nodes". Additionally, (3) to deal with very dense edges when composing a high-order meta-relation $r_c = r_a \circ r_b$ within a sampled subgraph, we subsample the set of edges in $r_a$ and in $r_c$ such that each target node have approximately $K$ neighbors or less. With these heuristics, the overall worse-case time-complexity of LATTE2GO is

$\mathcal{O}\left(|\mathcal{A}|^L\left(NK^L + NKd\right) + LNd^2\right)$ where $N = |\mathcal{V}|$ and $L \leq |\mathcal{T}|$, and the first term is $|\mathcal{T}|^L$ in the average case. With a dynamic programming implementation, generating the high-order meta-relations is parallelized on CPU and is computed with efficient sparse matrix multiplications.

All hyper-parameters are determined through a grid search based on the model's BPO AUPR performance on the temporal-holdout validation set. The hyper-parameter tuning and orchestration were conducted by Weight and Biases [10]. We use the Adam optimizer [79] with batch size = 2048 and initial learning rate = 0.001. To avoid over-fitting, we use weight decay = 0.01 and early stopping when the validation AUPR rate stops increasing after 5 epochs.

## 6.5 Results

### 6.5.1 Dataset characteristics

We used the protein-GO annotation dataset compiled from DeepGraphGO's benchmark dataset [150], which was built according to the CAFA4 outline. This benchmark dataset contain GO annotations for 239,987 UniProtKB-SwissProt protein sequences [35] with specified training, validation and testing sets based on time splits on before Jan. 2018, Dec. 2018, and Jan. 2020, respectively. When collecting the 'IDA', 'IPI', 'EXP', 'IGI', 'IMP', 'IEP', 'IC', and 'TA' evidence coded annotations set from SwissProt1 [16] and UniProtGOA [69], we added parent terms-propagated annotations for every child term annotations, and replaced all alias GO terms with the canonical GO term name [8]. The sample size characteristics used by all models in our experiments is shown in Table 6.1.

Table 6.1: Sample size characteristics of dataset splits

| Ontology | Terms | Training proteins | Validation proteins | Testing proteins |
|---|---|---|---|---|
| MFO | 6868 | 51549 | 490 | 426 |
| BPO | 21381 | 85104 | 1570 | 925 |
| CCO | 2832 | 76098 | 923 | 1224 |

### 6.5.2 Experimental settings

To generate node classification results on the CAFA4 benchmark dataset, we train and validate methods on each of the Biological Process (BP), Molecular Function (MF), and Cellular Component (CC) ontology independently. All models considered were trained on the same set of training protein-function annotations, early-stopping monitored metrics on the same validation set annotations and evaluated on the same test set annotations. Since we applied the same evaluation protocol in [150], this allows for direct comparison with competing methods evaluated in the article.

#### 6.5.2.1 Baseline methods

Since our model aims to predict embedding representation for both proteins and GO terms, we considered three types of AFP methods for comparison with LATTE2GO: homologous sequence transfer, sequence-only representation learning, and homogeneous PPI GNN-based methods.

- *LR-InterPro*: Given the protein feature vectors $\mathbf{h}_i^{(0)}$ extracted from InterPro features, a linear transform followed by a sigmoid activation is used to compute the GO term class scores.
- *BLAST-KNN*: BLAST is used to obtain a set of homologous proteins for a given query protein sequence, and GO term labels are propagated to the query protein by a similarity score. Its full implementation details are outlined in [150].

- *DeepGOCNN* [85]: A sequence-based model which uses 1D CNN protein sequence encoder followed by a flat multi-label classifier.
- *DeepGOPlus* [85]: A sequence-based model which combines deep convolutional neural network (CNN) model with sequence similarity-based predictions.
- DeepGraphGO [150]: A GNN-based method with two GCN [80] layers, which incorporates the InterPro protein features and the combined protein-protein network information from STRING database.
- *LATTE2GO-1*: The proposed LATTE2GO model with two layers that only considers first-order meta relations, i.e., $\mathcal{A}^{(l)} = \mathcal{A}, l \geq 1$. This model uses only protein-protein and GO-GO relations.
- *LATTE2GO-2*: LATTE2GO with two layers that considers both first-order and second-order meta relations. This model uses only protein-protein and GO-GO relations.

We set the following hyper-parameters identically for all methods: embedding dimension size at 512 and early stopping if the validation loss does not decrease after five epochs. For DeepGraphGO, the number of GNN hidden layers is 2, followed by an MLP that predicts node labels given the embedding outputs in an end-to-end manner. Regarding the mini-batch subgraph sampling for GNN methods, DeepGraphGO uses full-neighborhood expansion at each layer on a k-NN PPI subgraph where $k = 30$, whereas LATTE2GO uses HGSampling [65] on the complete set of interactions.

#### 6.5.2.2 Evaluation metrics

We used two evaluation metrics to compare AFP methods, $F_{max}$ and AUPR (Area under Precision-Recall curve), as used as the primary evaluation metrics in [72]. $F_{max}$ is a protein-centric measure of the maximum F1 score for any thresholds on the classification scores among all GO term classes, averaged over all proteins,

Table 6.2: Performance comparison results of LATTE2GO with DeepGraphGO

| Method | Fmax | | | AUPR | | |
|---|---|---|---|---|---|---|
| | MFO | BPO | CCO | MFO | BPO | CCO |
| LR-InterPro | 0.617 | 0.278 | 0.661 | 0.530 | 0.144 | 0.672 |
| BLAST-KNN | 0.590 | 0.274 | 0.650 | 0.455 | 0.113 | 0.570 |
| DeepGOCNN | 0.434 | 0.248 | 0.632 | 0.306 | 0.101 | 0.573 |
| DeepGOPlus | 0.593 | 0.290 | 0.672 | 0.398 | 0.108 | 0.595 |
| DeepGraphGO | 0.623 | 0.327 | **0.692** | 0.543 | 0.194 | **0.695** |
| LATTE2GO-1 | 0.778 | 0.539 | 0.691 | 0.753 | 0.534 | 0.689 |
| LATTE2GO-2 | **0.840** | **0.574** | 0.683 | **0.831** | **0.584** | 0.682 |

defined as: $F_{max} = max_t \left\{ \frac{2 \cdot pr(t) \cdot rc(t)}{pr(t)+rc(t)} \right\}$, where $pr(t)$ and $rc(t)$ denote the precision and recall obtained at a positive-class threshold value $t$, as defined in [150].

### 6.5.3 Comparison results

Outlined in Table 6.2, we report the $F_{max}$ and AUPR metrics for performance comparison between LATTE2GO and the baseline methods. Only the DeepGraphGO and LATTE2GO have been executed for comparison analysis, whereas the results of other methods were copied from DeepGraphGO's article [150].

