Multilayer Graph Embeddings for Omics Data Integration in Bioinformatics

Surabhi Jagtap

université PARIS-SACLAY

Multilayer Graph Embeddings for Omics Data Integration in Bioinformatics

Plongement de graphes multicouches pour l'intégration de données omiques en bioinformatique

Thèse de doctorat de l'université Paris-Saclay

École doctorale n° 580 :
Sciences et technologies de l'information et de la communication (STIC)
Spécialité de doctorat: Informatique mathématique
Graduate School : Sciences de l'ingénieur et des systèmes (SIS)
Référent : CentraleSupélec

Thèse préparée dans l'unité de recherche
Centre de vision numérique (CentraleSupélec, Inria), sous la direction de
Jean-Christophe PESQUET, Professeur, le co-encadrement de
Fragkiskos MALLIAROS, Maître de conférences, et
Laurent DUVAL, Chef de projet, IFP Energies Nouvelle

Thèse soutenue à Gif-sur-Yvette, le 2 Février 2023, par

Surabhi JAGTAP

Composition du jury

Membres du jury avec voix délibérative

Macha NIKOLSKI Directrice de recherche, CNRS, Université de Bordeaux	Présidente
Thierry ARTIÈRES Professeur, École Centrale Marseille	Rapporteur
Mehmet KOYUTÜRK Professeur, Case Western Reserve University, USA	Rapporteur & Examinateur
Laurence CALZONE Ingénieure de Recherche, Institut Curie	Examinatrice

THESE DE DOCTORAT

NNT : 2023UPAST014

université PARIS-SACLAY

ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre : Plongement de graphes multicouches pour l'intégration de données omiques en bioinformatique

Mots clés : Données multi-omiques, Réseaux biologiques, Apprentissage de la représentation graphique, Intégration de données

Résumé : Les systèmes biologiques sont composés de biomolécules en interaction à différents niveaux moléculaires. D'un côté, les avancées technologiques ont facilité l'obtention des données omiques à ces divers niveaux. De l'autre, de nombreuses questions se posent, pour donner du sens et élucider les interactions importantes dans le flux d'informations complexes porté par cette énorme variété et quantité des données multi-omiques. Les réponses les plus satisfaisantes seront celles qui permettront de dévoiler les mécanismes sous-jacents à la condition biologique d'intérêt.

On s'attend souvent à ce que l'intégration de différents types de données omiques permette de mettre en lumière les changements causaux potentiels qui conduisent à un phénotype spécifique ou à des traitements ciblés. Avec les avancées récentes de la science des réseaux, nous avons choisi de traiter ce problème d'intégration en représentant les données omiques à travers les graphes.

Dans cette thèse, nous avons développé trois modèles à savoir BraneExp, BraneNet et BraneMF pour l'apprentissage d'intégrations de nœuds à partir de réseaux biologiques multicouches générés à partir de données omiques. Notre objectif est de résoudre divers problèmes complexes liés à l'intégration de données multi-omiques, en développant des méthodes expressives et évolutives capables de tirer parti de la riche sémantique structurelle latente des réseaux du monde réel.

Title: Multilayer Graph Embeddings for Omics Data Integration in Bioinformatics

Keywords: Multi-omics data, Biological networks, Graph representation learning, Data integration

Abstract: Biological systems are composed of interacting bio-molecules at different molecular levels. With the advent of high-throughput technologies, omics data at their respective molecular level can be easily obtained. These huge, complex multi-omics data can be useful to provide insights into the flow of information at multiple levels, unraveling the mechanisms underlying the biological condition of interest.

Integration of different omics data types is often expected to elucidate potential causative changes that lead to specific phenotypes, or targeted treatments. With the recent advances in network science, we choose to handle this integration issue by representing omics data through networks.

In this thesis, we have developed three models, namely BraneExp, BraneNet, and BraneMF, for learning node embeddings from multilayer biological networks generated with omics data. We aim to tackle various challenging problems arising in multi-omics data integration, developing expressive and scalable methods capable of leveraging rich structural semantics of real-world networks.

Résumé

Les systèmes biologiques sont composés de biomolécules en interaction à différents niveaux moléculaires. Si, d'un côté, l'avancée technologique a facilité l'obtention des données omiques à ces niveaux-là, de l'autre côté, plusieurs questions se posent, pour extraire du sens dans le flux complexe d'informations portées par cette énorme variété et quantité des données multi-omiques. Les réponses qui les satisferont le mieux seront celles qui dévoileront les mécanismes sous-jacents à la condition biologique d'intérêt. Ceux-ci peuvent inclure l'inférence de la régulation des gènes, l'identification des biomarqueurs responsables du phénotype observé, la connaissance des voies biologiques qui sont affectées dans différentes conditions expérimentales. Chaque type de données omiques peut, à lui seul, fournir des informations sur des biomolécules associées au phénotype. Cependant, l'analyse des données omiques simples est limitée à des corrélations, reflétant principalement des processus réactifs plutôt que des processus causaux. On s'attend donc à ce que l'intégration de différents types de données omiques permette de mieux élucider les changements causaux potentiels qui conduisent à un phénotype spécifique ou qui permettraient des traitements ciblés.

Avec les avancées récentes de la science des réseaux, nous avons choisi de traiter ce problème d'intégration en représentant les données omiques au travers de graphes. Cette approche ouvre un vaste champ d'exploration et d'étude de la biologie moléculaire du système cellulaire des organismes à l'aide de techniques d'apprentissage par représentation de graphes (GRL). L'idée-clé derrière les approches GRL est d'apprendre une cartographie qui intègre des nœuds en tant que points dans un espace vectoriel de faible dimension. L'objectif est d'optimiser cette cartographie, de sorte que les relations géométriques dans cet espace appris reflètent la structure du graphe d'origine. Notre objectif est d'obtenir de telles représentations, également connues sous le nom de plongements, pour chaque biomolécule à partir d'un ensemble bien choisi de modalités omiques. Les plongements sont formés de telle sorte qu'ils englobent au mieux des informations multi-omiques, afin comprendre plus précisément les variations menant du génotype au phénotype. Les intégrations apprises peuvent alors être utilisées comme attributs d'entrée pour des tâches d'apprentissage automatique, en aval du processus d'interprétation biologique.

Dans cette thèse, nous avons développé trois modèles, à savoir BraneExp, BraneNet et BraneMF pour l'apprentissage de plongements de nœuds à partir de graphes biologiques multicouches, générés à partir de données omiques. Notre objectif est de résoudre divers problèmes complexes liés à l'intégration de

données multi-omiques, en développant des méthodes expressives et évolutives capables de tirer parti de la riche sémantique structurelle des graphs du monde réel. La fonction objective dans ces méthodes est indépendante des tâches en aval et les intégrations de nœuds sont apprises de manière totalement non supervisée. BraneExp tire parti des plongements de graphes "de famille exponentielle", qui généralisent les méthodes GRL basées sur la marche aléatoire multicouche à une instance de distribution de probabilité de la famille exponentielle. BraneNet effectue l'intégration en tirant parti d'une matrice d'informations mutuelles ponctuelles positives (PPMI) multicouche correctement choisie. Les plongements sont appris en réalisant une factorisation matricielle, se rapprochant du spectre de cette matrice PPMI. BraneMF calcule les matrices PPMI pour chaque couche et apprend les intégrations en utilisant le cadre de décomposition conjointe en valeurs singulières (SVD). Nous démontrons les applications des plongements appris pour résoudre d'importantes tâches bioinformatiques en aval, par exemple, l'inférence du réseau de régulation génique (GRN), le regroupement de biomolécules biologiquement liées, la prédiction des fonctions protéiques et les interactions protéine-protéine (PPI). Nous avons effectué une analyse approfondie en comparant les performances des méthodes développées aux méthodes d'intégration de référence pour les graphes multicouches.

Abstract

Biological systems are composed of interacting bio-molecules (e.g., genes, proteins, metabolites) at different molecular levels. With the advent of high-throughput technologies, omics data at their respective molecular level can be easily obtained. These huge complex multi-omics data can be useful to provide insights into the flow of information at multiple levels, unraveling the mechanisms underlying the biological condition of interest. These may include gene regulation inference, identification of bio-markers responsible for observed phenotype, and knowing the biological pathways that are affected within different experimental conditions. Individual type of omics data, on its own, provides information on bio-molecules associated with the phenotype. However, analysis of single omics data is limited to correlations, mostly reflecting reactive processes rather than causative ones. Integration of different omics data types is often expected to elucidate potential causative changes that lead to specific phenotypes, or targeted treatments.

With the recent advances in network science, we choose to handle this integration issue by representing omics data through networks. It has opened a wide area for us to explore and study the molecular biology of the organism's cellular system using graph representation learning (GRL) techniques. The key idea behind GRL approaches is to learn a mapping that embeds nodes as points in a low-dimensional vector space. The aim is to optimize this mapping such that the geometric relationships in this learned space reflect the structure of the original graph. Our goal is to derive representations, also known as embeddings for each bio-molecule from a well-chosen set of omics modalities. The embeddings are trained such that they are expected to encompass multi-omics information, so as to better understand genotype to phenotype variations. The learned embeddings can be used as feature inputs for downstream machine learning tasks.

In this thesis, we have developed three models namely BraneExp, BraneNet, and BraneMF for learning node embeddings from multilayer biological networks generated from omics data. We aim to tackle various challenging problems arising in multi-omics data integration, developing expressive and scalable methods capable of leveraging rich structural semantics of real-world networks. The objective function in these methods is independent of downstream tasks and the node embeddings are learned in a completely unsupervised manner. BraneExp takes advantage of exponential family graph embeddings that generalize multi-layer random walk-based GRL methods to an instance of the exponential family probability distribution. BraneNet performs integration by leveraging a properly

chosen multilayer random walk-based Positive Pointwise Mutual Information (PPMI) matrix. The embeddings are learned by performing matrix factorization, approximating the spectrum of this PPMI matrix. BraneMF computes PPMI matrices for each layer and learns embeddings by using a joint singular value decomposition (SVD) framework. We demonstrate the applications of the learned embeddings to solve important downstream bioinformatics tasks, for instance, Gene regulatory network (GRN) inference, clustering of biologically related bio-molecules, prediction of protein functions, and protein-protein interactions (PPIs). We have performed extensive analysis by comparing the performance of developed methods against the state-of-the-art integration methods for multilayer networks.

List of Publications

The proposed work has yielded the following publications:

- Journal papers:
  - **Jagtap, S.**, Çelikkanat, A., Pirayre, A., Bidard, F., Duval, L. and Malliaros, F.D., BraneMF: Integration of Biological Networks for Functional Analysis of Proteins. Bioinformatics (2022). [Jag+22a]
  - **Jagtap, S.**, Pirayre, A., Bidard, F., Duval, L. and Malliaros, F.D., BRANEnet: Embedding Multilayer Networks for Omics Data Integration. BMC Bioinformatics 23, 429 (2022). [Jag+22b]
- International conferences with proceedings:
  - **Jagtap, S.**, Çelikkanat, A., Pirayre, A., Bidard, F., Duval, L. and Malliaros, F.D., Multiomics Data Integration for Gene Regulatory Network Inference with Exponential Family Embeddings. In 29th European Signal Processing Conference (EUSIPCO), 2021. [Jag+21a]
  - **Jagtap, S.**, Pirayre, A., Bidard, F., Duval, L. and Malliaros, F.D., 2021, November. BRANet: Graph-based Integration of Multi-omics Data with Biological a priori for Regulatory Network Inference. In 17th International Conference on Computational Intelligence Methods for Bioinformatics and Biostatistics (CIBB), 2021. [Jag+21b]
- Poster presentations:
  - **Jagtap, S.**, Çelikkanat, A., Pirayre, A., Bidard, F., Duval, L. and Malliaros, F.D., BraneMF: Random Walk-based Matrix Factorization of a Multi-layer Network for Protein Function Prediction. European Conference on Computational Biology (ISMB/ECCB), 2021.
- Publication during the Ph.D. studies that is not included in the dissertation:
  - Hocq, R., **Jagtap, S.**, Boutard, M., Tolonen, A.C., Duval, L., Pirayre, A., Lopes Ferreira, N. and Wasels, F., Genome-Wide TSS Distribution in Three Related Clostridia with Normalized Capp-Switch Sequencing. Microbiology Spectrum, 10(2), pp.e02288-21 (2022). [Hoc+22]

Acknowledgment

I would like to thank Prof. Fragkiskos Malliaros, Dr. Aurélie Piyare, Dr. Laurent Duval and Dr. Frédérique-Michelot Bidard for their excellent supervision and guidance during my Ph.D. dissertation. Especially Fragkiskos have consistently encouraged me to stay afloat during my doctorate studies. His enthusiasm, significant interest in the research topic, and helpful demeanor has really motivated me. I am very grateful and lucky to perform my Ph.D. dissertation work under mentorship of all my supervisors.

This thesis was co-funded by IFPEN and Centre de Vision Numérique (CVN), CentraleSupélec. I would like to thank both organisations for their generous support. I also want to thank Prof. Jean-Christophe Pesquet, the thesis director and the director of CVN laboratory (CentraleSuéplec) and Yann Creff, head of control, signal and system department (IFPEN), for all your support.

I would like to express my sincere gratitude to the rapporteurs, Prof. Thierry Artieres and Prof. Mehmet Koyutürk, and the examiners, Dr. Macha Nikolski and Dr. Laurence Calzone, for their incisive judgment and invaluable comments and feedback.

Further, I would like to thank Jana Dutrey, Anne Batalie, Francoise Pinson-David, Carole Geunaud and Michel for all the help and support they have provided during my time at CentraleSupélec and IFPEN.

Many thanks to Abdulkadir Çelikkanat and Monacer Ericson Da Silva for their exceptional support and continuous encouragement during my Ph.D. studies. I have learned a lot from them, which helped me to develop my scientific and technical skills. Additionally, I would like to thank all my friends and colleagues, especially Kirti, Sachin, Kavya, Sagar, Pragya, Shalu, Sophia, Julien, Sarah, William, Mateos, Fufang, Danish to make my life in France joyful, exciting, and fantastic.

Many thanks Dr. Mickaël Ménager for his support and motivation. I would also like to thank all my collegues at Institut Imagine especially Francesco Carbone and Jiyoung Oh.

Lastly, I'm eternally grateful and much obliged to thank to my father, Vasantrao Jagtap; mother, Swati Jagtap; sister, Vaishnavi Jagtap; and brother, Nakshatra Jagtap for their endless love, support, and faith in me.

Contents

List of Figures

List of Tables

Abbreviations

ACC Accuracy

AUPR Area Under Precision-Recall Curve

BP Biological Process

BRANE Biologically-Related Apriori Network Estimation

BS-Seq bisulfite sequencing

CC Cellular Component

ChIP-Seq Chromatin Immuno Precipitation sequencing

CNN Convolutional Neural Networks

CV Cross Validation

DNA Deoxyribonucleic acid

GATs Graph Attention Networks

GCN Graph Convolutional Network

GO Gene Ontology

GRL Graph Representation Learning

GRN Gene Regulatory Network

ION Integrated Omics Network Factor

MCC Matthews Correlation Coefficient

MF Molecular Function

mRNA messenger RNA

NLP Natural Language Processing

PPI Protein Protein Interaction

PPMI Positive Pointwise Mutual Information

RBF Radial Basis Function

RNA Ribonucleic acid

RNN recurrent neural networks

SVD singular value decomposition

SVM Support Vector Machine

VGAE Variational Graph Auto-Encoders

WGCNA Weighted Correlation Network Analysis method

List of Symbols

γ	Weighting factor
$\mathbb{R}$	Set of real numbers
$\bar{\mathbf{A}}$	Supra-adjacency matrix
$\mathbf{A}$	Adjacency matrix
$\mathbf{\Omega}$	Embedding
$\mathbf{\Sigma}$	Singular value matrix
$\mathbf{C}$	Block matrix
$\mathbf{D}$	Degree matrix
$\mathbf{M}$	PPMI matrix
$\mathbf{P}$	Transition (power) matrix
$\mathbf{U}$	Singular vector matrix
$\mathbf{V}$	Singular vector matrix
$\mathbf{w}$	Walk
$\mathcal{C}_t(\mathbf{w}_{(n,i)})$	Context set of node w_l in the walk $\mathbf{w}$
$\mathcal{E}$	Set of edges
$\mathcal{O}$	Objective function
$\mathcal{V}$	Set of nodes
$\mathcal{W}$	Set of walks
σ	Learning rate
τ	Proximity score
θ	Node similarity score
$A(\eta)$	log-normalizer function
b	Batch size
d	Embedding size

e	Number of epochs
G	Multilayer network
$h(x)$	Base measure
K	Number of k-means clusters
k	Number of negative samples
L	Number of layers
n	Number of walks
p_r	Restart propability
p_t	Probability distribution at time t
S_C	Cosine similarity
$T(x)$	Suffucient statistic
t	Window size
w	Walk length
X	Set of x intra-omics networks
Y	Set of x inter-omics networks

1

Introduction

This chapter provides information about the context and motivation of the thesis, an overview of contributions, and an outline of the thesis. Section 1.1 contains a short paragraph about the biological question of interest that is addressed in this thesis. It includes a brief introduction to the approach, challenges, and potential applications. The thesis statement and overview of the contributions are presented in Section 1.2. Later, in Section 1.3, the outline of the thesis is provided in tabular format.

1.1 Context and Motivation

Over the last decade, omics technologies have advanced tremendously, culminating in the deciphering of genome, proteome, epigenome, and metabolome sequences. Their advent has showered us with a large amount of omics data at different molecular levels. This achievement is a major milestone in the understanding of a biological system, as omics technologies provide a catalog of all associated bio-molecules that are required for creating a living organism [Sub+20]. Yet, it is not sufficient to identify and characterize the molecules individually. Also, it is necessary to obtain a thorough understanding of the interactions between molecules and pathways that play a major role in cellular functioning. Over the past years, mathematical models have allowed us to investigate how complex regulatory processes are connected and how disruptions of these processes may contribute to the development of diseases or mutations in the strains of organisms.

Indeed, various studies have shown that joint analysis of omics datasets yields a better understanding and clearer picture of the cellular mechanisms and their regulatory behaviors in the biological system under the study [Sub+20]. For instance, a joint analysis of transcriptomic and proteomic data can provide useful insights about gene regulation that may not be deciphered from individual analysis of mRNA or protein expressions [HP13]. Also, in proteogenomics

approaches, genome and transcriptome data is used to generate customized protein sequence databases to help interpret proteomics data [Nes14]. Lately, multi-omics data integration (or integrative omics) was proposed as a combination of methods to fuse data obtained from different omic approaches, aiming at gaining insights into the interconnectedness of different biomolecules (e.g., proteins, RNAs, metabolites) and the flow of biological information that occurs within them. Network approaches have generated substantial interest based on their potential for integrative omics analysis and are expected to facilitate a new era of systems biology [Sub+20; Yan+18; Di +20; CBL16].

Considering the dimensionality and heterogeneity of omics data, a critical first step is data curation, which is the key enabler of initial multi-omics analyses [Yan+18]. This curation will significantly narrow the search space, giving further insights to retrieve high-profile biological information, for instance, modeling specific pathways, network module identification, network inference, and protein function prediction. In addition to the statistical analysis of integrated multi-omics data, an informative integrative visualization of high-dimensional multi-omics networks is another challenging goal that might be helpful to understand inter/intra-omics interactions that lead to the phenotype [Yue+20].

Graphs are a form of structured data employed extensively in informatics or computer science-related fields. Social networks, molecular relational structures, and biological networks can be easily modeled as graphs, which capture interactions (edges) between individual units (nodes). As a consequence of their ubiquity, graphs allow relational knowledge about interacting entities to be efficiently stored, accessed, and visualized [HYL17b]. The key idea behind Graph Representation Learning (GRL) approaches is to learn a mapping that embeds nodes as points in a low-dimensional vector space, $\mathbb{R}^d$. The aim is to optimize this mapping such that the geometric relationships in this learned space reflect the structure of the original graph. After optimizing the embedding space, the learned embeddings can be used as feature inputs for downstream machine learning tasks. The primary input to the representation learning algorithm is a graph $G = (\mathcal{V}, \mathcal{E})$ with an associated adjacency matrix $\mathbf{A}$. The goal is to use the information contained in $\mathbf{A} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$ to map each node $i \in \mathcal{V}$ to a vector, $\mathbf{\Omega}_i \in \mathbb{R}^d$, where $d \ll |\mathcal{V}|$.

The recent advances in this field have opened a wide area to explore and study the molecular biology of an organism's cellular system using representation learning techniques [HYL17b; Ham20; Yue+20]. Our goal is to derive representations, also known as embeddings, for each bio-molecule from a well-chosen set of omics modalities. The embeddings are trained such that they are expected to encompass multi-omics information to better understand genotype to phenotype variations. To that end, we aim to design integrative models for omics data analysis. The development of such models is of great significance for numerous applications that involve:

- Gene regulation inference (miRNA - mRNA, TF-target interactions).

- Protein function prediction (biological process, molecular functions).
- Protein Protein Interaction (PPI) prediction.
- Clustering of functionally related proteins.

Integration of multi-omics dataset to derive a holistic understanding of biological processes and diseases comes with its share of challenges [Sub+20]. The underlying heterogeneity in the individual omics data, the large size of data sets leading to compute-intensive analysis, and the lack of studies that help in prioritizing the diverse set of tools make multi-omics data integration and analysis a challenging task. Multi-omics data are generated using a wide range of platforms, and hence the data storage and formats vary considerably. Most of the multi-omics integrative analysis tools require data to be in specific formats (mostly in a "feature × sample" matrix), and therefore the individual omics data need pre-processing.

The pre-processing step includes data filtering, systematic normalization, removal of batch effects, and quality checks. It becomes imperative to carefully follow these pre-processing steps as they have a huge influence on the integrative analysis. For instance, the data filtering step plays an important role in filtering the noise and reducing the number of features that go into integrative models—as most of the integrative methods are computationally intensive, and hence it is a prerequisite to reduce the size of the input data sets. However, deciding appropriate criteria for filtering is challenging because of the lack of universal standards. Perez-Riverol *et. al.* [Per+17] have developed a workflow that could guide feature selection from high-dimensional omics datasets. In this regard, the development of new integrative methods must consider the efficient handling of large data sets. The primary key to any integrative analysis is the right choice of method that can address the biological question of interest. There are several studies that perform benchmarking of integrative tools [Lee+20; Dua+21; Can+21] but are not comprehensive enough in terms of choice of tools in the context of biological question of interest.

Most such comprehensive studies are needed to guide the community in a better understanding of the wide range of tools. Therefore, we consider the following points of interest to address the above challenges while developing graph-based multi-omics data integration models:

- Standardize the multi-omics data of different scales/platforms using networks. Nodes in these networks represent bio-molecules, and edges represent the relationship between them.
- Design mathematical models for multi-omics data integration that could effectively capture the complex interactions and relationships among bio-molecules towards improving expressiveness and preserving the unique properties of input data sources.

- Perform extensive evaluation of the proposed models in the context of biological questions of interest (downstream tasks).
- Comparison of the performance of proposed models to state-of-the-art network integration techniques.

This dissertation is done in the collaboration with IFP Energies nouvelles (IFPEN). At IFPEN, biologists work on various micro-organisms in the context of green chemistry, for example, bio-ethanol production. The production of bio-ethanol is driven by bio-catalysts (enzymes). Biologists aim to optimize biological mechanisms in fungal strains to improve their production of bio-catalysts. For this purpose, it is crucial to understand enzyme production mechanisms. This can be achieved by understanding the molecular biology of the organism's cellular system that is considered for the study [Sil16]. The advent of high throughput sequencing technologies generates huge omics data. Thus, whole genome sequencing (DNA-seq) allows genome assembly; RNA sequencing (RNA-seq) can be used to analyze the level of transcription of genes; Chromatin Immuno Precipitation sequencing (ChIP-Seq) and bisulfite sequencing (BS-Seq) can be used to identify epigenetic changes. Because of their heterogeneity, these datasets are mostly processed independently but hardly associated to get a full picture of biological mechanisms. Therefore, to obtain an understanding of biological mechanisms, this kind of information from multi-omics data can be integrated [Kub13]. For this purpose, we choose to handle this integration issue by representing omics data through graphs [BO04; CWZ09; Yu+13]. Over the past few years, Biologically-Related Apriori Network Estimation (BRANE) methods have been introduced at IFPEN to develop a suite of bioinformatics tools based on graphs and optimization, dedicated to transcriptomics data: BRANE Cut [Pir+15a], BRANE Clust [Pir+17], and BRANE Relax [Pir+15b]. With this thesis, we introduce methods for multi-omics data integration, namely BraneExp[Jag+21a], BraneNet [Jag+22b], and BraneMF [Jag+22a].

1.2 Thesis Statement and Overview of Contributions

The aim of this thesis is to perform an integrative analysis of biological networks with Graph Representation Learning approaches. Figure 1.1 gives an overview of the contributions of this thesis.

Three models, namely BraneExp, BraneNet, and BraneMF have been developed during this dissertation. The aim is to learn integrated node embeddings from multilayer biological networks generated from multi-omics data. The objective function in these methods is independent of the downstream tasks, and the node embeddings are learned in a completely unsupervised way. BraneExp takes advantage of exponential family graph embeddings that generalize multilayer random walk-based GRL methods to an instance of the exponential family probability distribution. BraneNet performs integration by leveraging a properly chosen multilayer random walk-based Positive Pointwise Mutual Information (PPMI) matrix.

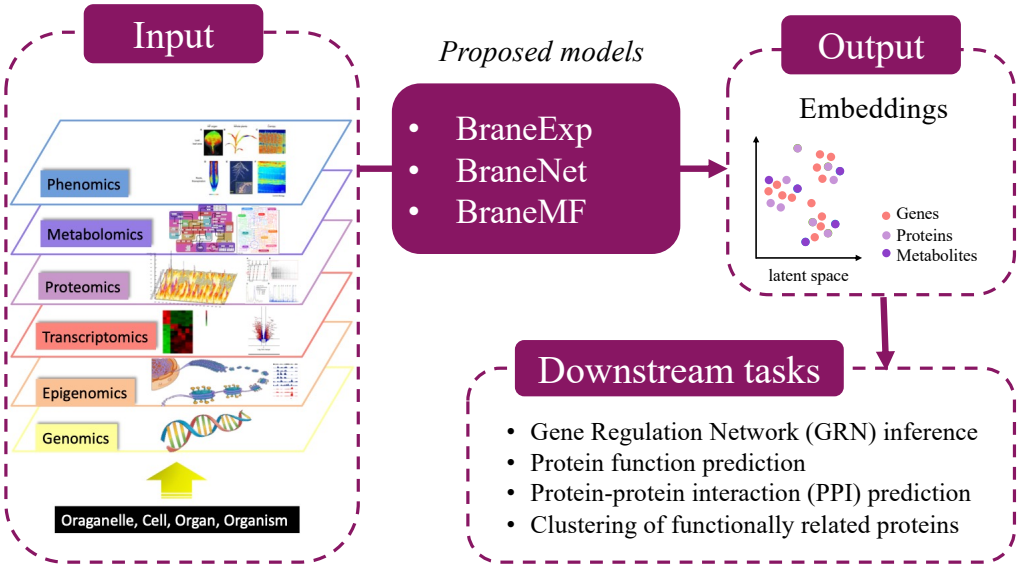


Figure 1.1: **Thesis statement and overview of contributions.** We have proposed three models, namely, BraneExp[Jag+21a], BraneNet[Jag+22b], and BraneMF [Jag+22a] for multi-omics data integration.Our models learn integrated embeddings that can be utilized for several downstream tasks, including GRN inference, Protein function prediction, PPI prediction, and clustering of proteins.

The embeddings are learned by performing matrix factorization, approximating the spectrum of this PPMI matrix. BraneMF computes PPMI matrices for each layer and learns embeddings by using a joint singular value decomposition (SVD) framework. We demonstrate the applications of the learned embeddings to solve important downstream bioinformatics tasks, including Gene Regulatory Network (GRN) inference, clustering of biologically related bio-molecules, prediction of protein functions, and protein-protein interactions (PPIs). We have performed extensive analysis by comparing the performance of the developed models against several state-of-the-art integration methods for multilayer networks.

1.3 Outline of the Thesis

CHAPTER 2	**BACKGROUND** *We describe multi-omics data and its integration. We also summarize the related work section for the methods developed for multilayer network integration.*
CHAPTER 3	**MATERIALS AND METHODS** *We present our models BraneExp, BraneNet, and BraneMF, for multi-omics data integration.*
CHAPTER 4	**RESULTS AND DISCUSSION** *We apply the output of integration models for the downstream prediction tasks of interest. Additionally, we conduct an extensive evaluation and comparison of our models against state-of-the-art network integration methods.*
CHAPTER 5	**CONCLUDING REMARKS** *We conclude the dissertation and propose possible future research directions.*

Table 1.1: Outline of the thesis.

2

Background

This chapter provides detailed information about the fundamental aspects of molecular biology, multi-omics, and network science. Firstly, in Section 2.1, basic concepts about molecular biology are explained, including the flow of information from genotype (genetic material) to phenotype (observable characteristics. Secondly, in Section 2.1.2, the basic concepts of multi-omics studies are defined with an introduction to major omics data types. A concise description of multi-omics data production, pre-processing, and potential applications are also provided. In Section 2.2, network science basics are outlined, and, in Section 2.2.1, biological networks are introduced, and their basic properties are explained. Later, in Section 2.3, the booming field of Graph Representation Learning (GRL) and its applications in multi-omics data integration (Section 2.4) are described. In Section 2.5, the details about related works and the state-of-the-art methods are discussed. Lastly, in Section 2.6, the challenges and the limitations of data integration in biology are raised.

2.1 Molecular Biology Basics

Molecular biology is the scientific field concerned with the study of biomolecules that participate in the processes of biological phenomena that involve the basic units of life. It includes studying nucleic acids (e.g., DNA and RNA) and proteins that are essential to life to understand biomolecular interactions. Therefore to understand the biological relevance of omics data integration, one should be aware of the basic concepts of molecular biology.

2.1.1 Cell — a multiplex molecular machine

The molecular system of a living organism is made up of cells, the basic structural, functional, and biological unit. An organism's survival depends on the ability of cells to store, retrieve, and translate the genetic instructions required to make and maintain a living organism. This hereditary information (genetic

material) is passed on from a parent cell to its daughter cells at cell division and from one generation to the next through the organism's reproductive cells [Alb+15]. The genetic instructions are stored in the genome that determines the characteristics of a species as a whole and of the individuals within it. The genome includes chromosomal Deoxyribonucleic acid (DNA) as well as DNA in plasmids and (in eukaryotes) organellar DNA, as found in mitochondria and chloroplasts. Through a complex series of interactions, the DNA sequence directs the production of all of the Ribonucleic acid (RNA)s and proteins of the organism at the appropriate time and within the appropriate cells. Proteins serve a diverse series of roles in the development and functioning of an organism: they can form part of the structure of the organism; have the capacity to build the structure; perform the metabolic reactions necessary for life; and participate in regulation as transcription factors, receptors, key players in signal transduction pathways, and other molecules [KGK17]. This is the basic cellular machinery that is explained as *central dogma of molecular biology*. This theory was proposed by Francis Crick in 1970 [Cri70]. He described the flow of genetic information, which is stored in DNA sequences (genome) and transferred to RNA (transcriptome). Further proteins are synthesized, which leads to the determination of cellular phenotypes.

More precisely, in the genome, a gene is a DNA fragment carrying the instructions for making a protein. This information is encoded via a specific order of four nitrogenous bases that are A, T, C, G. It is the coding sequence that will be transcribed into messenger RNA (mRNA). In addition, a gene is also composed of a promoter containing an initiation sequence as well as regulatory sequences (enhancers and silencers). The promoter is located upstream of the coding sequence. Finally, at the end of the coding sequence, a terminator is found. When gene expression is promoted, the coding sequence is transcribed into a mRNA by an enzyme named RNA polymerase. Except for the nitrogenous base T, which is replaced by the nitrogenous base U, the mRNA conserves the same sequence of nucleic bases as the corresponding gene. The mRNA, after a maturation step, is translated into a polymer of amino acids. The synthesized polymer corresponds to the protein, and its amino acid sequence is dictated by the sequence of nitrogenous bases of the mRNA[Cri70].