*LATTE2GO-1* show a significant performance increase on $F_{max}$ and AUPR for MFO and BPO, compared to DeepGraphGO. Our method also uses InterPro protein features, but the STRING PPI contains multi-relational protein-protein associations, leading to better representations of proteins given its topological networks. We hypothesize that protein representation learning is more effective not only with the separation of physical v.s. genetic protein-protein associations, but also with the specified data source of PPI, which may imply interactions in different biological contexts. Additionally, due to the large size of the BPO target classes, we hypothesize that incorporating the GO multi-relational associations graph leads to better representations of GO terms while providing as good classification performance as the

typical node classification setting. Importantly, representing the Gene Ontology as a graph allows us to infer annotations on sparse or unannotated terms not seen in the training dataset.

*LATTE2GO-2* also show a performance boost compared to *LATTE2GO-1*, signifying the approach of generating higher-order meta relations. Since we have utilized the meta relation semantics to connect higher-order relations, this allows for extracting semantic-specific higher-order structures and exploring possible permutations of meta-relations. In contrast with multi-layer GNNs where each message-passing layer applies on the same first-order graph structure, we argue our approach of generating and aggregating meta relations has two advantages: (1) to decouple the higher-order metapath and retain semantics information in higher-order neighborhoods, and (2) to alleviate the "over-squashing" problem [5], where the higher-order context are combined with the lower-order context across layers. In our experiments, we have identified that up to second-order leads to sufficient performance, motivated by the consensus that neighbor-of-neighbor proteins in the interaction topology are likely to share the same functions [33].

### 6.5.4 Ablation analysis

The core components in LATTE2GO are selecting various node types and interaction types included in the integrated graph, generating higher-order relationships, and concatenating multiple higher-order embeddings. To assess the effectiveness of these four components, we perform an ablation study by changing these model hyperparameters and observe the changes in BPO AUPR metrics on the test dataset of human- and mouse-only proteins. To orchestrate this analysis, we used Weight and Biases [10] to execute a grid search for all of the settings for each component.

Shown in Fig. 6.2, we report the maximum performance achieved by various combinations of node types and edge types and the box-plot of the AUPR distribution on various hyper-parameters on higher-order relations and layer embedding concatenation. In the "Heterogeneous node types" plot, we observed a poorer performance when including all of the RNAs, proteins, and GO node types, and only the protein-only or protein-and-BPO achieved the highest performance. This surprising result suggests that adding multi-omics RNA interactions to proteins does not improve function prediction in LATTE2GO. However, it can also be interpreted that the protein-and-BPO heterogeneous graph can achieve as good of a performance as a protein-only heterogeneous graph, reinforcing our idea of GNN for both proteins and the GO. In the "Split PPI interaction" types plot, we construct the STRING data as either heterogeneous or homogeneous PPIs, reporting the maximum value on the grid search. We observed a significant improvement in AUPR, which supports our hypothesis of multi-relational PPI for more accurate AFP. In the other two experiments, we can also conclude that generating second-order meta relations does improve AUPR, while it is inconclusive whether concatenating layer embeddings improves performance with *LATTE2GO-2*.

### 6.5.5 Interpretation of relation attention scores

LATTE2GO's fundamental properties are the construction of higher-order meta relations and the attention mechanism that weighs the importance of those relations. To demonstrate the interpretability of our attention-based aggregator, we visualize the predicted attention scores for each meta relations to observe the salient meta-relations for protein representation learning. Given the learned weights $\beta(r, l, i), \forall r$ at layer $l$, we can assess the averaged meta relation weights for all nodes $i$ of a node type and the individual meta relation weights for each node. In Fig. 6.3, we report the average meta

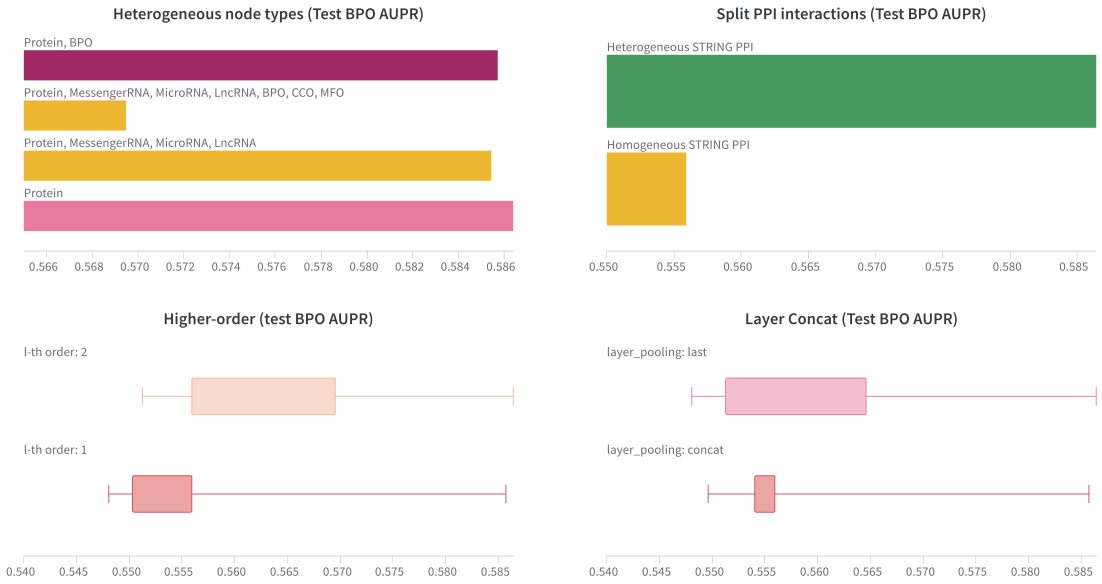



Figure 6.2: Ablation analysis reporting differences on AUPR metric on the node types used in the heterogeneous graph (top left), on separating protein-protein associations in STRING-db (top right), on generating higher-order meta-relations (bottom left), and on whether to concatenate layer embeddings (bottom right).

relation attention weights for *LATTE2GO-2* across two layers. For the BPO nodes, we can observe that the "is_a" and "is_a" ∘ "is_a" have the highest weights, which is expected as this relation defines the hierarchical structure of the GO. For protein nodes, "cooccurrence", "database", "textmining", and "coexpression" contains the most information for predicting protein functions. This result may offer insights for new studies to analyze the salient relationships to characterize protein function. Note that individual nodes can have varying levels of participation in various relations so that LATTE2GO can select the most effective meta relation for individual nodes depending on their local and global properties in the heterogeneous topology.

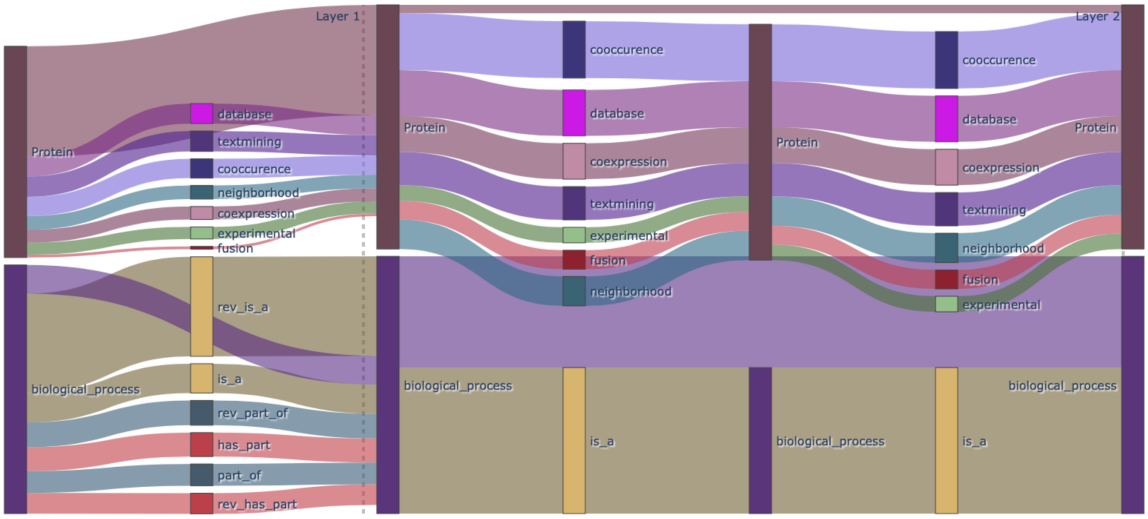



Figure 6.3: Sankey flow plot showing the aggregation of meta-relations and self connections for *LATTE2GO-2* predicting protein-BPO functions. Each block represents either a node type or meta relation, and the links width represent the attention weight in-proportion to other links of the same target node type. The first- and second-order meta relation attention weights were averaged over all nodes of each node types in a subgraph batch.

## 6.6 Conclusion

This paper explored an end-to-end graph neural network framework for automatic protein function predictions in heterogeneous graphs. By approaching this problem with an expressive representation of the protein-protein interactions and GO knowledge graph, our aggregation mechanism can fully utilize the semantic context in the raw data without the manual design of specific features. We believe the versatility of this graph-based approach will enable substantial improvement in AFP by enabling researchers to consider relationships between proteins to other entities, such as the InterPro and Enzyme Commission Ontology, which contain entries related hierarchically. Additionally, with the consideration for second-order relationships among the multi-relations, LATTE2GO demonstrated significantly higher performance than GNNs that only perform first-order message-passing. In future work, we believe this

feature can be further explored to be more effective in the inductive prediction setting, where there are sparse interactions around proteins or non-existent annotations on specific GO terms.

REFERENCES


[1] The gene ontology resource: enriching a gold mine. *Nucleic acids research*, 49(D1):D325–D334, 2021.

[2] Lada A Adamic and Eytan Adar. Friends and neighbors on the web. *Social networks*, 25(3):211–230, 2003.

[3] Vikram Agarwal, George W Bell, Jin-Wu Nam, and David P Bartel. Predicting effective microrna target sites in mammalian mrnas. *elife*, 4:e05005, 2015.

[4] Reka Albert. Scale-free networks in cell biology. *Journal of cell science*, 118(21):4947–4957, 2005.

[5] Uri Alon and Eran Yahav. On the bottleneck of graph neural networks and its practical implications. *arXiv preprint arXiv:2006.05205*, 2020.

[6] Paulo P Amaral, Michael B Clark, Dennis K Gascoigne, Marcel E Dinger, and John S Mattick. lncrnadb: a reference database for long noncoding rnas. *Nucleic acids research*, 39(suppl_1):D146–D151, 2010.

[7] Ehsaneddin Asgari and Mohammad RK Mofrad. Continuous distributed representation of biological sequences for deep proteomics and genomics. *PloS one*, 10(11):e0141287, 2015.

[8] Michael Ashburner, Catherine A Ball, Judith A Blake, David Botstein, Heather Butler, J Michael Cherry, Allan P Davis, Kara Dolinski, Selina S Dwight, Janan T Eppig, et al. Gene ontology: tool for the unification of biology. *Nature genetics*, 25(1):25–29, 2000.

[9] Federico Battiston, Vincenzo Nicosia, and Vito Latora. Structural measures for multiplex networks. *Physical Review E*, 89(3):032804, 2014.

[10] Lukas Biewald. Experiment tracking with weights and biases, 2020. Software available from wandb.com.

[11] Bishop, J A, Bishop, Justin A, Benjamin, Hila, Benjamin, H, Cholakh, Hila, Cholakh, H, Chajut, A, Chajut, Ayelet, Clark, D P, Clark, Douglas P, Westra, W H, and Westra, William H. Accurate Classification of Non-Small Cell Lung Carcinoma Using a Novel MicroRNA-Based Approach. *Clinical Cancer Research*, 16(2):610–619, January 2010.

[12] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.

[13] Jorrit Boekel, John M Chilton, Ira R Cooke, Peter L Horvatovich, Pratik D Jagtap, Lukas Käll, Janne Lehtiö, Pieter Lukasse, Perry D Moerland, and Timothy J Griffin. Multi-omic data analysis using galaxy. *Nature biotechnology*, 33(2):137–139, 2015.

[14] Rosalin Bonetta and Gianluca Valentino. Machine learning techniques for protein function prediction. *Proteins: Structure, Function, and Bioinformatics*, 88(3):397–413, 2020.

[15] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, pages 2787–2795, 2013.

[16] Emmanuel Boutet, Damien Lieberherr, Michael Tognolli, Michel Schneider, and Amos Bairoch. Uniprotkb/swiss-prot. In *Plant bioinformatics*, pages 89–112. Springer, 2007.

[17] Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? *arXiv preprint arXiv:2105.14491*, 2021.

[18] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a" siamese" time delay neural network. In *Advances in neural information processing systems*, pages 737–744, 1994.

[19] Dechao Bu, Kuntao Yu, Silong Sun, Chaoyong Xie, Geir Skogerbø, Ruoyu Miao, Hui Xiao, Qi Liao, Haitao Luo, Guoguang Zhao, et al. Noncode v3. 0: integrative annotation of long noncoding rnas. *Nucleic acids research*, 40(D1):D210–D215, 2011.

[20] George A Calin and Carlo M Croce. Microrna signatures in human cancers. *Nature reviews cancer*, 6(11):857–866, 2006.

[21] Iván Cantador, Peter Brusilovsky, and Tsvi Kuflik. Second workshop on information heterogeneity and fusion in recommender systems (hetrec2011). In *Proceedings of the fifth ACM conference on Recommender systems*, pages 387–388, 2011.