The growing availability of data describing complex traits has challenged the central dogma of molecular biology (Figure 2.1) [Kar09; Alb+15]. A deeper investigation of biological mechanisms has uncovered complex molecular activities, namely, reverse transcription, epigenetic (environmental) regulation, post-transcription modifications, and post-translation modifications. On the basis of these discoveries, the flow of information from one level to the other is not unidirectional. Every element of the genome interacts directly or indirectly with many other genomic components. It has been seen that feedback cycles among molecular levels not only exist, but they can affect important biological processes [Alb+15]. Therefore, understanding the mechanisms in biological systems is extremely challenging. In order to study a biological system holistically, it is extremely necessary to obtain, integrate, analyze and interpret omics data of multiple molecular levels, e.g.,

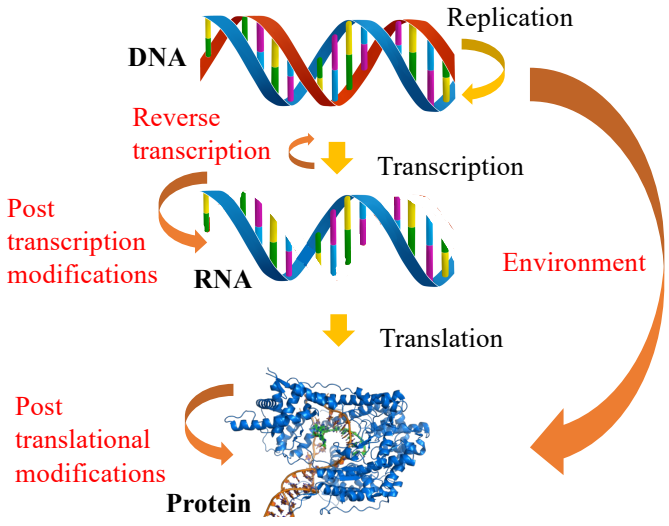


Figure 2.1: **Revised central dogma of molecular biology.**

genome, proteome, and transcriptome.

2.1.2 Multi-omics Paradigm

Multi-omics approaches enable us to study omics data holistically at various molecular levels. The different omic strategies employed during multi-omics are mainly genome, proteome, transcriptome, epigenome, and microbiome (Figure 2.2). Nevertheless, each type of omics data, on its own, provides a list of biomolecules associated with the phenotype. For example, genomics uses a combination of recombinant DNA, DNA sequencing methods, and bioinformatics to sequence, assemble, and analyze the structure and function of genomes. Transcriptomics is the study of complete sets of RNA transcripts that are produced by the genome, under specific circumstances or in a specific cell known as a transcriptome. Epigenomics is concerned with epigenetic modifications on the genetic material of a cell, known as an epigenome. Proteomics is the large-scale study of sets of proteins produced in an organism, system, or biological context known as a proteome. Metabolomics is the large-scale study of small molecules, commonly known as metabolites, within cells, biofluids, tissues, or organisms. Collectively, these small molecules and their interactions within a biological system are known as a metabolome [Man+18].

These massive complex multi-omics data can be useful to identify markers responsible for the observed phenotype and also to give insights into the biological pathways or the biological processes that are different in "test" (e.g., mutant/disease) versus "control" (e.g., wildtype/healthy). However, analysis of only one data type is limited to correlations, mostly reflecting reactive processes rather than causative ones. Integration of different omics data types is often expected to elucidate potential causative changes that lead to specific phenotypes or treatment targets. Such characterization and association studies of multi-omics data are

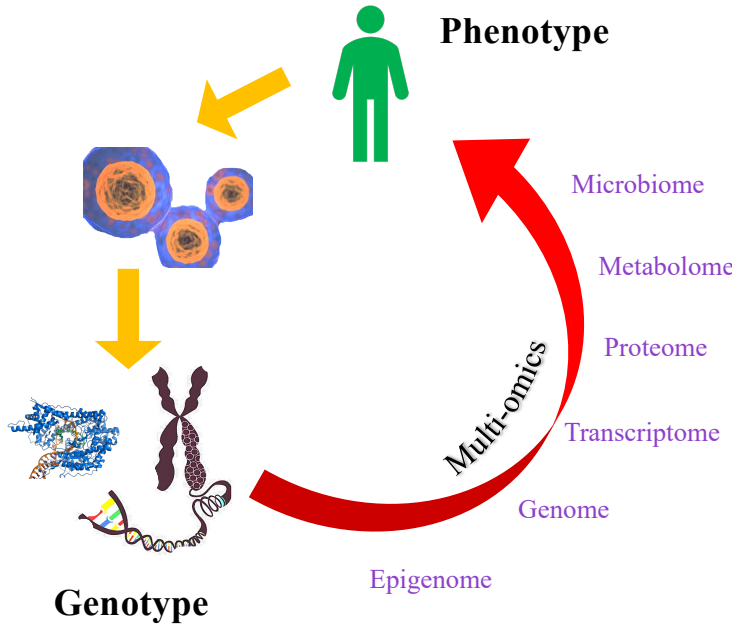


Figure 2.2: **Genotype-phenotype cycle.** The figure shows the cycle that goes around genotype and phenotype. For example, from a phenotype (could be a diseased patient), we obtain genetic information in the form of omics data. Then, we interpret the phenotype by analyzing and integrating these omics data.

often represented as *graphs* [Hub+07]. For instance, the nodes in these graphs can be a biomolecule, and the edges represent the relationship between them. These relationships can exhibit gene regulatory mechanisms, protein-protein interactions, or metabolic networks. The description of such interactions can be resolved by mathematical abstraction offered by graph theory. The beauty and usefulness of this abstraction allow us to develop the concepts and the tools to better understand the biological system. Within the field of biology, potential applications of network analysis [Hub+07] include the identification of biomarkers, predicting the role of proteins of unknown function, or discovering new regulatory pathways.

2.2 Network Science Basics

A network is a structure made up of a set of interacting nodes. Complex networks have been studied extensively to describe a wide range of disciplines, such as biology (e.g., protein interaction networks), information technology (e.g., telecommunication networks, internet), and the social sciences (e.g., collaboration, communication, economics, and political networks). Formally, networks are modeled as graphs [Wes+01].

Before going into the basics of graph theory, we would like to define the basic terms of set theory. Let S be a set defined as $S = \{s_1, s_2, \ldots, s_n\}$. Two sets S_1 and S_2 are said to be equal (written as $S_1 = S_2$) if every element of S_1 is a member

of S_2, and every element of S_2 is a member of S_1. If every member of set S_1 is a member of set S_2 (not necessarily vice versa) then the set S_1 is a subset of set S_2, written $S_1 \subseteq S_2$. Two sets S_1 and S_2 can be combined into a new set. The union of the sets $S_1 \cup S_2$ is the set of all objects that are members of either S_1 or S_2. The intersection of the sets $S_1 \cap S_2$ is the set of all objects that are members of both S_1 and S_2. An empty set is denoted by $\emptyset$. All sets used in this thesis are the set of natural numbers, including zero ($\mathbb{N}_0$), the set of integers ($\mathbb{Z}$), and the set of real numbers ($\mathbb{R}$).

Definition 2.2.1 (**Graph**). A graph G is described as $G = (\mathcal{V}, \mathcal{E}, \mathcal{W})$, where $\mathcal{V} = (v_1, \ldots, v_n)$ is the set of nodes, while $\mathcal{E} = \{e_{ij}; i, j = 1, \ldots, n\}$ is a set of edges connecting the nodes in $\mathcal{V}$. Edge e_{ij} represents the connection between nodes v_i and v_j and weight $w_{ij} \in \mathcal{W}$ can be associated with the edge e_{ij} to describe the strength of interaction (weight) between nodes v_i and v_j [Wes+01].

This definition describes undirected graphs, that is, graphs where connections between vertices are without a direction. Undirected graphs are used, for example, to model protein interaction graphs and correlation graphs. On the other hand, a directed graph or digraph is a graph $G = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ where $\mathcal{E}$ consists of the ordered pairs of nodes. These pairs (v_i, v_j) and (v_j, v_i) do not represent the same edge. It is also possible to have multiple edges for the same pair of nodes. This type of graph is called a multigraph. In Figure 2.3, the various types of graphs are illustrated. Note that the mentioned graphs can be weighted, unweighted, directed, or undirected [Wes+01].

Definition 2.2.2 (**Multilayer graph**). A multilayer graph of L-layers is a set $G = \{G_l\}_{l=1}^{L} = \{(\mathcal{V}_l, \mathcal{E}_l)\}_{l=1}^{L}$ of graphs, where $\mathcal{V}_l := \{v_1, \ldots, v_{|\mathcal{V}|_l}\}$ and $\mathcal{E}_l := \{e_{1_l}, \ldots, e_{M_l}\}$ are the nodes and the edges (undirected) respectively. N_l and M_l denote the number of nodes and edges for each layer.

Definition 2.2.3 (**Adjacency matrix**). A graph G can be represented by its adjacency matrix $\mathbf{A}$ that maps the association between the $\mathcal{V}$. If a graph has n number of nodes, then the adjacency matrix of that graph of dimension $n \times n$. If there is an edge e_{ij} between two nodes i and j, the matrix element a_{ij} is 1 or w_{ij} (in case of weighted graphs). For the absence of an edge, a_{ij} is 0 [Wes+01].

In undirected graphs, the adjacency matrix is symmetric. In the case of directed graphs, the matrix is not symmetric, thus differentiating its upper triangular part from its lower triangular part (a_{ij} is not the same as a_{ji}) [Wes+01].

Definition 2.2.4 (**Subgraph**). A subgraph $G' = (\mathcal{V}', \mathcal{E}')$ of the graph $G = (\mathcal{V}, \mathcal{E})$ is a graph where $\mathcal{V}' \subseteq \mathcal{V}$ and $\mathcal{E}' \subseteq \mathcal{E}$ respectively.

Definition 2.2.5 (**Node degree**). The degree of a node v is the number of connections that v has to other nodes in the network. In the case of a directed graph, the out-degree of v refers to the number of directed edges incident from v, whereas the in-degree of v refers to the number of directed edges incident to v [Wes+01].

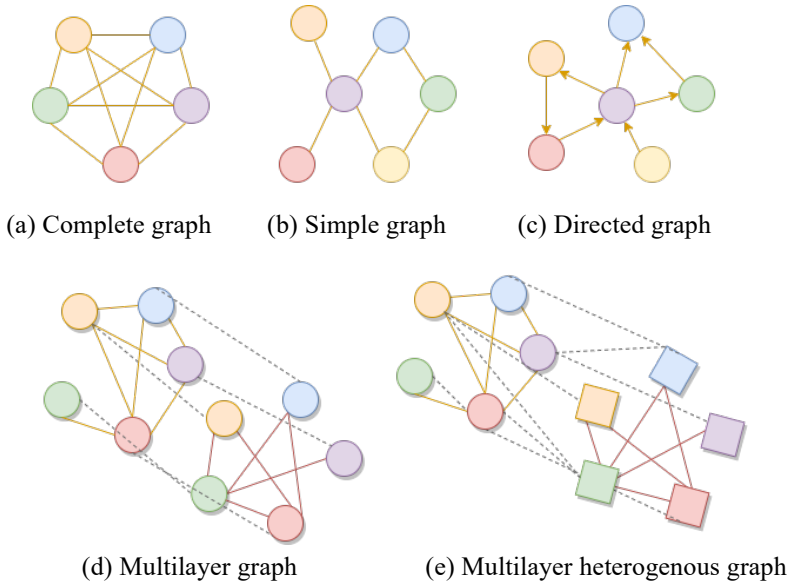
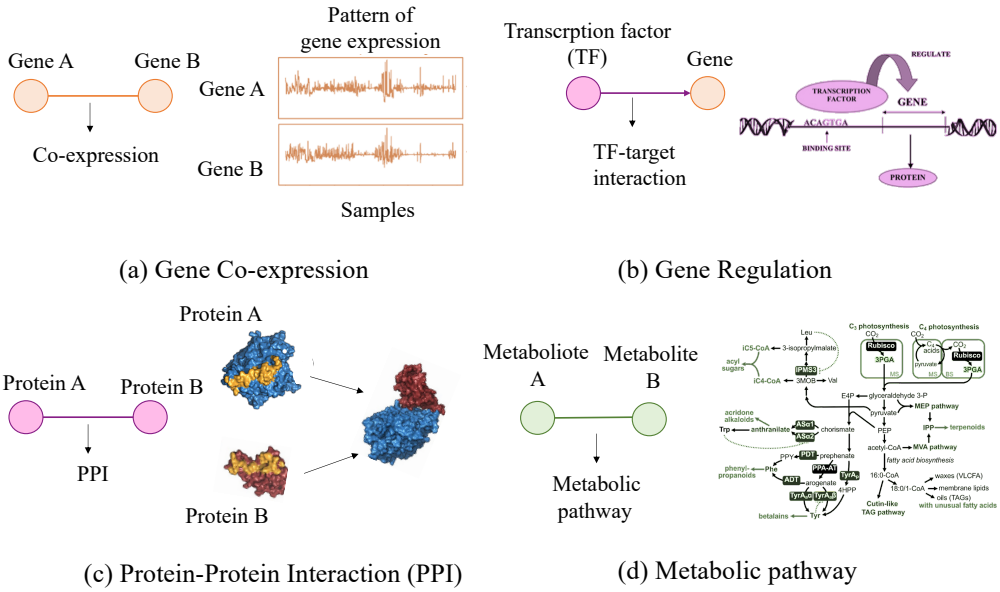


Figure 2.3: **Illustration of different graph types**. (a) A complete graph is a simple, undirected graph in which every pair of distinct nodes is connected by a unique edge. (b) A simple graph is an undirected graph without loops and multiple edges, whose edge set is a subset of a complete graph. (c) A directed graph, also called a digraph, is a graph in which the edges have a direction, i.e., they indicate an orientation. (d) A multilayer graph has different graph layers that share the same set of nodes but different types of edges. (e) In a multilayer heterogeneous graph, the graph layers are composed of different types of nodes and edges.

Definition 2.2.6 (**Network density**). The density of a graph is a measure of how many edges between nodes exist compared to the possible number of edges. The density of an undirected graph is calculated as $\frac{|\mathcal{E}|}{|\mathcal{V}|(|\mathcal{V}|-1)/2}$, where $|\mathcal{V}|$ and $|\mathcal{E}|$ is total number of nodes and edges respectively. In the case of a directed network, there is no need to divide the numerator by two. As such, the density for directed network is $\frac{|\mathcal{E}|}{|\mathcal{V}|(|\mathcal{V}|-1)}$.

Definition 2.2.7 (**Walk**). A walk is a sequence $v_1, e_{12}, \ldots, v_k$, of v_i nodes and e_i edegs such that for $1 \leq i \leq k$, the edge e_i has endpoints $v_{(i-1)}$ and v_i. A walk can be open (starting and ending nodes are different) or closed (the walk starts and ends at the same node). If all edges of a walk are distinct, then the walk is called a path. The length of a walk (w) is given by the number of edges required to reach v_k starting from v_1 [Wes+01].

The networks found in the real world are often claimed to be scale-free, meaning that the fraction of nodes with degree k follows a power law $k^{-\alpha}$, a pattern with broad implications for the structure and dynamics of complex systems (e.g., biological networks)[BA99]. The power law, also called the scaling law, states that a relative change in one quantity results in a proportional relative change in another. The most notable characteristic of a scale-free network is the relative commonness of nodes with a degree that greatly exceeds the average.



(a) Gene Co-expression

(b) Gene Regulation

(c) Protein-Protein Interaction (PPI)

(d) Metabolic pathway

Figure 2.4: **Biological networks.** (a) Gene co-expression: gene pairs show co-ordinated expression patterns across a group of samples. (b) Gene regulation (e.g., by TF): an edge between a TF and its target gene represents transcriptional regulation. (c) Protein-protein interaction: proteins are connected by an edge if they physically interact. (d) Metabolic pathway: metabolites are connected when they belong to the same metabolic pathway.

The highest-degree nodes are often called "hubs". The emergence of the hubs is a consequence of a scale-free property of the network. They have a significant impact on the network topology and can be found in many real networks, such as biological networks. [Wes+01].

2.2.1 Biological networks

Over a few decades, the in-silico study of biological systems has been done using networks. For instance, nodes in biological networks can be any biomolecules (e.g., genes, proteins, or metabolites) and edges represent the relationship between them [Pržl9]. These relationships can exhibit gene regulatory mechanisms and protein-protein interactions. The description of such interactions can be resolved by using concepts from graph theory. This allows the development of tools to better understand biomolecular relationships. Networks represent the molecular-level patterns of interaction and the mechanisms of control in the biological cell. Majorly, these networks are gene co-expression networks, genetic regulatory networks (GRN), protein-protein interaction (PPI) networks, and metabolic networks [Pržl9]. In Figure 2.4, various types of biological networks are shown with a brief description.