[22] Yukuo Cen, Yan Wang, Zhenyu Hou, Qibin Chen, and Jie Tang. Cogdl: An extensive research toolkit for deep learning on graphs, 2020.

[23] Alejandra Cervera, Ville Rantanen, Kristian Ovaska, Marko Laakso, Javier Nuñez-Fontarnau, Amjad Alkodsi, Julia Casado, Chiara Facciotto, Antti Häkkinen, Riku Louhimo, et al. Anduril 2: upgraded large-scale data integration framework. *Bioinformatics*, 35(19):3815–3817, 2019.

[24] Kari Chansky, Jean-Paul Sculier, John J Crowley, Dori Giroux, Jan Van Meerbeeck, and Peter Goldstraw. The international association for the study of lung cancer staging project: prognostic factors and pathologic tnm stage in surgically managed non-small cell lung cancer. *Journal of Thoracic Oncology*, 4(7):792–801, 2009.

[25] Andrew Chatr-Aryamontri, Rose Oughtred, Lorrie Boucher, Jennifer Rust, Christie Chang, Nadine K Kolas, Lara O'Donnell, Sara Oster, Chandra

Theesfeld, Adnane Sellam, et al. The biogrid interaction database: 2017 update. *Nucleic acids research*, 45(D1):D369–D379, 2017.

[26] Geng Chen, Ziyun Wang, Dongqing Wang, Chengxiang Qiu, Mingxi Liu, Xing Chen, Qipeng Zhang, Guiying Yan, and Qinghua Cui. Lncrnadisease: a database for long-non-coding rna-associated diseases. *Nucleic acids research*, 41(D1):D983–D986, 2012.

[27] Xing Chen, Chenggang Clarence Yan, Cai Luo, Wen Ji, Yongdong Zhang, and Qionghai Dai. Constructing lncrna functional similarity network based on lncrna-disease associations and disease semantic similarity. *Scientific reports*, 5:11338, 2015.

[28] Liang Cheng, Pingping Wang, Rui Tian, Song Wang, Qinghua Guo, Meng Luo, Wenyang Zhou, Guiyou Liu, Huijie Jiang, and Qinghua Jiang. Lncrna2target v2. 0: a comprehensive database for target genes of lncrnas in human and mouse. *Nucleic acids research*, 47(D1):D140–D144, 2018.

[29] Hyunghoon Cho, Bonnie Berger, and Jian Peng. Diffusion component analysis: unraveling functional topology in biological networks. In *International Conference on Research in Computational Molecular Biology*, pages 62–64. Springer, 2015.

[30] Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. Attention-based models for speech recognition. In *Advances in neural information processing systems*, pages 577–585, 2015.

[31] Chih-Hung Chou, Nai-Wen Chang, Sirjana Shrestha, Sheng-Da Hsu, Yu-Ling Lin, Wei-Hsiang Lee, Chi-Dung Yang, Hsiao-Chin Hong, Ting-Yen Wei, Siang-Jyun Tu, et al. mirtarbase 2016: updates to the experimentally validated mirna-target interactions database. *Nucleic acids research*, 44(D1):D239–D247, 2015.

[32] Chih-Hung Chou, Sirjana Shrestha, Chi-Dung Yang, Nai-Wen Chang, Yu-Ling Lin, Kuang-Wen Liao, Wei-Chi Huang, Ting-Hsuan Sun, Siang-Jyun Tu, Wei-Hsiang Lee, et al. mirtarbase update 2018: a resource for experimentally validated microrna-target interactions. *Nucleic acids research*, 46(D1):D296–D302, 2017.

[33] Hon Nian Chua, Wing-Kin Sung, and Limsoon Wong. Exploiting indirect neighbours and topological weight to predict protein function from protein–protein interactions. *Bioinformatics*, 22(13):1623–1630, 2006.

[34] RNAcentral Consortium. Rnacentral: a comprehensive database of non-coding rna sequences. *Nucleic acids research*, page gkw1008, 2016.

[35] UniProt Consortium. Uniprot: a worldwide hub of protein knowledge. *Nucleic acids research*, 47(D1):D506–D515, 2019.

[36] George S Davidson, Brian N Wylie, and Kevin W Boyack. Cluster stability and the use of noise in interpretation of clustering. In *infovis*, pages 23–30, 2001.

[37] Thomas Derrien, Rory Johnson, Giovanni Bussotti, Andrea Tanzer, Sarah Djebali, Hagen Tilgner, Gregory Guernec, David Martin, Angelika Merkel, David G Knowles, et al. The gencode v7 catalog of human long noncoding rnas: analysis of their gene structure, evolution, and expression. *Genome research*, 22(9):1775–1789, 2012.

[38] Carol E DeSantis, Chun Chieh Lin, Angela B Mariotto, Rebecca L Siegel, Kevin D Stein, Joan L Kramer, Rick Alteri, Anthony S Robbins, and Ahmedin Jemal. Cancer treatment and survivorship statistics, 2014. *CA: a cancer journal for clinicians*, 64(4):252–271, 2014.

[39] Bijan K Dey, Karl Pfeifer, and Anindya Dutta. The h19 long noncoding rna gives rise to micrornas mir-675-3p and mir-675-5p to promote skeletal muscle differentiation and regeneration. *Genes & development*, 28(5):491–501, 2014.

[40] Lee R Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302, 1945.

[41] Yuxiao Dong, Nitesh V Chawla, and Ananthram Swami. metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 135–144, 2017.

[42] Yuxiao Dong, Ziniu Hu, Kuansan Wang, Yizhou Sun, and Jie Tang. Heterogeneous network representation learning. IJCAI, 2020.

[43] Aurora Esquela-Kerscher and Frank J Slack. Oncomirs—micrornas with a role in cancer. *Nature Reviews Cancer*, 6(4):259, 2006.

[44] Tina A Eyre, Fabrice Ducluzeau, Tam P Sneddon, Sue Povey, Elspeth A Bruford, and Michael J Lush. The hugo gene nomenclature database, 2006 updates. *Nucleic acids research*, 34(suppl_1):D319–D321, 2006.

[45] Alessandro Fatica and Irene Bozzoni. Long non-coding rnas: new players in cell differentiation and development. *Nature Reviews Genetics*, 15(1):7, 2014.

[46] Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with pytorch geometric. *arXiv preprint arXiv:1903.02428*, 2019.

[47] Tao-yang Fu, Wang-Chien Lee, and Zhen Lei. Hin2vec: Explore meta-paths in heterogeneous information networks for representation learning. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1797–1806, 2017.

[48] Hongchang Gao and Heng Huang. Deep attributed network embedding. In *Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI))*, 2018.

[49] Sarah Geisler and Jeff Coller. Rna in unexpected places: long non-coding rna functions in diverse cellular contexts. *Nature reviews Molecular cell biology*, 14(11):699, 2013.

[50] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. PMLR, 2017.

[51] David S Goodsell. The machinery of life. 2009.

[52] John C Gower. A general coefficient of similarity and some of its properties. *Biometrics*, pages 857–871, 1971.

[53] Palash Goyal and Emilio Ferrara. Graph embedding techniques, applications, and performance: A survey. *arXiv preprint arXiv:1705.02801*, 2017.

[54] Sam Griffiths-Jones, Russell J Grocock, Stijn Van Dongen, Alex Bateman, and Anton J Enright. mirbase: microrna sequences, targets and gene nomenclature. *Nucleic acids research*, 34(suppl_1):D140–D144, 2006.

[55] Sam Griffiths-Jones, Harpreet Kaur Saini, Stijn van Dongen, and Anton J Enright. mirbase: tools for microrna genomics. *Nucleic acids research*, 36(suppl_1):D154–D158, 2007.

[56] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864, 2016.

[57] Yuanfang Guan, Chad L Myers, David C Hess, Zafer Barutcuoglu, Amy A Caudy, and Olga G Troyanskaya. Predicting gene function in a hierarchical context with an ensemble of classifiers. *Genome biology*, 9(1):1–18, 2008.

[58] Aric Hagberg, Pieter Swart, and Daniel S Chult. Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008.

[59] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in neural information processing systems*, pages 1024–1034, 2017.

[60] Yajing Hao, Wei Wu, Hui Li, Jiao Yuan, Jianjun Luo, Yi Zhao, and Runsheng Chen. Npinter v3. 0: an upgraded database of noncoding rna-associated interactions. *Database*, 2016, 2016.

[61] Jennifer Harrow, Adam Frankish, Jose M Gonzalez, Electra Tapanari, Mark Diekhans, Felix Kokocinski, Bronwen L Aken, Daniel Barrell, Amonida Zadissa, Stephen Searle, et al. Gencode: the reference human genome annotation for the encode project. *Genome research*, 22(9):1760–1774, 2012.

[62] Mohammed A Hassan, Kaltoom Al-Sakkaf, Mohammed Razeeth Shait Mohammed, Ashraf Dallol, Jaudah Al-Maghrabi, Alia Aldahlawi, Sawsan Ashoor, Mabrouka Maamra, Jiannis Ragoussis, Wei Wu, et al. Integration of transcriptome and metabolome provides unique insights to pathways associated with obese breast cancer patients. *Frontiers in Oncology*, 10:804, 2020.

[63] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[64] Zhiyuan Hu, Cheng Fan, Daniel S Oh, JS Marron, Xiaping He, Bahjat F Qaqish, Chad Livasy, Lisa A Carey, Evangeline Reynolds, Lynn Dressler, et al. The molecular portraits of breast tumors are conserved across microarray platforms. *BMC genomics*, 7(1):96, 2006.

[65] Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. Heterogeneous graph transformer. In *Proceedings of The Web Conference 2020*, pages 2704–2710, 2020.

[66] Hsi-Yuan Huang, Yang-Chi-Dung Lin, Shidong Cui, Yixian Huang, Yun Tang, Jiatong Xu, Jiayang Bao, Yulin Li, Jia Wen, Huali Zuo, et al. mirtarbase

update 2022: an informative resource for experimentally validated mirna–target interactions. *Nucleic acids research*, 50(D1):D222–D230, 2022.

[67] Kexin Huang, Cao Xiao, Lucas Glass, Marinka Zitnik, and Jimeng Sun. Skipgnn: Predicting molecular interactions with skip-graph networks. *arXiv preprint arXiv:2004.14949*, 2020.

[68] Zhou Huang, Jiangcheng Shi, Yuanxu Gao, Chunmei Cui, Shan Zhang, Jianwei Li, Yuan Zhou, and Qinghua Cui. Hmdd v3. 0: a database for experimentally supported human microrna–disease associations. *Nucleic acids research*, 2018.

[69] Rachael P Huntley, Tony Sawford, Prudence Mutowo-Meullenet, Aleksandra Shypitsyna, Carlos Bonilla, Maria J Martin, and Claire O'Donovan. The goa database: gene ontology annotation updates for 2015. *Nucleic acids research*, 43(D1):D1057–D1063, 2015.

[70] Kentaro Inamura and Yuichi Ishikawa. MicroRNA In Lung Cancer: Novel Biomarkers and Potential Tools for Treatment. *Journal of Clinical Medicine*, 5(3):36, March 2016.

[71] Vahid Jalili, Enis Afgan, Qiang Gu, Dave Clements, Daniel Blankenberg, Jeremy Goecks, James Taylor, and Anton Nekrutenko. The galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2020 update. *Nucleic Acids Research*, 2020.

[72] Yuxiang Jiang, Tal Ronnen Oron, Wyatt T Clark, Asma R Bankapur, Daniel D'Andrea, Rosalba Lepore, Christopher S Funk, Indika Kahanda, Karin M Verspoor, Asa Ben-Hur, et al. An expanded evaluation of protein function prediction methods shows an improvement in accuracy. *Genome biology*, 17(1):1–19, 2016.

[73] Ioanna Kalvari, Joanna Argasinska, Natalia Quinones-Olvera, Eric P Nawrocki, Elena Rivas, Sean R Eddy, Alex Bateman, Robert D Finn, and Anton I Petrov.

Rfam 13.0: shifting to a genome-centric resource for non-coding rna families. *Nucleic acids research*, 46(D1):D335–D342, 2017.

[74] Minoru Kanehisa and Susumu Goto. Kegg: kyoto encyclopedia of genes and genomes. *Nucleic acids research*, 28(1):27–30, 2000.

[75] Juanjuan Kang, Qiang Tang, Jun He, Le Li, Nianling Yang, Shuiyan Yu, Mengyao Wang, Yuchen Zhang, Jiahao Lin, Tianyu Cui, et al. Rnainter v4. 0: Rna interactome repository with redefined confidence scoring system and improved accessibility. *Nucleic acids research*, 50(D1):D326–D332, 2022.

[76] Dimitra Karagkouni, Maria D Paraskevopoulou, Serafeim Chatzopoulos, Ioannis S Vlachos, Spyros Tastsoglou, Ilias Kanellos, Dimitris Papadimitriou, Ioannis Kavakiotis, Sofia Maniou, Giorgos Skoufos, et al. Diana-tarbase v8: a decade-long collection of experimentally supported mirna–gene interactions. *Nucleic acids research*, 46(D1):D239–D245, 2018.

[77] Dimitra Karagkouni, Maria D Paraskevopoulou, Spyros Tastsoglou, Giorgos Skoufos, Anna Karavangeli, Vasilis Pierros, Elissavet Zacharopoulou, and Artemis G Hatzigeorgiou. Diana-lncbase v3: indexing experimentally supported mirna targets on non-coding transcripts. *Nucleic acids research*, 48(D1):D101–D110, 2020.