Gene Co-expression Networks (GCN)

A gene co-expression network identifies the genes that have a tendency to show a coordinated expression pattern across a group of samples obtained from transcriptomics data (Figure 2.4a). The co-expression network can be represented as a gene-gene similarity matrix whose elements are the co-expression values defined based on correlation measures or mutual information between each pair of genes. These relationships describe the similarity between expression patterns of the gene pair across all the samples [Van+18a]. Alternatively, least absolute error regression or a Bayesian approach can be used to construct a co-expression network [Van+18a]. The latter two approaches have an added value since they can be used to identify causal links [Van+18a].

After the construction of a co-expression network, modules (also known as clusters of co-expressed genes) are identified using one of several available clustering techniques [Oye+16]. Clustering in co-expression analyses is used to group genes with similar expression patterns across multiple samples to produce groups of co-expressed genes rather than only pairs. Modules can subsequently be interpreted by functional enrichment analysis, a method to identify and rank enriched functional categories such as biological process, molecular function, and cell in a list of genes [Hub+07; Van+18a].

Gene Regulatory Networks (GRNs)

Gene regulatory mechanisms are used by cells to increase or decrease the production of specific gene products (i.e., proteins). A gene can be regulated at different levels of omics, i.e., epigenetic regulation, transcriptional regulation, and post-translation modifications. This could be positive regulation (turning on gene expression), negative regulation (turning off gene expression), and co-regulation (turning multiple genes on or off together). Gene regulatory networks (GRNs) are mathematical models of such regulatory mechanisms. The nodes in a GRN are genes and regulators that are connected by edges representing regulatory relationships. The GRNs are difficult to infer and mostly misunderstood [EDH14].

Cellular differences are determined by the expression of different sets of genes. For instance, a cancer cell acts differently from a normal cell since it expresses different genes. Interestingly, in eukaryotes, the default state of gene expression is OFF rather than ON, as in prokaryotes [Hoo08]. Why is this the case? The secret lies in chromatin. It is made up of DNA and histone proteins located in the cell nucleus. Histones are among the most evolutionarily conserved proteins, vital to the well-being of eukaryotes and tolerating little change. If a certain gene is closely linked to histones, this gene is "turned off". But how do eukaryotic genes manage to escape this "silencing"? This is where the histone code comes into play. This code involves modifications of the positively charged amino acids of histones to create some areas where the DNA is more open and others where it is very tightly bound. DNA methylation is a mechanism that appears to be coordinated with histone modifications, particularly those that lead to the repression of gene

expression. Small non-coding RNAs such as RNAi may also be involved in the regulatory processes that form "silent" chromatin. When the tails of histone molecules are acetylated at certain sites, these molecules have less interaction with DNA, making it more open [Hoo08; EDH14].

The area surrounding a potential transcription zone, also known as the promoter region, must be unraveled before transcription can begin. This is a difficult process that requires the coordination of histone modifications, transcription factor binding, and other chromatin remodeling processes. Specific DNA sequences are then accessible for specific proteins to bind once the DNA has been opened. Many of these proteins are activators, while others are repressors; all of these proteins are referred to as transcription factors in eukaryotes (TFs). Each TF has a DNA binding domain and an effector domain that recognizes a $6-10$ base-pair motif in the DNA. If a TF binds to its matching pattern in a fragment of DNA, investigators can identify its footprint (Figure 2.4b).

Protein-Protein Interaction (PPI) networks

The primary mode of protein-protein interaction is physical. The complicated interlocking folded shapes of proteins create so-called protein complexes but without the exchange of particles that define chemical reactions [Hub+07]. In a protein-protein interaction network, two nodes (i.e., two proteins) are connected by an undirected edge if the corresponding proteins form a complex structure (Figure 2.4c). The interactions that involve three or more proteins are represented by multiple edges. [Van+18a].

Metabolic networks

Metabolism is the chemical process by which cells break down food and nutrients into usable building blocks and then reassemble those building blocks to form biomolecules. Typically, this breakdown and reassembly involve chains or pathways, sets of successive chemical reactions that convert initial inputs into useful end products through a series of reaction steps. The complete set of all reactions in all pathways forms the metabolic network. The nodes in a metabolic network are chemical compounds produced and/or consumed by the reactions, also known as metabolites. They are small molecules like carbohydrates, lipids, as well as amino acids and nucleotides. The metabolites consumed are called the substrates of the reaction, while those produced are called the products [Hub+07; Van+18a]. The edge between two nodes represents the participation of both metabolites in the same reaction, either as substrates or as products.

After constructing such biological networks, as mentioned above, from omics data, network analysis is performed. The potential applications of such network analysis include the identification of biomarkers, determining the role of proteins of unknown function, the discovery of new regulatory pathways and sample classification (e.g., disease or healthy) [Hub+07; Pržˇ19].

2.3 Graph Representation Learning

High-dimensional graph data often comes in irregular forms. They are more difficult to analyze than image/video/audio data defined on regular lattices [Che+20]. Various graph embedding techniques have been developed to convert the raw graph data into a low-dimensional vector representation while preserving the intrinsic graph properties. Graph representation learning methods [HYL17b; Che+20; Yue+20] aim to generate vector representations for various types of graph elements such that the learned representations, i.e., embeddings, capture the structure and semantics of a rich, graph-structured or networked dataset.

The main challenge in machine learning on networks is finding a way to extract information about interactions between nodes (e.g., node similarity) and incorporate that information into a machine learning model. To extract this information from networks, classical machine learning approaches rely on summary statistics (e.g., degrees or clustering coefficients) or carefully engineered features to measure local neighborhood structures (e.g., network motifs). In contrast to classical approaches, representation learning approaches encodes network structure into low-dimensional representations, using transformation techniques based on deep learning and nonlinear dimensionality reduction [HYL17b].

Figure 2.5 shows the illustration of GRL on multilayer graphs. GRL methods take a graph as the input, where the graph can be homogeneous, heterogeneous, single/multilayer, with/without auxiliary information. The output of a graph embedding method is a set of vectors representing the input graph. It could be node embeddings, edge embeddings, or the embeddings of the whole graph. The preferred output form is application-oriented and task-driven. GRL encompasses a wide range of methods, including graph theoretic techniques rooted in classic network science, manifold learning, topological data analysis, graph neural networks, and generative graph models.

Classical methods are categorized into linear and nonlinear. The linear methods include Principal Component Analysis (PCA) [AW10], Linear Discriminant Analysis (LDA) [Ize13], and Multidimensional Scaling (MDS)[Sae+18]. These methods are referred to as "subspace learning" [Yan+06] under the linear assumption. However, linear methods might fail if the underlying data are highly non-linear [Sau+06]. Then, non-linear dimensionality reduction (NLDR) [DC92] can be used for manifold learning. The objective is to learn the nonlinear topology automatically. The NLDR methods include Isometric Feature Mapping (Isomap) [SMR06], Locally Linear Embedding (LLE) [RS00], and Kernel Methods [Har+11].

Random walk-based methods are popular methods to perform GRL [PAS14; GL16]. They sample a graph with a large number of paths by generating a set of random walks starting from each node in the graph [HSS21]. This path is a Markov chain over the set of nodes $\mathcal{V}$. Random walks indicate the context of

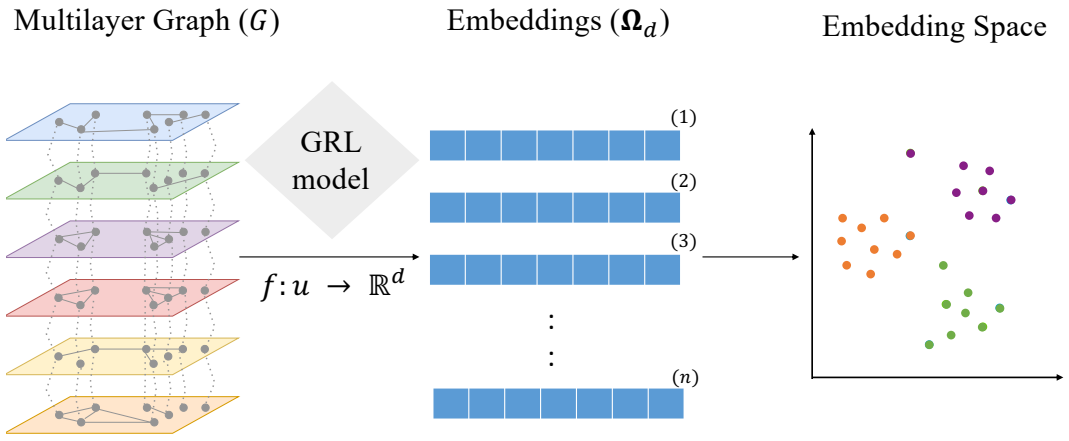


Figure 2.5: **Illustration of Graph Representation Learning.** A multilayer graph (G) is given as input to a GRL model, and d-dimensional embeddings ($\mathbf{\Omega}_d$) are learned in such a way that similar nodes are close in the embedding space.

connected vertices. The randomness of walks gives the ability to perform locally as well as global explorations by walking through neighboring vertices. After that, the probability models [Mik+13; Rud+16] can be applied on these randomly sampled paths to learn the node representations. Figure 2.6 shows the standard workflow of learning node embeddings using random walk-based GRL models. The transition probability for a node to traverse to another node is computed through a path given by a walker jumping to node v from u and is characterized by the adjacency matrix $\mathbf{A}$.

Let the vector $p_t \in \mathbb{R}^{\mathcal{V}}$ denotes the probability distribution at time t. $p_t(u)$ indicates the value of p_t at node u—that is the probability of being at node u at time t. A probability vector p is a vector such that $p(u) \geq 0$, for all $u \in \mathcal{V}$, and $\sum p(u) = 1$. Our initial probability distribution, p_0, will typically be concentrated on one node. That is, there will be some node u for which $p_0(u) = 1$. In this case, we say that the random walk starts at u. Thus, $p_{(t+1)}(u)$ is given as:

$$p_{(t+1)}(u) = \sum_{v:(u,v\in\mathcal{V})} \frac{\mathbf{A}_{uv}}{\deg(v)} p_t(v), \tag{2.1}$$

where $\deg(v)$ is the degree of node v. The transition probabilities between all pairs of nodes are represented by a transition matrix $\mathbf{P} \in \mathbb{R}^n$: $\mathbf{P} = \mathbf{D}^{-1}\mathbf{A}$, where $\mathbf{D}$ is the diagonal degree matrix. Standard random walks provide a natural way to capture node neighborhoods in undirected connected graphs. One can also design biased random walks to explore different notions of neighborhood [GL16; NM18]. For directed graphs, a PageRank process [Pag+99] is often applied in lieu of standard random walks to guarantee ergodicity. Few examples of popular random-walk-based methods include DeepWalk [PAS14], node2vec [GL16], and exponential family graph embeddings (EFGE) [CM20a].

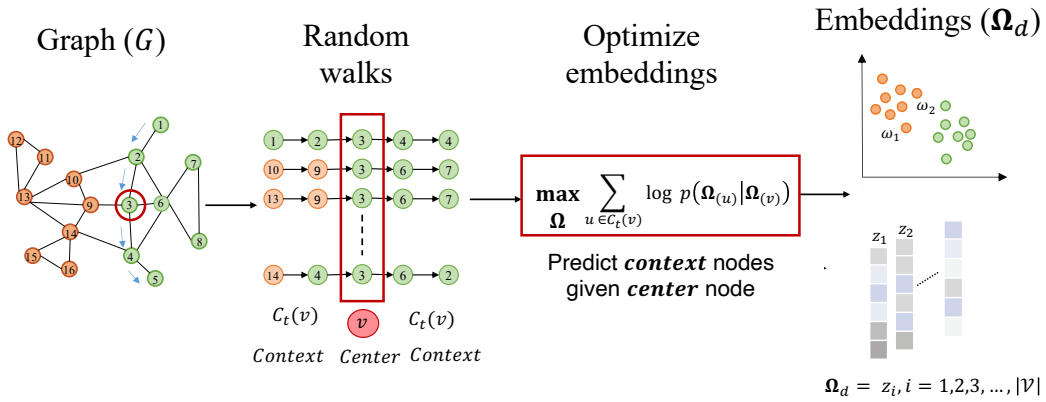


Figure 2.6: **Illustration of graph representation learning using random walks**. From graph G, n random walks are sampled of length w. In window size t, for each center node, the context is defined by its neighboring nodes. The embeddings are learned by maximizing the likelihood of the random walk co-occurrences. The encoder is then trained to learn $\mathbf{\Omega}_d$ embeddings.

Matrix factorization-based methods are popular approaches for computing node embeddings. One of the first such approaches was Graph Factorization (GF) [Ahm+13]. Several variations have been introduced in the literature. GraRep [CLX15] defines different loss functions for capturing the different k-step local relational information. It then optimizes each model with matrix factorization techniques and constructs the global representations for each node by combining different representations learned from different models. HOPE [Ou+16] is an efficient model to preserve higher-order proximities of large-scale graphs, capable of capturing asymmetric transitivity. Finally, NetMF [Qiu+18] unifies LINE, DeepWalk, and node2vec in the framework of matrix factorization, where the factorized matrices have closed forms.

Neural network models have recently become popular. Being inspired by the success of recurrent neural networks (RNN) and Convolutional Neural Networks (CNN), researchers attempt to generalize and apply them to graphs. Natural Language Processing (NLP) models often use RNNs to find a vector representation for words. The Word2Vec [Mik+13] model aims to learn the continuous feature representation of words by optimizing a neighborhood-preserving likelihood function. Another family of neural network-based embedding methods adopts CNN models. The input is either a set of paths sampled from a graph or the whole graph itself [Zho+20]. Some apply the original CNN model designed for the Euclidean domain and reformat the input graph to fit it [Zho+20]. Other approaches generalize deep neural models to graphs. Popular neural network-based methods for graph embeddings include Graph Convolutional Network (GCN) [KW16], Graph Attention Networks (GATs) [Vel+17], Variational Graph Auto-Encoders (VGAE) [KW13], GraphSAGE [HYL17a] and Structural Deep Network Embedding (SDNE) [WCZ16].

2.4 Multi-omics Data Integration with GRL

Multi-omics data enable us to gain more accurate insights into the biological data and can be useful to make effective predictions of the unknown biological mechanisms (Section 2.1.2). Although a single omic study can identify molecules and biomarkers of the main pathologies or experimental conditions, it can provide only partial information. To unravel the hidden information, one shall effectively integrate multi-omics data [Sub+20].

Multi-omics data integration was proposed as a combination of methods to fuse data obtained from different omic approaches, aiming at gaining insights on the interconnectedness of the different biomolecules (e.g., proteins, RNAs, metabolites) and the flow of biological information that occurs within them. Network approaches have generated substantial interest based on their potential for integrative omics analysis and are expected to facilitate a new era of systems biology [Sub+20].

The greater availability of data has allowed many multi-omics studies and fostered the expansion and construction of public databases to ensemble the greatest amount of data in standardized file formats and user-friendly interfaces. The Ensemble Genome Project and the Human Proteome Project aimed at collecting the major genes and proteins underlying the main biological processes in the cell [Leg+11; Hub+02]. In such repositories, the main multi-omics data are RNA-Seq, DNA-Seq, miRNA-Seq, SNV, CNV, DNA methylation, proteomics, whole genome sequencing, and the genomic variations data (somatic and germ- line mutation). The major omics data types are defined in Section 2.1.2.

In the last decade, the availability of such an amount of data and information has led to various methodologies and algorithms for their analysis [DBB09; Jud+16; Pic+21]. Concerning single omic dataset processing, the two most common types of analysis are:

- Extraction of the most relevant features for the detection of new biological signatures or pathways.
- Classification and clustering of samples to create predictive models for pathology or to discover new molecular subtypes.

In a multi-omics scenario, these two approaches are still valid, but the algorithms used to integrate and analyze the data need to be properly modified and optimized. Several articles have reviewed the state-of-the-art for multi-omics integration [Sub+20; Pic+21; Lov+22; Can+21], focusing on the sample clustering problem (e.g., clustering of patients and healthy controls). Below, the various multi-omics methods are grouped into four main categories [Lov+22]:

- **Graph-based**: Description of samples in the form of graphs or similarity matrices.
- **Dimensionality reduction-based**: The integration is given by the joint reduction of the dimensionality among the various omics.
- **Statistics-based**: The prevailing approach for the integration is based on Bayesian models.
- **Neural Networks-based**: Techniques based on the creation of artificial neural networks and deep learning methods to integrate multi-omics data for building predictive models to understand the pathology and discover new molecular subtypes/signatures.