[78] Ulas Karaoz, TM Murali, Stan Letovsky, Yu Zheng, Chunming Ding, Charles R Cantor, and Simon Kasif. Whole-genome annotation by using evidence integration in functional-linkage networks. *Proceedings of the National Academy of Sciences*, 101(9):2888–2893, 2004.

[79] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[80] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[81] KC Kishan, Rui Li, Feng Cui, and Anne Haake. Gne: A deep learning framework for gene network inference by aggregating biological information. *bioRxiv*, page 300996, 2018.

[82] Masatoshi Kitagawa, Kyoko Kitagawa, Yojiro Kotake, Hiroyuki Niida, and Tatsuya Ohhata. Cell cycle regulation by long non-coding rnas. *Cellular and molecular life sciences*, 70(24):4785–4794, 2013.

[83] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML Deep Learning Workshop*, volume 2, 2015.

[84] Maxim V Kuleshov, Matthew R Jones, Andrew D Rouillard, Nicolas F Fernandez, Qiaonan Duan, Zichen Wang, Simon Koplev, Sherry L Jenkins, Kathleen M Jagodnik, Alexander Lachmann, et al. Enrichr: a comprehensive gene set enrichment analysis web server 2016 update. *Nucleic acids research*, 44(W1):W90–W97, 2016.

[85] Maxat Kulmanov and Robert Hoehndorf. Deepgoplus: improved protein function prediction from sequence. *Bioinformatics*, 36(2):422–429, 2020.

[86] Maxat Kulmanov and Robert Hoehndorf. Deepgozero: improving protein function prediction from sequence and zero-shot learning based on ontology axioms. *Bioinformatics*, 38(Supplement$_1$), 062022.

[87] Tuuli Lappalainen, Michael Sammeth, Marc R Friedländer, Peter Ac't Hoen, Jean Monlong, Manuel A Rivas, Mar Gonzalez-Porta, Natalja Kurbatova, Thasso Griebel, Pedro G Ferreira, et al. Transcriptome and genome sequencing uncovers functional variation in humans. *Nature*, 501(7468):506–511, 2013.

[88] Eleonora Leucci, Francesca Patella, Johannes Waage, Kim Holmstrøm, Morten Lindow, Bo Porse, Sakari Kauppinen, and Anders H Lund. microrna-9 targets the long non-coding rna malat1 for degradation in the nucleus. *Scientific reports*, 3:2535, 2013.

[89] Benjamin P Lewis, I-hung Shih, Matthew W Jones-Rhoades, David P Bartel, and Christopher B Burge. Prediction of mammalian microrna targets. *Cell*, 115(7):787–798, 2003.

[90] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[91] Stavros Makrodimitris, Roeland CHJ Van Ham, and Marcel JT Reinders. Automatic gene function prediction in the 2020's. *Genes*, 11(11):1264, 2020.

[92] Ryuta Matsuno and Tsuyoshi Murata. Mell: effective embedding method for multiplex networks. In *Companion Proceedings of the The Web Conference 2018*, pages 1261–1268, 2018.

[93] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[94] Alex L Mitchell, Teresa K Attwood, Patricia C Babbitt, Matthias Blum, Peer Bork, Alan Bridge, Shoshana D Brown, Hsin-Yu Chang, Sara El-Gebali, Matthew I Fraser, et al. Interpro in 2019: improving coverage, classification and access to protein sequence annotations. *Nucleic acids research*, 47(D1):D351–D360, 2019.

[95] Chiara Monti, Mara Zilocchi, Ilaria Colugnat, and Tiziana Alberio. Proteomics turns functional. *Journal of proteomics*, 198:36–44, 2019.

[96] S Mostafavi, D Ray, D Warde-Farley, C Grouios, and Q Morris. Genemania: a real-time multiplea ssociation network integration algorithm for predicting gene function genome biol. 2008; 9 suppl 1: S4, 2008.

[97] Cancer Genome Atlas Research Network et al. Comprehensive genomic characterization of squamous cell lung cancers. *Nature*, 489(7417):519, 2012.

[98] Cancer Genome Atlas Research Network et al. Comprehensive molecular profiling of lung adenocarcinoma. *Nature*, 511(7511):543–550, 2014.

[99] Mingdong Ou, Peng Cui, Jian Pei, Ziwei Zhang, and Wenwu Zhu. Asymmetric transitivity preserving graph embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1105–1114. ACM, 2016.

[100] Rose Oughtred, Chris Stark, Bobby-Joe Breitkreutz, Jennifer Rust, Lorrie Boucher, Christie Chang, Nadine Kolas, Lara O'Donnell, Genie Leung, Rochelle McAdam, Frederick Zhang, Sonam Dolma, Andrew Willems, Jasmin Coulombe-Huntington, Andrew Chatr-aryamontri, Kara Dolinski, and Mike Tyers. The BioGRID interaction database: 2019 update. *Nucleic Acids Research*, 47(D1):D529–D541, 11 2018.

[101] Xiaoyong Pan and Hong-Bin Shen. Learning distributed representations of rna sequences and its application for predicting rna-protein binding sites with a convolutional neural network. *Neurocomputing*, 305:51–58, 2018.

[102] Maria D Paraskevopoulou, Ioannis S Vlachos, Dimitra Karagkouni, Georgios Georgakilas, Ilias Kanellos, Thanasis Vergoulis, Konstantinos Zagganas, Panayiotis Tsanakas, Evangelos Floros, Theodore Dalamagas, et al. Diana-lncbase v2: indexing microrna targets on non-coding transcripts. *Nucleic acids research*, 44(D1):D231–D238, 2015.

[103] Typhaine Paysan-Lafosse, Matthias Blum, Sara Chuguransky, Tiago Grego, Beatriz Lázaro Pinto, Gustavo A Salazar, Maxwell L Bileschi, Peer Bork, Alan Bridge, Lucy Colwell, et al. Interpro in 2022. *Nucleic Acids Research*, 2022.

[104] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710, 2014.

[105] Janet Piñero, Àlex Bravo, Núria Queralt-Rosinach, Alba Gutiérrez-Sacristán, Jordi Deu-Pons, Emilio Centeno, Javier García-García, Ferran Sanz, and Laura I Fur-

long. Disgenet: a comprehensive platform integrating information on human disease-associated genes and variants. *Nucleic acids research*, page gkw943, 2016.

[106] Sue Povey, Ruth Lovering, Elspeth Bruford, Mathew Wright, Michael Lush, and Hester Wain. The hugo gene nomenclature committee (hgnc). *Human genetics*, 109(6):678–680, 2001.

[107] Meng Qu, Jian Tang, Jingbo Shang, Xiang Ren, Ming Zhang, and Jiawei Han. An attention-based collaboration framework for multi-view network representation learning. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1767–1776, 2017.

[108] Daniel Quang and Xiaohui Xie. Danq: a hybrid convolutional and recurrent deep neural network for quantifying the function of dna sequences. *Nucleic acids research*, 44(11):e107–e107, 2016.

[109] Rashika Ramola, Iddo Friedberg, and Predrag Radivojac. The field of protein function prediction as viewed by different domain scientists. *bioRxiv*, 2022.

[110] Shancheng Ren, Yaping Shao, Xinjie Zhao, Christopher S Hong, Fubo Wang, Xin Lu, Jia Li, Guozhu Ye, Min Yan, Zhengping Zhuang, et al. Integration of metabolomics and transcriptomics reveals major metabolic pathways and potential biomarker involved in prostate cancer. *Molecular & Cellular Proteomics*, 15(1):154–163, 2016.

[111] Florian Rohart, Benoit Gautier, Amrit Singh, and Kim-Anh Le Cao. mixomics: An r package for 'omics feature selection and multiple data integration. *PLoS computational biology*, 13(11):e1005752, 2017.

[112] M Saito, A J Schetter, S Mollerup, T Kohno, V Skaug, E D Bowman, E A Mathe, S Takenoshita, J Yokota, A Haugen, and C C Harris. The Association of MicroRNA Expression with Prognosis and Progression in Early-Stage, Non-Small Cell Lung Adenocarcinoma: A Retrospective Analysis of Three Cohorts. *Clinical Cancer Research*, 17(7):1875–1882, March 2011.

[113] Ajanthah Sangaralingam, Abu Z Dayem Ullah, Jacek Marzec, Emanuela Gadaleta, Ai Nagano, Helen Ross-Adams, Jun Wang, Nicholas R Lemoine, and Claude Chelala. 'multi-omic'data analysis using o-miner. *Briefings in bioinformatics*, 20(1):130–143, 2019.

[114] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, pages 593–607. Springer, 2018.

[115] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[116] Yu Shi, Fangqiu Han, Xinwei He, Xinran He, Carl Yang, Jie Luo, and Jiawei Han. mvn2vec: Preservation and collaboration in multi-view network embedding. *arXiv preprint arXiv:1801.06597*, 2018.

[117] Noah Simon, Jerome Friedman, Trevor Hastie, and Robert Tibshirani. A sparse-group lasso. *Journal of Computational and Graphical Statistics*, 22(2):231–245, 2013.

[118] Temple F Smith and Michael S Waterman. Comparison of biosequences. *Advances in applied mathematics*, 2(4):482–489, 1981.

[119] Tim Stuart and Rahul Satija. Integrative single-cell analysis. *Nature Reviews Genetics*, 20(5):257–272, 2019.

[120] Michael PH Stumpf, Carsten Wiuf, and Robert M May. Subnets of scale-free networks are not scale-free: sampling properties of networks. *Proceedings of the National Academy of Sciences*, 102(12):4221–4224, 2005.

[121] Yiwei Sun, Suhang Wang, Tsung-Yu Hsieh, Xianfeng Tang, and Vasant Honavar. Megan: A generative adversarial network for multi-view network embedding. *arXiv preprint arXiv:1909.01084*, 2019.

[122] Yizhou Sun, Jiawei Han, Xifeng Yan, Philip S Yu, and Tianyi Wu. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *Proceedings of the VLDB Endowment*, 4(11):992–1003, 2011.

[123] Gabriel Synnaeve, Thomas Schatz, and Emmanuel Dupoux. Phonetics embedding learning with side information. In *Spoken Language Technology Workshop (SLT), 2014 IEEE*, pages 106–111. IEEE, 2014.

[124] Damian Szklarczyk, Annika L Gable, Katerina C Nastou, David Lyon, Rebecca Kirsch, Sampo Pyysalo, Nadezhda T Doncheva, Marc Legeay, Tao Fang, Peer Bork, et al. The string database in 2021: customizable protein–protein networks, and functional characterization of user-uploaded gene/measurement sets. *Nucleic acids research*, 49(D1):D605–D612, 2021.

[125] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *Proceedings of the 24th international conference on world wide web*, pages 1067–1077, 2015.

[126] Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31, 2012.

[127] Nhat Tran, Vinay Abhyankar, KyTai Nguyen, Ishfaq Ahmad, Jon Weidanz, and Jean Gao. Microrna dysregulational synergistic network: Learning context-specific microrna dysregulations in lung cancer subtypes. In *Bioinformatics and Biomedicine (BIBM), 2017 IEEE International Conference on*, pages 142–145. IEEE, 2017.

[128] Nhat C Tran and Jean X Gao. Openomics: A bioinformatics api to integrate multi-omics datasets and interface with public databases. *Journal of Open Source Software*, 6(61):3249, 2021.

[129] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[130] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. *International Conference on Learning Representations*, 2018.

[131] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[132] Pieter-Jan Volders, Kenny Helsens, Xiaowei Wang, Björn Menten, Lennart Martens, Kris Gevaert, Jo Vandesompele, and Pieter Mestdagh. Lncipedia: a database for annotated human lncrna transcript sequences and structures. *Nucleic acids research*, 41(D1):D246–D251, 2012.

[133] Daixin Wang, Peng Cui, and Wenwu Zhu. Structural deep network embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1225–1234. ACM, 2016.

[134] Sheng Wang, Hyunghoon Cho, ChengXiang Zhai, Bonnie Berger, and Jian Peng. Exploiting ontology graph for predicting sparsely annotated gene function. *Bioinformatics*, 31(12):i357–i364, 2015.

[135] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. Heterogeneous graph attention network. In *The World Wide Web Conference*, pages 2022–2032, 2019.

[136] Mark N Wass, Geraint Barton, and Michael JE Sternberg. Combfunc: predicting protein function using heterogeneous data sources. *Nucleic acids research*, 40(W1):W466–W470, 2012.

[137] Leah A Wasser and Chris Holdgraf. pyopensci promoting open source python software to support open reproducible science. *AGUFM*, 2019:NS21A–13, 2019.

[138] Jason F Wiggins, Lynnsie Ruffino, Kevin Kelnar, Michael Omotola, Lubna Patrawala, David Brown, and Andreas G Bader. Development of a lung cancer therapeutic based on the tumor suppressor microrna-34. *Cancer research*, 70(14):5923–5930, 2010.

[139] Matthew D Wilkerson, Xiaoying Yin, Katherine A Hoadley, Yufeng Liu, Michele C Hayward, Christopher R Cabanski, Kenneth L Muldrew, C Ryan Miller, Scott H Randell, Mark A Socinski, et al. Lung squamous cell carcinoma mrna expression subtypes are reproducible, clinically-important and correspond to different normal cell types. *Clinical cancer research*, pages clincanres–0199, 2010.