Biological networks reveal different types of information based on the input data, mathematical model, and the type of relationship we want to study. For example, Bayesian networks allow the use of prior information to capture conditional dependencies between probabilistic events [Pee05]. These probabilities are thus used to define relationships between nodes. Another approach is text mining, where networks are built based on scientific publications, on the assumption that molecules likely to interact share contextual information [Haa+09]. Finally, gene correlation networks, such as those generated by the Weighted Correlation Network Analysis method (WGCNA) [LH08], are based on the correlation between gene pairs over a large number of samples. Several approaches have been proposed to learn node representation for multilayer graphs which are briefly discussed in Section 2.5. Nevertheless, most of these multi-omics data integration methods are complicated and computationally expensive on real multi-omics data. We are interested in developing less complex frameworks to learn integrated embeddings from multi-omics data targeting several applications in bioinformatics.

2.5 Related Works

A plethora of GRL approaches are based on random walks [PAS14; GL16; NM18; ÇM20], matrix factorization [LG14; Qiu+18], or neural networks [HYL17b; Yue+20]. Inspired by Word2Vec-based single-layer network embedding techniques [Mik+13; PAS14; GL16], a few GRL methods have been proposed for multilayer networks.

Principled Multilayer Network Embedding (PMNE) [Liu+17] is an extension of a single-layer graph embedding to a multilayer network. It proposes approaches for multilayer graph mining techniques that can be applied to any graph embedding method developed for single-layer graphs. Multiplex Network Embedding (MNE) [Zha+18] is a multilayer network embedding method that generates random walks for each layer and then applies the Skip-Gram model [Mik+13] to learn joint embeddings for each node. The final node embedding is composed of three

parts: common embeddings, relation-based embedding, and transformation matrix. Multi-Net [BK18] uses random walks, namely classical, diffusive and physical, to obtain node sequences [Sol+16; Guo+16]. Then, it merges the node's neighborhood (context) and learns a d-dimensional feature vector for each node by maximizing the likelihood of the occurrence of node neighbors across all layers. Multi-node2vec (Multi-n2v) [Wil+18] extends NODE2VEC to multilayer networks. The model collects a bag-of-words from each layer by performing a vertex neighborhood search. Then, the optimization procedure computes the features by using the Skip-Gram neural network model on the identified neighborhoods. Recently, MultiVERSE [Pio+21] computes a similarity matrix using random walk with restart (RWR). Then, it applies an optimized version of VERSE [Tsi+18], a vertex-to-vertex similarity-oriented embedding method, to compute the representations. FAME [Liu+20] decomposes the heterogeneous multiplex network into homogeneous and bipartite sub-networks. Then, it uses a spectral transformation module to automatically aggregate and decouple sub-networks with the exploration of their multi-relational topological signals.
For biological networks, Similarity Network Fusion (SNF) [Wan+14] constructs a similarity network for each data type and then iteratively integrates these networks using a network fusion methodology. Mashup [CBP16] is a network integration framework based on matrix factorization that builds compact low-dimensional vector representations of proteins. It takes as input a collection of PPI networks and generates embeddings that best explain their wiring patterns across all networks. OhmNet [ZL17] is an unsupervised feature learning approach for multilayer networks with a predefined hierarchy describing relationships between the layers. deepNF [GBB18] is a network fusion method relying on multimodal deep autoencoders (MDA). It takes as input a collection of PPI networks and applies a neural network, an autoencoder (AE), that is composed of two sections. First, it is the encoder part, in which the input data is transformed to low-dimensional features; second, the decoder, where features are mapped back to the input data [Vin+10]. To capture the structural properties of networks, deepNF and Mashup are based on Random Walks with Restarts (RWR). The vectors learned from both methods are then fed into a support vector machine (SVM) classifier to predict functional classes of proteins. More recently, deepMNE-CNN [Pen+21] has been introduced, a multi-network embedding approach that applies a semi-autoencoder-based model to learn protein features. MOSS [Gon+22] performs a sparse singular value decomposition (sSVD) to learn embeddings. Graph2GO [FGZ20] extends variational graph autoencoders (VGAE) to multilayer networks. It establishes to integrate networks derived from heterogeneous information, including sequence similarity, protein-protein interaction, and protein features, amino acid sequence, sub-cellular location, and protein domains.

Most of these network integration frameworks are engrossed towards Gene Ontology (GO) prediction. The GOs are developed to systematically describe the functional properties of proteins to facilitate the computational prediction of their functions [Con04]. They serve as the gold standard and main source in functional proteogenomics. If two proteins have a similar function, apart from

their direct relationship in the network, they can have many further characteristics in common, such as biological processes, molecular function, cellular location, regulated by the same transcription factor, have the same epigenetic mark or belong to the same metabolic pathway. In order to determine such similarities between *a priori* unlinked proteins, it is necessary to obtain an informative representation of proteins and their proximity that is not fully captured by handcrafted features directly extracted from the PPI network. GRL-driven models are candidates for the above tasks. Given a multilayer network, GRL algorithms can embed it into a new compact vector space in such a way that both the original network structure and other latent features are captured. Indeed, existing methods are challenged when applied to biological datasets that demand comprehensive handling of data heterogeneity. Also, existing GRL methods for multilayer networks depend on numerous parameters—thus being computationally intensive in finding optimal parameter settings. Besides, biologists generally dispose of low levels of ground truth. To efficiently search for appropriate ground truth when biological information is not fully known becomes a difficult and time consuming task. Hence, there is a huge scope to develop new methods that can address these challenges. In this study, we derive embeddings purely via a data-driven fashion such that the probability of the context of a protein is maximized.

From the above-mentioned methods, we have selected eight multilayer network integration methods as our baseline models, namely SNF [Wan+14], Mashup [CBP16], deepNF [GBB18], MultiNet [BK18], Multi-node2vec [Wil+18], OhmNet [ZL17], MultiVERSE [Pio+21], and Graph2GO [FGZ20].

2.6 Data Integration Challenges in Biology

Data integration in the life sciences is a persistent task that has just recently emerged as a significant difficulty, in part due to technological advancements providing more data of all kinds and in larger quantities. Despite the fact that publicly available and well-maintained data repositories adequately support the availability of genomics data (with the pertinent exception of clinical data), there is a need for improved (and novel) annotation standards and requirements in data repositories to facilitate better integration and reuse of publicly available data [Zit+19]. The data exploitation aspect of data integration is probably the one that requires the most attention, as it involves:

- Use of prior knowledge—and its efficient storage.
- Development of statistical methods to analyze heterogeneous data sets.
- Creation of data exploration tools that incorporate both useful summary statistics and new visualization capabilities.

It has been observed that this field demands user-friendly tools targeting the integration of heterogeneous datasets and the relevance of integrative omic

studies. Moreover, efficient data integration in life sciences may require specific skills. This could be challenging when the significance of multi-omics data in a particular cellular system is hazy. Another aspect of data integration challenges is the impact of big data analytics in the life sciences. The term big data intuitively describes a situation present in many research fields: the amount of data generated by instruments is exploding and in many cases doubling over short periods of time. Biology is not an exception: "since 2008, genomics data is outpacing Moore's Law by a factor of 4" [ODS13]. This situation results in the requirement for developing scalable infrastructures able to manage these quantities of data while making it available for efficient access and indexing.

Multi-omics data integration to produce a comprehensive understanding of biological processes and pathologies does have its share of challenges. Efficient integration, analysis, and interpretation is a difficult undertaking due to the underlying variability in individual omics data, massive data sets requiring computationally expensive analysis, and a lack of research that aid in prioritizing the varied collection of tools [Sub+20]. Since multi-omics data are produced on so many different platforms, the formats and storage of the data vary greatly. Individual omics data must be pre-processed because the majority of multi-omics integrative analysis tools require data to be in specified formats (most commonly in a "features × samples" matrix). Data filtering, systematic normalization, batch effect elimination, and quality checks are all part of the pre-processing stage. Due to their significant impact on the integrative analysis, these pre-processing procedures must be used cautiously. Most of the integrative approaches are computationally intensive. It is, therefore, necessary to limit the size of the input data sets by filtering the noise and lowering the number of features that go into integrative models. However, choosing adequate filtering criteria might be difficult because there are not any global standards [Sub+20].

The effective processing of big data sets must be taken into consideration while developing new integrative approaches and tools. The appropriate selection of techniques that can address the relevant biological topic is the cornerstone of any integrative study. There are studies that benchmark integrative tools [RS18; Tin+19; Cha+20], but they fall short in terms of the selection of tools in the context of the relevant biological inquiry. To help the community better comprehend the vast range of tools, further in-depth research is required. Another dimension that could add value to multi-omics data interpretation is clinical information. Currently, there are no robust methods to integrate omics data with non-omics data, such as clinical metadata. The recent advances in this field are progressing primarily with efforts to reduce the challenges. Further developments in integrative analysis of multi-omics data must aim to ease the interoperability of multiple data sets and to develop a framework that can help in the seamless analysis of multi-omics data.

3

Materials and Methods

This chapter provides detailed information about the datasets, proposed models, and downstream tasks. Firstly, in Section 3.1, information about data acquisition and an overview of the datasets employed in this thesis are provided. Secondly, in Section 3.2, all three proposed models, viz. BraneExp, BraneMF, and BraneNet are explained with their mathematical formulation. Lastly, in Sections, 3.6 and 3.7, all the downstream tasks and metrics utilized to evaluate the performance of models are provided.

3.1 Datasets

Yeast has been a popular model organism for basic biological research. It is one of the simplest eukaryotic organisms, but many essential cellular processes are the same in *yeast* and *humans* [Nie19]. At IFP energies nouvelles, biologists work on various fungi and micro-organisms in the context of green chemistry, such as bioethanol production. For this purpose, we choose the datasets of a well-studied *yeast* model organism, i.e., *Saccharomyces cerevisiae*. We test the proposed models (BraneExp, BraneMF, and BraneNet) on *yeast* multi-omics datasets.

3.1.1 Multilayer Protein-Protein Interaction (PPI) network

Protein–protein interaction (PPI) networks are an important ingredient for the system-level understanding of cellular processes. Such networks can be used for filtering and assessing functional genomics data and for providing an intuitive platform for structural, functional, and evolutionary annotations of proteins. Exploring the PPI networks can suggest new directions for future experimental research [Sch+09; Szk+20]. The STRING database (https://string-db.org/) is a public repository of PPIs that contains information from numerous omics data sources, including experimental, co-expression, conserved neighborhood, fusion, and databases. It is freely accessible and regularly updated.

For this study, we obtain six PPI networks for *yeast* from STRING database [Szk+20], namely *Neighborhood*, *Fusion*, *Co-occurrence*, *Co-expression*, *Experimental*, and *Database*. The PPI networks are built for 6,691 proteins. The overview of the *yeast* STRING PPI networks used in our study is given in Table 3.1. From these six networks, we construct a multilayer PPI network G where each layer of G contains a PPI network obtained from a different data source (Figure 3.1). More formally, a multilayer network of L-layers is a set of $G = \{G_l\}_{l=1}^{L} = \{(\mathcal{V}_l, \mathcal{E}_l)\}_{l=1}^{L}$ graphs, where $\mathcal{V}_l := \{v_{1_l}, \ldots, v_{|\mathcal{V}|_l}\}$ and $\mathcal{E}_l := \{e_{1_l}, \ldots, e_{M_l}\}$ are the vertex and the edge sets, respectively. $|\mathcal{V}|_l$ and M_l denote the number of nodes and edges for each layer.

Throughout the thesis, we assume that the layers share the same set of nodes, so $\mathcal{V}_l = \mathcal{V}_j = \mathcal{V}$ for every $1 \leq l < j \leq L$. However, in practical case the existed PPI networks are of different node sizes. In such case, we build the per layer adjacency matrix by taking union of nodes in all layers ($|\mathcal{V}|_1 \cup |\mathcal{V}|_2 \cup \ldots \cup |\mathcal{V}|_L$). Nevertheless, the integrated embeddings are learned only for the nodes shared in all the layers ($|\mathcal{V}|_1 \cap |\mathcal{V}|_2 \cap \ldots \cap |\mathcal{V}|_L$). All edges are weighted and undirected. Note that nodes that do not share an edge remain isolated in the graph. We use $\mathbf{A}^{(l)}$ to denote the adjacency matrix of the associated layer G_l.

Network	Nodes	Edges	Density	Evidence
Neighborhood	1,324	7,656	0.008741	Gene order and sequence homology
Fusion	500	492	0.003943	Orthology and fusion
Cooccurrence	799	1,231	0.003861	Orthology
Coexpression	4,069	54,317	0.006562	Gene expression data
Experimental	4,149	48,190	0.005600	Biochemical, biophysical genetic experiments
Database	3,136	29,231	0.005946	Human curation

Table 3.1: **Overview of the *yeast* STRING PPI networks used in the study.** The above table shows the number of nodes and edges in the respective PPI networks with their density and the sources of each network.

3.1.2 Yeast multi-omics data

Apart from PPI networks, we have used multi-omics datasets to test our models. These datasets were obtained from various bioinformatics data sources.

Genome sequence

The genome sequence of *Saccharomyces cerevisiae* was obtained from the SGD genome database [Che+98]. The genome is approximately 12 Mb, organized in 16

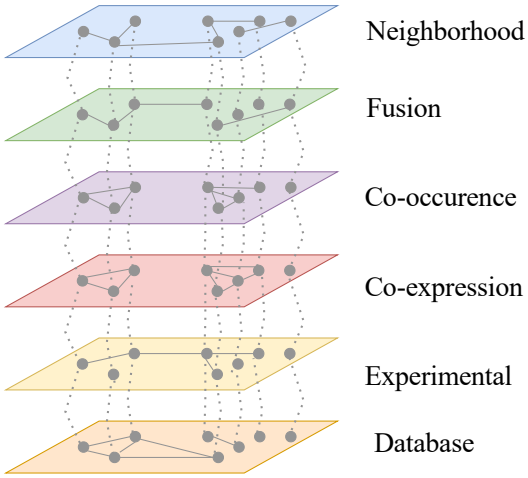


Figure 3.1: **Illustration of the multilayer PPI network built from STRING database**. All the layers share the same nodes. These nodes are connected by inter-layer dotted lines.

chromosomes and approximately 7,000 protein coding genes.

Transcription factor binding sites (TFBSs)

The set of TFBS profiles was obtained from the JASPAR database [For+20]. A total of 194 TFBS profiles were found in this database.

Transcriptomics

Transcriptomics data for *Saccharomyces cerevisiae* for 61 experimental studies was obtained from NCBI-GEO database [Bar+12].

Yeast heat shock multi-omics data

From a recently published data descriptor, we have obtained multi-omics datasets [Nuñ+20]. These datasets present three basic layers of the transcriptional circuit, including one type of epigenetic modification (H4K12ac mark for identification of active promoters obtained from ChIP-Seq), gene expression (RNA-seq), and targeted metabolomics (NMR). The dataset is comprised of 7,126 genes, 1,970 H4K12ac peaks, and 37 metabolites.

3.1.3 Functional annotations

Omics technologies have made it clear that a large fraction of the gene/proteins specifying the core biological functions are shared. Gene ontologies (GO) serve the knowledge of the biological role shared among such genes/proteins. They are developed to systematically describe the functional properties of proteins to

facilitate the computational prediction of their functions [Con04]. They are represented as a gold standard and the main source in protein functional annotations. There are three major types of gene ontologies that are specific to the functional domain and are divided into three different categories.

- **Biological process (BP)**: A biological process refers to a biological objective to which the gene or gene product contributes. A process is accomplished by one or more ordered assemblies of molecular functions. BPs often involve a chemical or physical transformation. For instance, an example of a higher level of BP can be *cell growth* and *maintenance* or *signal transduction*. Moreover, lower-level BP terms are *translation*, *pyrimidine metabolism*, or *cAMP biosynthesis* [Con04].

- **Molecular function (MF)**: A molecular function is defined as the biochemical activity of proteins, including specific binding to ligands or structures. MF describes only what is done without specifying where or when the event actually occurs. For example, broad MF terms are *enzyme*, *transporter*, or *ligand*. Moreover, lower level MF terms are *adenylate cyclase* or *Toll receptor ligand* [Con04].

- **Cellular component (CC)**: A cellular component refers to the place in the cell where a protein is active. The cellular component includes such terms as *ribosome* or *proteasome*, specifying where multiple gene products would be found. It also includes terms such as *nuclear membrane* or *Golgi apparatus* [Con04].

The functional annotations are downloaded from Gene Ontology [Con04] database. Each category of GO is represented in levels (i.e., levels I, II, and III). A lower level (i.e., level I) represents more specific terms, whereas a higher (i.e., level III) represents more general terms. Table 3.2 shows the number of terms per category.

Terms	Level I	Level II	Level III
Biological Process (BP)	855	535	244
Molecular Function (MF)	216	126	53
Cellular Component (CC)	181	113	54

Table 3.2: **Overview of the Gene Ontology (GO) terms used in the study**. Level I: 10 < proteins per term < 30; Level II: 30 < proteins per term < 100; and Level III: 100 < proteins per term < 300.

3.1.4 Bioinformatics resources

All the databases and computational tools used in this dissertation are shown in Table 3.3.

Database	Source	Citation
SGD	https://www.yeastgenome.org/	[Che+98]
NCBI-GEO	https://www.ncbi.nlm.nih.gov/geo/	[Bar+12]
JASPAR	https://jaspar.genereg.net/analysis	[For+20]
GO consortium	http://geneontology.org/	[Con04]
STRING	https://string-db.org/	[Szk+20]
RSAT	http://rsat.sb-roscoff.fr/	[Ngu+18]
YEASTRACT	http://www.yeastract.com/	[Mon+20]
YeastEnrichr	https://maayanlab.cloud/YeastEnrichr/	[Kul+16]

Table 3.3: **Sources of the databases utilized in this study.**

3.2 Proposed Models

Graph Representation Learning (GRL) algorithms allow us to encode graph structure into compact embedding vectors [HYL17b]. We formally define the task as a multilayer network embedding problem. Given a multilayer network, we aim to learn low-dimensional latent node representations (i.e., embeddings) so that the structure of the input network layers is properly integrated and preserved in the new space. We propose three models in this dissertation, viz. BraneExp, BraneNet and BraneMF. The models are presented in detail in the below sections. Besides, we define the objective function of proposed models in a way that is independent of downstream machine learning tasks, and the embeddings are learned in a purely unsupervised way. We employ these embeddings for different downstream prediction tasks dedicated to the functional analysis of proteins. All these tasks are mentioned in Section 3.6.

A multilayer graph of L-layers is a set of $G = \{G_l\}_{l=1}^{L} = \{(\mathcal{V}_l, \mathcal{E}_l)\}_{l=1}^{L}$ graphs, where $\mathcal{V}_l := \{v_1, \ldots, v_{|\mathcal{V}|_l}\}$ and $\mathcal{E}_l := \{e_{1_l}, \ldots, e_{M_l}\}$ are the vertex and the edge sets, respectively. $|\mathcal{V}|_l$ and M_l denote the number of nodes and edges for each layer. Throughout the dissertation, we assume that the layers share the same set of nodes, so $\mathcal{V}_l = \mathcal{V}_j = \mathcal{V}$ for every $1 \leq l < j \leq L$; edges are weighted and undirected. We use $\mathbf{A}^{(l)}$ to denote the adjacency matrix of the associated layer G_l. Our goal is to learn a low-dimensional feature representation for all $|\mathcal{V}|$ nodes. This integrated d-dimensional representation of G is given by $\mathbf{\Omega}_d \in \mathbb{R}^{|\mathcal{V}| \times d}$ ($d \ll |\mathcal{V}|$).

3.3 BraneExp: A Random Walk-based Network Integration Framework with Exponential Family Embeddings

Inspired by the aforementioned GRL methods and their limitations (Sections 2.4 and 2.5), in this work, we have considered expressive conditional probability models to relate nodes within random walk sequences, towards extracting informative latent node representations. We capitalize on exponential family distributions to capture interactions between nodes in random walks that traverse nodes within and across input network layers. More precisely, we introduce BraneExp, a network integration framework with the concept of exponential family graph embeddings [CM20a], that generalizes multilayer random walk-based GRL methods to an instance of exponential family conditional distribution.

We first describe how relevant node pairs are sampled with random walks—a key step towards multi-omics data integration. Then, we explain the methodology employed to learn node representations by modeling the underlying interactions among nodes with exponential family distributions. A general overview of the proposed methodology is depicted in Figure 3.2.

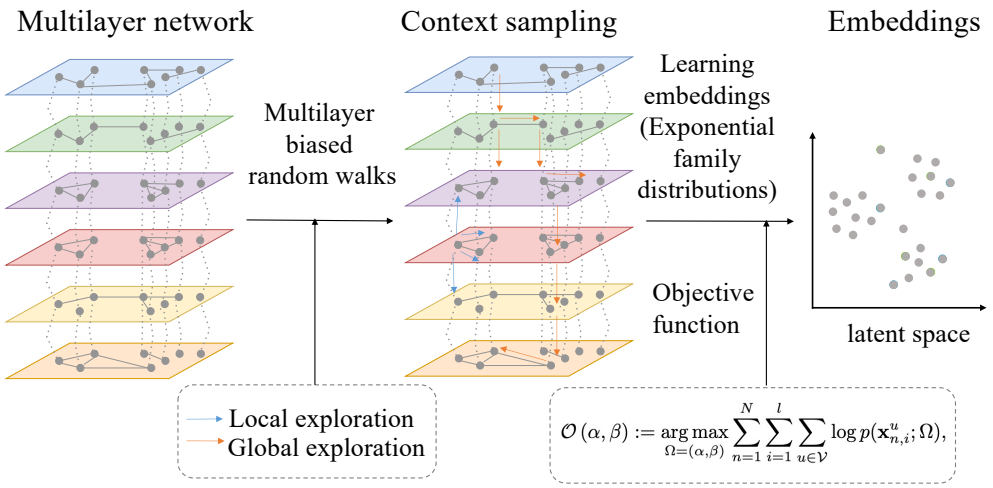


Figure 3.2: **Illustration of the BraneExp model.** Firstly, given a multilayer network, the model performs context sampling that is adapted to explore local and global structures of the network. Secondly, embeddings are learned by optimizing the objective function generalizing multilayer random walk-based GRL methods to an instance of exponential family conditional distribution.

3.3.1 Context sampling using multilayer biased random walks

The simulation of random walks plays a significant role in our approach since we use them to model the interactions among nodes and also to facilitate data integration. Due to the multilayer network structure, each layer can possess a completely different inner structure, and each node in the graph layers might play a different role. Therefore, our task becomes more challenging than the classical network representation learning problems [Che+20] as we target to circumvent it by simulating random walks. From a multilayer network and simulated random walks, our goal is to learn node representations in a lower-dimensional space so that their embeddings reflect the underlying patterns commonly shared by these network layers. To do this, we leverage random walks to sample nodes sharing similar characteristics across different omics layers. Although random walks have been utilized before in representation learning [Che+20], here we introduce a flexible approach for multilayer network structures.

To extract the node's context in a multilayer graph, we propose a random walk-based sampling procedure that can explore local and global structures in the graph. Figure 3.3 illustrates local and global exploration in a graph. Local exploration helps in discovering the clustering structure around the node of interest, whereas global exploration contributes to capturing global associations within nodes in the graph [NM18]. More precisely, local exploration is an algorithm that efficiently visits and marks all the key nodes in a graph in an accurate breadth-wise fashion. This algorithm selects a single node (initial or source point) in a graph and then visits all the nodes adjacent to the selected node. Once the algorithm visits and marks the starting node, then it moves toward the nearest unvisited nodes and analyses them. Once visited, all nodes are marked. These iterations continue until all the nodes of the graph have been successfully visited and marked. And global exploration is an algorithm for finding or traversing graphs in a depth-ward direction. The execution of the algorithm begins at the root node and explores each branch before backtracking. It uses a stack data structure to remember, get the subsequent node, and start a search, whenever a dead-end appears in any iteration. To capture such local and global associations, we propose a *biased* version of random walks adapted to multilayer graphs [NM18]. It combines both types of exploration (i.e., local and global) with a decay parameter α to control the importance of nodes with respect to their distance from the node of interest.

More formally, for each node $v_i \in \mathcal{V}$, a proximity score τ_{v_i} is computed to estimate how far the candidate node v_i is from the source node. When the i-th node in the walk is discovered, the proximity score of every node adjacent to that is increased by α^{i-1} and $\bar{\alpha}^{i-1}$, for nodes in the same and different layer respectively, where $\alpha, \bar{\alpha} \in [0, 1]$. Then, the probability distribution of selecting the next node for the current walk is computed based on the proximity scores of the neighborhood nodes of the most recently visited node. For local explorations, the probability of a node being the next one in the random walk sequence should be proportional to its proximity score, i.e.,

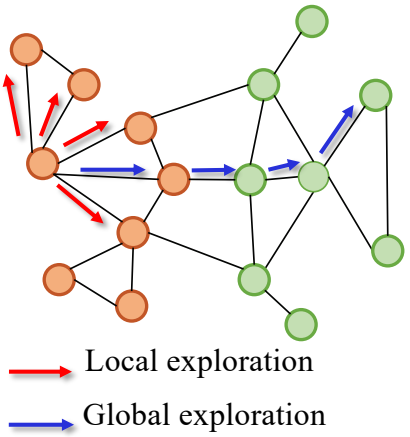


Figure 3.3: **Illustration of graph exploration strategies.** Above figure shows the illustration of local and global exploration strategies. Firstly local exploration, also called Breadth First Search (BFS), explores neighboring nodes by following the "go wide, bird's eye-view" philosophy. Secondly, in global exploration, also called Depth First Search (DFS), from an initial node, the algorithm searches for nodes going down one path until it reaches the end. It follows the "go deep, head first" philosophy.

$$p_{v_i} = \frac{\tau_{v_i}}{\sum_{w \in \mathcal{V}_{(u)}} \tau_w}. \tag{3.1}$$

In the case of global exploration, the probability is set to be inversely proportional to that score, i.e.,

$$p_{v_i} = \frac{1/\tau_{v_i}}{\sum_{w \in \mathcal{V}_{(u)}} 1/\tau_w}, \tag{3.2}$$

where u is the most recently visited node, and $\mathcal{V}(u)$ defines the set of neighbors of u. Thus, given the desired exploration strategy, the context set for each node $\mathbf{w}_{(n,i)} \in \mathcal{V}$ is given by

$$\mathcal{C}_t(\mathbf{w}_{(n,i)}) := \{\mathbf{w}_{(n,j)} \in \mathcal{V} : -\max\{1, i-t\} \leq j \neq i \leq \min\{i+t, l\}\}, \tag{3.3}$$

where $\mathbf{w}_{(n,j)}$ indicates the node appearing at the j-th position of the n-th random walk, and t is the window size. We call each element of $\mathcal{C}_t(\mathbf{w}_{(n,i)})$ as the *context* of a *center* node $\mathbf{w}_{(n,i)}$.

3.3.2 Learning embeddings with exponential family distributions

Random walk-based methods generate node sequences and learn node representations by maximizing the co-occurrence probability of nodes within a certain

distance [PAS14; GL16; NM18]. Similarly, we define our objective function as follows:

$$\mathcal{O}\Big(\alpha,\beta\Big) := \underset{\Omega=(\alpha,\beta)}{\arg\max} \sum_{n=1}^{N} \sum_{i=1}^{l} \sum_{u\in\mathcal{V}} \log p(\mathbf{x}_{n,i}^{u};\boldsymbol{\Omega}), \tag{3.4}$$

where $\mathbf{x}_{n,i}^{u}$ is the observed variable indicating the relationship between the pair of nodes $(\mathbf{w}_{(n,i)}, u) \in \mathcal{V}^2$, and $\boldsymbol{\Omega} = (\alpha, \beta)$ are the parameters of the model, which correspond to the node embedding vectors. Note that we obtain two different representations for each node. Here, $\alpha[v]$ indicates the embedding of node v if it is considered as *context*, and $\beta[v]$ denotes its representation if it is interpreted as *center* node. Although the conventional choice for modeling node relationships is the *softmax* function [PAS14], it limits capturing possible intricate patterns in node interactions across the layers of the network structure. Therefore, here we extend it with a general framework based on *exponential families*, which is a set of parametric probability distributions satisfying the following form:

$$p(\mathbf{x}) = h(\mathbf{x}) \exp\Big(\eta\, T(\mathbf{x}) - A(\eta)\Big), \tag{3.5}$$

where $h(\mathbf{x})$ is the base measure, $T(\mathbf{x})$ is the sufficient statistic, and $A(\eta)$ is the log-normalizer function. Note that many widely utilized distributions, such as the ones of *Bernoulli*, *Dirichlet*, and *Normal*, are actually exponential families. The main benefit of this generic formulation is that it provides an elegant and flexible way to model the complex interactions between center and context nodes in random walk sequences [CM20a; Rud+16]. By plugging the exponential form into the objective function provided in Equation (3.4), we obtain the following:

$$\underset{\Omega=(\alpha,\beta)}{\arg\max} \sum_{n=1}^{N} \sum_{i=1}^{l} \sum_{u\in\mathcal{V}} \log h\Big(\mathbf{x}_{n,i}^{u}\Big) + \eta_{\mathbf{w}_{(n,i)}}^{u}\, T\Big(\mathbf{x}_{n,i}^{u}\Big) - A\Big(\eta_{\mathbf{w}_{(n,i)}}^{u}\Big). \tag{3.6}$$

Here, we define the natural parameter $\eta_{\mathbf{w}_{(n,i)}}^{u}$ as the product of embeddings, $\alpha[u]^{\top} \cdot \beta[\mathbf{w}_{(n,i)}]$.

In our approach, we employ the Bernoulli distribution to model node co-occurrences by setting $h(\mathbf{x}) = 1$, $T(\mathbf{x}) = x$ and $A(\eta) = \log(1 + e^{\eta})$. Let $x_{n,i}^{u}$ be a Bernoulli random variable indicating the occurrence of u in the context of node $\mathbf{w}_{(n,i)}$. Note that this is equal to 1 if node u appears at any position index $i + j$, for $-t \leq j \neq 0 \leq t$. Then, we can rewrite our objective function as follows:

$$\underset{\Omega=(\alpha,\beta)}{\arg\max} \sum_{n=1}^{N} \sum_{i=1}^{l} \sum_{|j|} \Bigg(\underbrace{\log p\Big(y_{n,i+j}^{\mathbf{w}_{(n,i+j)}}\Big)}_{\text{positive instances}} + \sum_{u\in\mathcal{V}\setminus\{\mathbf{w}_{(n,l+j)}\}} \underbrace{\log p\Big(y_{n,i+j}^{u}\Big)}_{\text{negative instances}} \Bigg), \tag{3.7}$$

Algorithm 1 BraneExp

Input: Multilayer graph $G = (\mathcal{V}, \mathcal{E})_{l=1}^{L}$
Number of walks n
Walk length w
Window size t
Embedding dimension d
Output: d-dimension protein features, $\mathbf{\Omega}_d$

1. Perform n random walks of length w for node $v \in \mathcal{V}_l$ within layer l and across layers from l to L.

2. Learn d-dimensional node representations by optimizing the objective function in Equation 3.4.

return $\mathbf{\Omega}_d$

where $y^{u}_{n,i+j}$ indicates the occurrence of node u at the $(i+j)$-th position of the n-th walk. However, the optimization step is very costly due to the size of the negative instances. Therefore, we approximate it by leveraging the *negative sampling* approach [Mik+13]:

$$\arg\max_{\Omega=(\alpha,\beta)} \sum_{n=1}^{N} \sum_{i=1}^{l} \sum_{|j|} \left(\log p\left(y_{n,i+j}^{\mathbf{w}_{(n,i+j)}}\right) + k \mathop{\mathbb{E}}_{u\sim p^{-}} \left[\log p\left(y_{n,i+j}^{u}\right) \right] \right), \tag{3.8}$$

where k indicates the number of negative instances sampled from the noise distribution p^{-}. We employ the strategy described in [Mik+13], and negative instances are sampled from the whole vertex set with respect to their number of occurrences in the generated walks raised to the power of 0.75. In the experimental evaluation, we generate $k = 5$ negative samples for each positive instance. The BraneExp algorithm is given in Alg. 1.

BraneExp has the following conceptual advantages:

- It preserves both the intra-layer and inter-layer interactions, thereby learning rich features;

- It is a scalable method as it uses the optimization procedure, which leverages negative sampling.

More recently, a theoretical connection between *Skip-Gram*-based network embedding algorithms and matrix factorization is shown in [Qiu+18]. It has presented an approximation algorithm for computing network embedding for single-layer networks. Next, in BraneNet, we extend this idea towards multilayer networks by factorizing its supra-adjacency random walk matrix.