[140] F Alexander Wolf, Philipp Angerer, and Fabian J Theis. Scanpy: large-scale single-cell gene expression data analysis. *Genome biology*, 19(1):15, 2018.

[141] Shunquan Wu, Shenglin Huang, Jie Ding, Y Zhao, L Liang, Te Liu, Rong Zhan, and Xianghuo He. Multiple microrna modulate p21cip1/waf1 expression by directly targeting its 3 untranslated region. *Oncogene*, 29(15):2302, 2010.

[142] Juan Xu, Chuan-Xing Li, Yong-Sheng Li, Jun-Ying Lv, Ye Ma, Ting-Ting Shao, Liang-De Xu, Ying-Ying Wang, Lei Du, Yun-Peng Zhang, et al. Mirna–mirna synergistic network: construction via co-regulating functional modules and disease mirna topological features. *Nucleic acids research*, 39(3):825–836, 2010.

[143] Juan Xu, Chuan-Xing Li, Jun-Ying Lv, Yong-Sheng Li, Yun Xiao, Ting-Ting Shao, Xiao Huo, Xiang Li, Yan Zou, Qing-Lian Han, et al. Prioritizing candidate disease mirnas by topological features in the mirna target–dysregulated network: Case study of prostate cancer. *Molecular cancer therapeutics*, 10(10):1857–1866, 2011.

[144] Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. Representation learning on graphs with jumping knowledge networks. *arXiv preprint arXiv:1806.03536*, 2018.

[145] Jingwen Yan, Shannon L Risacher, Li Shen, and Andrew J Saykin. Network approaches to systems biology analysis of complex disease: integrative methods for multi-omics data. *Briefings in bioinformatics*, 19(6):1370–1381, 2018.

[146] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*, 2014.

[147] Yaming Yang, Ziyu Guan, Jianxin Li, Jianbin Huang, and Wei Zhao. Interpretable and efficient heterogeneous graph convolutional network. *arXiv preprint arXiv:2005.13183*, 2020.

[148] Zhen Yang, Fei Ren, Changning Liu, Shunmin He, Gang Sun, Qian Gao, Lei Yao, Yangde Zhang, Ruoyu Miao, Ying Cao, Yi Zhao, Yang Zhong, and Haitao Zhao. dbDEMC: a database of differentially expressed miRNAs in human cancers. *BMC Genomics*, 11(Suppl 4):S5, 2010.

[149] Je-Hyun Yoon, Kotb Abdelmohsen, and Myriam Gorospe. Functional interactions among microrna and long noncoding rnas. In *Seminars in cell & developmental biology*, volume 34, pages 9–14. Elsevier, 2014.

[150] Ronghui You, Shuwei Yao, Hiroshi Mamitsuka, and Shanfeng Zhu. Deepgraphgo: graph neural network for large-scale, multispecies protein function prediction. *Bioinformatics*, 37(Supplement_1):i262–i271, 2021.

[151] Guoxian Yu, Yingwen Zhao, Chang Lu, and Jun Wang. Hashgo: hashing gene ontology for protein function prediction. *Computational biology and chemistry*, 71:264–273, 2017.

[152] Jiao Yuan, Wei Wu, Chaoyong Xie, Guoguang Zhao, Yi Zhao, and Runsheng Chen. Npinter v2. 0: an updated database of ncrna interactions. *Nucleic acids research*, 42(D1):D104–D108, 2013.

[153] Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J Kim. Graph transformer networks. In *Advances in Neural Information Processing Systems*, pages 11983–11993, 2019.

[154] Bin Zhang, Steve Horvath, et al. A general framework for weighted gene co-expression network analysis. *Statistical applications in genetics and molecular biology*, 4(1):1128, 2005.

[155] Bing Zhang, Jing Wang, Xiaojing Wang, Jing Zhu, Qi Liu, Zhiao Shi, Matthew C Chambers, Lisa J Zimmerman, Kent F Shaddox, Sangtae Kim, et al. Proteogenomic characterization of human colon and rectal cancer. *Nature*, 513(7518):382–387, 2014.

[156] Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V Chawla. Heterogeneous graph neural network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 793–803, 2019.

[157] Fuhao Zhang, Hong Song, Min Zeng, Yaohang Li, Lukasz Kurgan, and Min Li. Deepfunc: a deep learning framework for accurate prediction of protein functions from protein sequences and interactions. *Proteomics*, 19(12):1900019, 2019.

[158] Hongming Zhang, Liwei Qiu, Lingling Yi, and Yangqiu Song. Scalable multiplex network embedding. In *IJCAI*, volume 18, pages 3082–3088, 2018.

[159] Wenyu Zhang, Jin Zang, Xinhua Jing, Zhandong Sun, Wenying Yan, Dongrong Yang, Feng Guo, and Bairong Shen. Identification of candidate mirna biomarkers from mirna regulatory network with application to prostate cancer. *Journal of translational medicine*, 12(1):66, 2014.

[160] Jun Zhao, Zhou Zhou, Ziyu Guan, Wei Zhao, Wei Ning, Guang Qiu, and Xiaofei He. Intentgc: a scalable graph convolution framework fusing heterogeneous information for recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2347–2357, 2019.

[161] Da Zheng, Minjie Wang, Quan Gan, Zheng Zhang, and Geroge Karypis. Scalable graph neural networks with deep graph library. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3521–3522, 2020.

[162] Guangjie Zhou, Jun Wang, Xiangliang Zhang, Maozu Guo, and Guoxian Yu. Predicting functions of maize proteins using graph convolutional network. *BMC bioinformatics*, 21(16):1–16, 2020.

[163] Naihui Zhou, Yuxiang Jiang, Timothy R Bergquist, Alexandra J Lee, Balint Z Kacsoh, Alex W Crocker, Kimberley A Lewis, George Georghiou, Huy N Nguyen, Md Nafiz Hamid, et al. The cafa challenge reports improved protein function prediction and new functional annotations for hundreds of genes through experimental screens. *Genome biology*, 20(1):1–23, 2019.

[164] Sheng Zhou, Jiajun Bu, Xin Wang, Jiawei Chen, and Can Wang. Hahe: Hierarchical attentive heterogeneous information network embedding. *arXiv preprint arXiv:1902.01475*, 2019.

[165] Yitan Zhu, Peng Qiu, and Yuan Ji. Tcga-assembler: open-source software for retrieving and processing tcga data. *Nature methods*, 11(6):599–600, 2014.

## BIOGRAPHICAL STATEMENT

Nhat C. Tran was born in Da Nang, Vietnam in 1993. He received his B.S. degree in 2015 and his Ph.D. degree in 2022, all in Computer Science from The University of Texas at Arlington in 2022. His research interest is in open-source software, machine learning and bioinformatics, specifically in the areas of graph neural networks, multi-omics integration, non-coding RNAs, and protein function prediction.