3.4 BraneNet: Multilayer Network Embedding as Matrix Factorization

We propose BraneNet, a novel multi-omics integration framework for multilayer networks. BraneNet leverages random walk information within a matrix factorization framework. Our goal is to efficiently integrate multi-omics data to study different regulatory aspects of multilayered processes that occur in organisms. Our method also considers inter- and intra- omics relationships that could be supported with *a priori* knowledge. In Figure 3.4, an illustration of BraneNet is presented. The model takes as input a multilayer network. It first builds a supra-adjacency of size $|\mathcal{V}| \times |\mathcal{V}|$, where N is a set of nodes shared by all layers. matrix. Then, we compute a random walk-based PPMI matrix $\mathbf{M}$ for $\bar{\mathbf{A}}$ via a closed-form solution. The matrix $\mathbf{M}_{|\mathcal{V}|\times|\mathcal{V}|}$ is then factorized using Singular Value Decomposition (SVD) and the d-dimensional embedding vectors $\mathbf{\Omega}_d \in \mathbb{R}^{|\mathcal{V}|\times d}$ ($d \ll |\mathcal{V}|$) are given by $\mathbf{U}_d\sqrt{\Sigma_d}$.

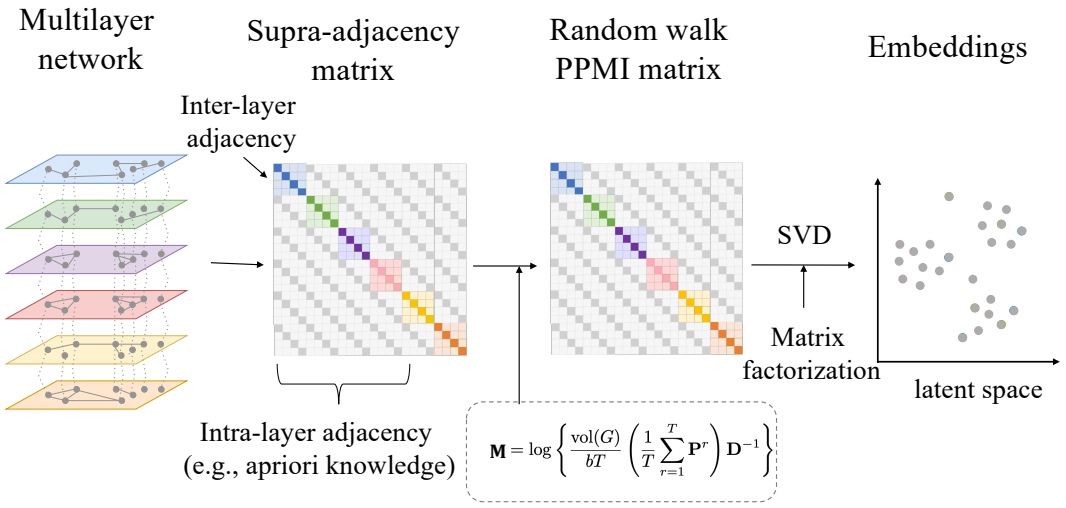


Figure 3.4: **Schematic representation of BraneNet.** A multilayer network $\bar{\mathbf{A}}$ is composed of intra- and inter-omics relationships. For $\bar{\mathbf{A}}$, the random walk-based PPMI matrix $\mathbf{M}$ is computed. To obtain embeddings, $\mathbf{M}$ is factorized, and the final embeddings $\mathbf{\Omega}_d$ are obtained.

3.4.1 Construction of a supra-adjacency matrix

X is a set of x intra-omics networks represented as $\mathbf{X}^{(1)}_{n_1\times n_1}, \mathbf{X}^{(2)}_{n_2\times n_2}, \ldots, \mathbf{X}^{(x)}_{n_x\times n_x}$, while Y is a set of $y = \frac{x(x-1)}{2}$ inter-omics networks represented as $\mathbf{Y}^{(1)}_{n_1\times n_2}, \mathbf{Y}^{(2)}_{n_1\times n_3}, \ldots, \mathbf{Y}^{(y)}_{n_{x-1}\times n_x}$. A multilayer network G of $|\mathcal{V}|$ nodes (biomolecules) and $|E|$ edges (interactions) is built using sets $\mathcal{X}$ and $\mathcal{Y}$. The network is represented

by its supra-adjacency matrix $\bar{\mathbf{A}}_{|\mathcal{V}|\times|\mathcal{V}|}$ that is defined as:

$$\bar{\mathbf{A}} = \bigoplus_{x} \mathbf{A}^{(x)} + \mathbf{C}, \tag{3.9}$$

where $\bigoplus_x \mathbf{A}^{(x)}$ is the intra-omics adjacency matrices and $\mathbf{C}$ is a block matrix with zero diagonal blocks that stores inter-omics connections obtained from elements in Y. The final output of $\bar{\mathbf{A}}$ has intra-omics networks represented as blocks in the main diagonal and inter-omics networks represented as off-diagonal matrices.

3.4.2 Representation learning

To embed nodes from different omics modalities into a common latent space towards integrating inter- and intra-omics relationships, we construct a random walk matrix $\mathbf{M}$ for the multilayer graph G. $\mathbf{M}$ is defined by the random walk transition probabilities to traverse nodes within and across layers. The flexibility of random walks to traverse within and across layers allows us to capture inter- and intra-layer node neighbourhood information. This is an important and useful property to consider while performing data integration from multilayer networks [Jag+21a; BK18]. For instance, starting from node v in G, a random walk traverses the multilayer graph, moving across neighborhood nodes of v chosen uniformly at random. This process repeats for a predefined number of walks per node.

Nevertheless, for large networks, simulating random walks is computationally expensive, and therefore it is not a recommended approach. To address this limitation, we leverage the relationship between random walk-based GRL algorithms that rely on the *Skip-Gram* model (for instance, emphDeepWalk [PAS14]) and matrix factorization [Qiu+18]. Focusing on a specific instance of such approaches, the *DeepWalk* method first generates a corpus $\mathcal{W}$ by performing random walks on a graph [PAS14]. A corpus $\mathcal{W}$ is a bag of multisets that counts the multiplicity of nodes v and their context c. *DeepWalk* then trains a *Skip-Gram* model on $\mathcal{W}$. To be formal, it assumes a corpus of node sequences represented as $v_1, v_2, \ldots, v_w$, where w is the length of the random walk. The context of node v_i is given as the surrounding nodes in a $2t$-sized window $\{v_{i-t}, \ldots, v_{i-1}, v_{i+1}, \ldots, v_{i+t}\}$, $t > l$. Following [LG14] and [Qiu+18], the closed form expression of the *DeepWalk* matrix for any graph G is given by:

$$\underbrace{\log\left(\frac{\#(v,\mathcal{C})|\mathcal{W}|}{\#(v).\#(\mathcal{C})}\right) - \log b}_{\text{Skip-Gram}} = \underbrace{\log\left(\frac{\text{vol}(G)}{bt}\left[\frac{1}{t}\sum_{r=1}^{t}\mathbf{P}^r\right]\mathbf{D}^{-1}\right)}_{\textit{DeepWalk} \text{ matrix}}. \tag{3.10}$$

On the left-hand side, $\#(v,\mathcal{C}), \#(v)$, and $\#(\mathcal{C})$ denote the number of times node-context pair $(v,\mathcal{C})$, node v and context $\mathcal{C}$ appear in $\mathcal{W}$, while b is the number of negative samples. The right-hand side is represented by $\mathbf{D}$ as the degree matrix of graph G, and the power matrix $\mathbf{P}$ defined as $\mathbf{D}^{-1}\mathbf{A}$. Here, vol$(G)$ is the volume (size)

Algorithm 2 BraneNet

Input: Multilayer graph $G = (\mathcal{V}, \mathcal{E})_{l=1}^{L}$
Parameters: window size: T and embedding dimension: d
Output: d-dimension protein features, $\mathbf{\Omega}_d$

1. Build a supra-adjacency matrix $\bar{\mathbf{A}}$ using each graph layer G_l (Equation 3.9).
2. Obtain its degree matrix $\mathbf{D}$ and adjacency matrix $\mathbf{A}$
3. Compute power matrix $\mathbf{P}$ where $\mathbf{P} = \mathbf{D}^{-1}\mathbf{A}$.
4. Compute PPMI matrix $\mathbf{M}$ for $\bar{\mathbf{A}}$ as given in Equation 3.11.
5. Compute protein features $\mathbf{\Omega}_d = \mathbf{U}_d\sqrt{\bar{\Sigma}_d}$.

return $\mathbf{\Omega}_d$

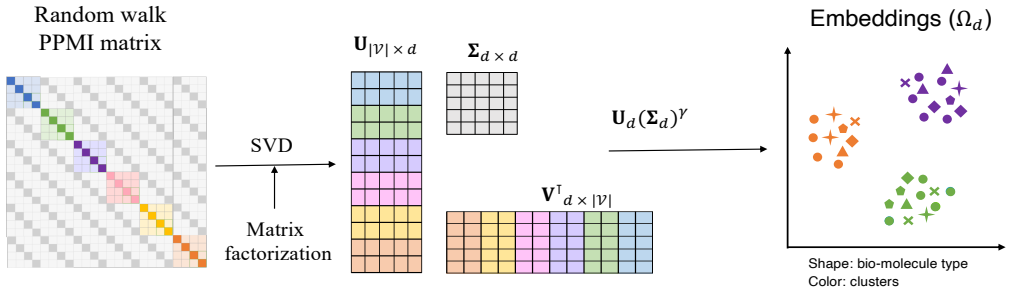
of G. In particular, a multilayer random walk matrix $\mathbf{M}$ is defined by computing the closed form of a properly normalized PPMI-based random walk transition matrix. This PPMI matrix is the well-studied pointwise mutual information (PMI) matrix that represents node similarities shifted by a global constant. It has been shown that normalized PPMI is better at optimizing *Skip-Gram's* objective and shows better performance than emphword2vec derived models [PAS14; GL16] in Natural Language Processing (NLP tasks) [LG14; Qiu+18]. For any graph G, $\mathbf{M}$ is given by:

$$\mathbf{M} = \log\left\{\frac{\mathrm{vol}(G)}{bt}\left(\frac{1}{t}\sum_{r=1}^{t}\mathbf{P}^r\right)\mathbf{D}^{-1}\right\}, \tag{3.11}$$

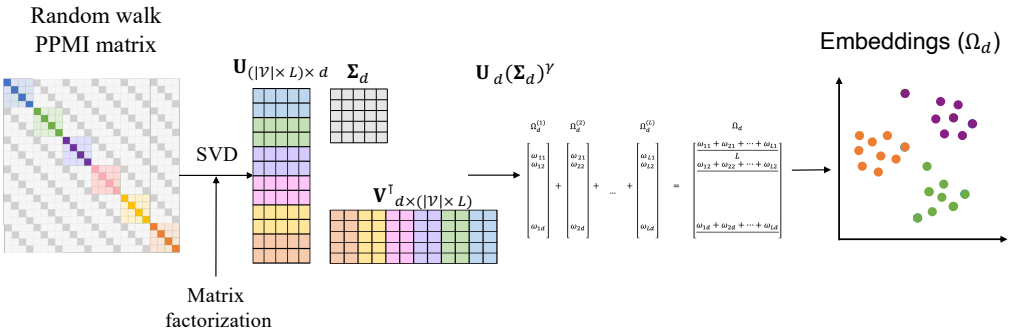
where $\mathbf{P} = \mathbf{D}^{-1}\bar{\mathbf{A}}$. Matrices $\bar{\mathbf{A}}$ and $\mathbf{D}$ are, respectively, the adjacency and diagonal degree matrices of the graph G and $\mathrm{vol}(G)$ is the sum of the node degrees of G. t corresponds to the window size and b is number of negative samples [Qiu+18]. In order to obtain node embeddings from matrix $\mathbf{M}$, we perform spectral decomposition using SVD [Bis20], given by $\mathbf{M} = \mathbf{U}\Sigma\mathbf{V}^\top$. Since $\mathbf{M}$ is a real and symmetric matrix, $\mathbf{U}$ and $\mathbf{V}$ correspond to singular vector matrices, and Σ is the singular value matrix. The integrated embedding matrix $\mathbf{\Omega}_d$ of dimension $|\mathcal{V}| \times d$ is given by the first d eigenvectors of $\mathbf{M}$, appropriately weighted by the square root of Σ_d as, $\mathbf{\Omega}_d = \mathbf{U}_d\sqrt{\Sigma_d}$. The BraneNet algorithm is given in Alg. 2.

Here, we integrate multilayer heterogeneous biological networks using *apriori* knowledge. In the case of commonly studied multilayer homogeneous networks, BraneNet learns embeddings for nodes in each layer. However, to obtain a joint embedding vector across layers could be possible by performing operations on the learned embeddings for each layer. For instance, taking average or finding mean

embedding by applying optimization techniques. To clarify, the illustration of learning BraneNet embeddings for both homogeneous and heterogeneous multilayer networks is shown in Figure 3.5b and Figure 3.5a, respectively. Nevertheless, this post-processing of embeddings is a shallow approach with the possibility of losing important features. The ideal instance would be to learn embeddings jointly across multiple layers.



(a) Embedding a multilayer heterogeneous network



(b) Embedding a multilayer homogeneous network

Figure 3.5: **Embedding multilayer heterogeneous and homogeneous networks using BraneNet.** (a) Heterogeneous networks. The PPMI matrix $\mathbf{M}$ is decomposed into $\mathbf{U}, \Sigma$ and $\mathbf{V}$ matrices using Singular Value Decomposition (SVD). $\mathbf{U}$ and $\mathbf{V}$ are the singular vector matrices and Σ is the singular value matrix. The embeddings are computed by taking the product of d columns of $\mathbf{U}$ and the top d values of the diagonal matrix, where d is the size of the embedding. We use weighted Σ values whose weights are given by parameter γ. (b) Homogeneous networks. The PPMI matrix $\mathbf{M}$ is factorized using SVD, where the singular vector matrices $\mathbf{U}$ and $\mathbf{V}$ are obtained of size $|\mathcal{V}| \times L$. That is, the embeddings are learned for each node in each layer. To obtain a joint embedding, post-operations on the embeddings learned per layer are performed (e.g., taking a point-wise average).

3.5 BraneMF: Multilayer Network Embedding by Jointly Decomposing Random Walk Matrices

With the concept of joint matrix factorization that generalizes random walk-based network embedding models, we introduce BraneMF. BraneMF is an integration

framework to learn protein features from multiple PPI networks. A schematic representation is given in Fig. 3.6. Firstly, we compute a random walk-based multilayered PPMI matrix that captures node proximity. Secondly, we learn protein features by jointly factorizing the layers of this matrix using SVD. Lastly, we utilize the learned protein features for various prediction tasks. More precisely, BraneMF brings the best of two worlds: expressiveness of well-celebrated random walk-based embedding models (e.g., DeepWalk, node2vec) and the solid formulation of matrix factorization—going further by extending them to integrate multiple sources.

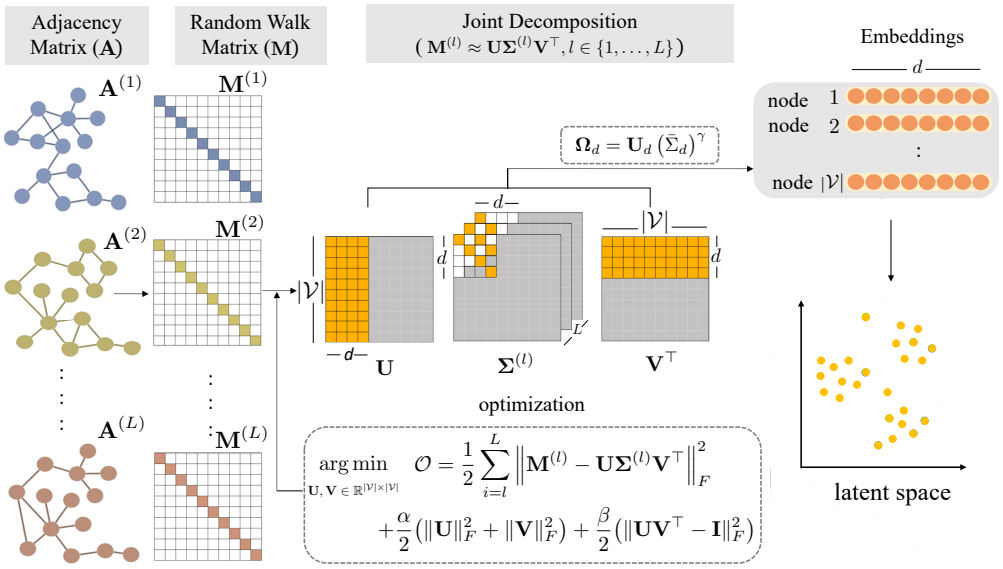


Figure 3.6: **Schematic representation of BraneMF.** The framework takes as input a set of PPI networks represented by their adjacency matrices $\mathbf{A}^{(l)}, l \in \{1, 2, \ldots, L\}$. For each PPI network, the random walk matrix $\mathbf{M}^{(l)}$ is computed. For integrative analysis, we learn protein features by jointly decomposing these random walk matrices $\mathbf{M}^{(l)}$ into $\mathbf{U}\mathbf{\Sigma}^{(l)}\mathbf{V}^\top$. The protein features $\mathbf{\Omega}_d$ are given by $\mathbf{U}_d(\bar{\mathbf{\Sigma}}_d)^\gamma$, where d is the embedding dimension, and t is a factor that scales the magnitude of the singular values.

3.5.1 Computation of random walk-based PPMI matrices

Network properties, particularly topological ones, can unravel important information about the graph structure. While handling multiple heterogeneous networks that correspond to diverse characteristics, it is essential to extract relevant information concealed in their topology. We aim to extract such information from a multilayer graph G, constructing a set of PPMI matrices that can delineate node similarity via random walks. Random walks, defined as node paths that consist of a series of random steps on the graph, have been utilized as a similarity measure for a variety of problems in graph theory.

The PPMI matrix $\mathbf{M}^{(l)}$ for graph layer l, can be computed using the closed form of the *DeepWalk* matrix $\mathbf{M}^{(l)}$ as shown in Eq. (3.10) and Eq. (3.11). The set of PPMI matrices $\mathbf{M} = \{\mathbf{M}^{(l)}\}_{l=1}^{L}$ for a multilayer graph G is given by:

$$\mathbf{M} = \left\{ \log\left(\frac{\mathrm{vol}(G_l)}{bt} \left[\frac{1}{t} \sum_{r=1}^{t} \left(\mathbf{P}^{(l)}\right)^r \right] \left(\mathbf{D}^{(l)}\right)^{-1} \right) \right\}_{l=1}^{L}. \tag{3.12}$$

Each matrix $\mathbf{M}^{(l)}$ corresponds to the DeepWalk matrix of $\mathcal{G}_l$ when the length of random walks goes to infinity. In this regard, $\mathbf{M}^{(l)}$ is different from the PPMI matrices computed in previous approaches. As discussed in Sec. 2.5, the PPMI matrix for deepNF and Mashup is computed using Random Walks with Restart (RWR), considering an additional parameter that controls the restart probability of the random walk. Despite both capturing node proximity, the DeepWalk matrix significantly differs from RWR; the formulation ensures that its latent factors will derive embeddings that capture node co-occurrences in random walks.

3.5.2 Joint representation learning for multilayer networks

The set of matrices $\mathbf{M}$ computed as above captures node proximity that still represents high-dimension protein features. As a consequence of the curse of dimensionality, these features are not compatible with downstream prediction tasks. Therefore, we want to obtain low-dimension integrated protein features that could be easily fed to any downstream machine learning tasks of interest. Nevertheless, our integration framework is developed on the construction of random walk-based PPMI matrices, Equation (3.12), on which joint matrix factorization is eventually performed. In order to learn the spectrum of one layer in graph G, the singular values and singular vectors of its PPMI matrix $\bar{\mathbf{M}}$ can be obtained using SVD, as $\bar{\mathbf{M}} = \mathbf{U}\Sigma\mathbf{V}^\top$, where $\mathbf{U}$ and $\mathbf{V}$ correspond to the left and right singular vector matrices, and Σ is the diagonal singular value matrix. In the case of a multilayer graph G composed of L layers, we have L symmetric PPMI matrices. As a natural extension, we propose to approximate each PPMI matrix $\mathbf{M}^{(l)}$ by a set of jointly decomposed singular vector and singular value matrices shared by all layers, given by: $\mathbf{M}^{(l)} \approx \mathbf{U}\Sigma^{(l)}\mathbf{V}^\top$, $l \in \{1,\ldots,L\}$. The same correspondence keeps, where $\mathbf{U}$ and $\mathbf{V}^\top$ are orthogonal matrices containing the joint singular vectors and $\Sigma^{(l)} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$ contains the corresponding singular values per layer. The minimization of the following objective function $\mathcal{O}$ yields $\mathbf{U}$ and $\mathbf{V}$:

$$\underset{\mathbf{U},\mathbf{V}\in\mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}}{\arg\min}\ \mathcal{O} = \frac{1}{2}\sum_{l=1}^{L} \|\mathbf{M}^{(l)} - \mathbf{U}\Sigma^{(l)}\mathbf{V}^\top\|_F^2 + \frac{\alpha}{2}(\|\mathbf{U}\|_F^2 + \|\mathbf{V}\|_F^2) + \frac{\beta}{2}\|\mathbf{U}\mathbf{V}^\top - \mathbf{I}\|_F^2, \tag{3.13}$$

$\mathbf{U}$ and $\mathbf{V}^\top$ represent the joint orthonormal matrices, $\mathbf{I}$ is the $|\mathcal{V}|\times|\mathcal{V}|$ identity matrix, and $\|\cdot\|_F$ denotes the Frobenius norm. The first term of the objective function $\mathcal{O}$ measures the overall approximation error when all layers are decomposed over $\mathbf{U}$. The second term, the norms of $\mathbf{U}$ and $\mathbf{V}^\top$, is added to improve numerical stability for the solutions; and the last term is a constraint to ensure that $\mathbf{V}^\top$ is close to the

inverse of $\mathbf{U}$ ($\mathbf{M}^{(l)}$ is a symmetric matrix, thus its SVD can be given by $\mathbf{U}\Sigma\mathbf{U}^{-1}$).

We solve the problem in Equation (3.13) to get $\mathbf{U}$ and $\mathbf{V}^\top$. Since Equation (3.13) is not jointly convex on $\mathbf{U}$ and $\mathbf{V}^\top$, we adopt an alternating scheme to find a local minimum for $\mathcal{O}$ by fixing $\mathbf{V}^\top$ first and optimizing on $\mathbf{U}$, and vice versa [Don+12]. As a consequence, a good initialization is important. In practice, we suggest to compute the SVD of the mean for all matrices $\mathbf{M}^{(l)}$, and initialize $\mathbf{U}$, Σ, and $\mathbf{V}^\top$ with the resulting matrices. The stopping condition is defined by the convergence behavior of the cost function—the difference between its values for two consecutive iterations. Notice that, the objective function $\mathcal{O}$ is differentiable with respect to matrices $\mathbf{U}$ and $\mathbf{V}^\top$, whose derivation is given as:

$$\begin{aligned}
\frac{\partial \mathcal{O}}{\partial \mathbf{U}} &= -\left(\sum_{l=1}^{L}(\mathbf{M}^{(l)} - \mathbf{U}\Sigma^{(l)}\mathbf{V}^\top)\right)\mathbf{V}\Sigma^{(l)} + \alpha\mathbf{U} + \beta(\mathbf{U}\mathbf{V}^\top - \mathbf{I})\mathbf{V}^\top, \\
\frac{\partial \mathcal{O}}{\partial \mathbf{V}^\top} &= \Sigma^{(l)}\mathbf{U}^\top\left(\sum_{i=1}^{L}(\mathbf{M}^{(l)} - \mathbf{U}\Sigma^{(l)}\mathbf{V}^\top)\right) + \alpha\mathbf{V}^\top + \beta\mathbf{U}^\top(\mathbf{U}\mathbf{V}^\top - \mathbf{I}).
\end{aligned} \tag{3.14}$$

We minimize $\mathcal{O}$ over $\mathbf{U}$ and $\mathbf{V}^\top$. $\mathbf{U}$ is the set of joint singular vectors, namely a joint spectrum shared by all layers in G; $\bar{\Sigma}$ is the joint singular value matrix computed by taking the average of $\Sigma^{(l)}$ matrices. The integrated embeddings $\mathbf{\Omega}_d$ are obtained by multiplying the first d columns of $\mathbf{U}$ scaled by the γ-th power of the singular value magnitudes:

$$\mathbf{\Omega}_d = \mathbf{U}_d(\bar{\Sigma}_d)^\gamma. \tag{3.15}$$

This optimization process is similar to [Don+12], that uses an eigendecomposition to find low-rank eigenvector matrices that are shared by all graph layers. However, these matrices were not random walk-based and the joint decomposition is performed differently. The joint SVD process described above is essentially based on integrating information from multiple graph layers. It tends to treat each graph equitably, building a solution that smoothens out the specificities of each layer. The BraneMF algorithm is given in Alg. 3.

3.6 Downstream Tasks

We demonstrate the applicability of learned features from the proposed models for essential multi-omics inference tasks. The overview of downstream tasks is shown in Figure 3.7. Each downstream task is detailed below.

3.6.1 Gene Regulatory network inference

Gene regulatory networks (GRN) impart how signals propagate through biomolecules and result in transcriptomic modifications. These regulatory networks are computational modules of a biological system that carry out decision-making processes. They enable us to determine the ultimate response

Algorithm 3 BraneMF

Input: Multilayer graph G_l, $l = 1,\ldots,L$;
Parameters: window size: T, embedding dimension: d, and weighting factor: γ
Output: d-dimension protein features, $\mathbf{\Omega}_d$

1. For each $\mathcal{G}_l$, obtain its degree matrix $\mathbf{D}^{(l)}$ and adjacency matrix $\mathbf{A}^{(l)}$
2. Compute power matrix $\mathbf{P}^{(1)},\ldots,\mathbf{P}^{(T)}$ for each l in $\mathcal{G}_l$ (where $\mathbf{P} = \mathbf{D}^{-1}\mathbf{A}$)
3. Compute PPMI matrix $\mathbf{M}^{(l)}$ for $\mathcal{G}$ as given in Equation 3.12
4. Solve the optimization problem in Equation 3.13 to obtain $\mathbf{U}$ and $\bar{\mathbf{\Sigma}}$
5. Compute protein features $\mathbf{\Omega}_d = \mathbf{U}_d(\bar{\mathbf{\Sigma}}_d)^{\gamma}$

return $\mathbf{\Omega}_d$

of an organism to a stimulus. Although there has been an intense research effort on GRN inference using gene expression for more than a decade, and much progress has been made, it remains a challenging problem [Pir+15a; Pir+17].

Even the most sophisticated inference techniques are far from perfect. Mainly by leveraging gene co-expression networks with the relationships between regulators, such as transcription factors (TFs) and the target genes they control, one can achieve a better understanding of regulatory interactions, providing us the access points to modulate such responses [Kso+21; Van+18b]. However, it is a challenging task to effectively combine this information in such a way that the rich and relevant features from the input datasets are preserved. Indeed, recent breakthroughs in graph representation learning have inspired us to solve the GRN inference task by modeling heterogeneous datasets as multilayer graphs and encoding latent representations for them. Here, we propose to integrate co-expression and TF-target networks.

First, we build a gene co-expression network from 61 microarray experiments deposited in NCBI-GEO database [Bar+12]. To define co-expression, the Pearson correlation for each gene pair was computed. Two genes are connected by an edge if the correlation between them is greater than 0.9 [Du+19]. Then, we infer TF-target relationships by using TFBS deposited in JASPAR database [For+20] and promoter sequences of genes [Eng+14]. We scan the TFBS in the promoters using the *matrix-scan* tool in RSAT [Ngu+18], and we infer the edges based on the presence of binding sites in the promoter of the gene. For the same set of nodes, a multilayer network is constructed from the co-expression and TF-target networks. We learn node embeddings using the proposed models described in Section 3.2. To infer regulatory interactions from the learned embeddings, we define the similarity between the embedding vectors by computing the cosine similarity

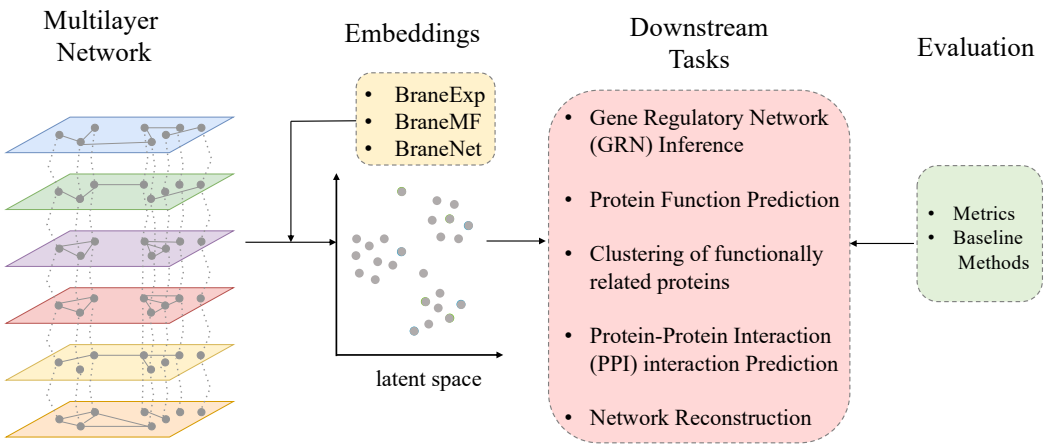


Figure 3.7: **Illustration of the downstream tasks.** After learning embeddings with the proposed models, we evaluate their performance using above mentioned downstream tasks and compare them with the baseline methods.

for each TF-gene interaction. Let u and v be the TF and gene, respectively. The embedding vectors for two nodes u and v is given by $\mathbf{\Omega}[u]$ and $\mathbf{\Omega}[v]$. Their cosine similarity can be derived by using the Euclidean dot product formula:

$$\mathbf{\Omega}[u] \cdot \mathbf{\Omega}[v] = \| \mathbf{\Omega}[u] \| \ \| \mathbf{\Omega}[v] \| \cos\theta. \tag{3.16}$$

Given two vectors of attributes, $\mathbf{\Omega}[u]$ and $\mathbf{\Omega}[v]$, the cosine similarity, $\cos(\theta)$, is represented using a dot product and magnitude as:

$$S_C(\mathbf{\Omega}[u], \mathbf{\Omega}[v]) := \cos(\theta) = \frac{\mathbf{\Omega}[u] \cdot \mathbf{\Omega}[v]}{\|\mathbf{\Omega}[u]\| \, \|\mathbf{\Omega}[v]\|} = \frac{\sum_{i=1}^{n} \mathbf{\Omega}[u]_i \mathbf{\Omega}[v]_i}{\sqrt{\sum_{i=1}^{n} \mathbf{\Omega}[u]_i^2} \sqrt{\sum_{i=1}^{n} \mathbf{\Omega}[v]_i^2}}, \tag{3.17}$$

where $\mathbf{\Omega}[u]_i$ and $\mathbf{\Omega}[v]_i$ are components of vector $\mathbf{\Omega}[u]$ and $\mathbf{\Omega}[v]$ respectively. The resulting similarity ranges from 1, meaning exactly opposite, to 1, meaning exactly the same, with 0 indicating orthogonality or decorrelation, while in-between values indicate intermediate similarity or dissimilarity. The illustration of the GRN inference is shown in Figure 3.8.

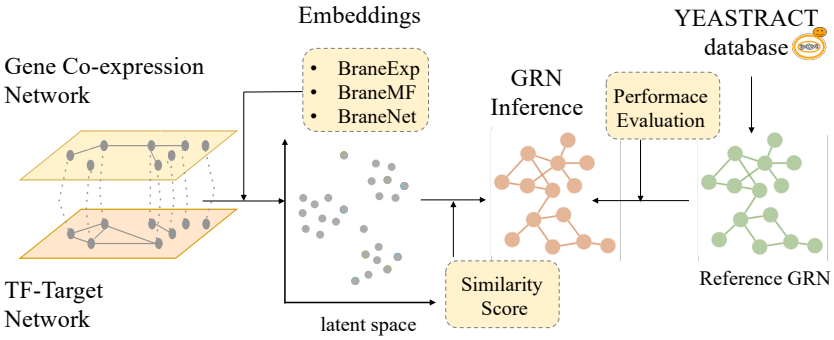


Figure 3.8: **Illustration of the GRN inference task.** Embeddings are computed by integrating the gene co-expression network and TF-target network. TF-target interactions are then inferred using the similarity score defined by computing the scalar product for each TF-gene interaction. The inferred GRN is compared with the reference GRN of *yeast* obtained from the YEASTRACT database.

3.6.2 Protein function prediction

Protein function prediction is a crucial part of genome annotation. It is generally accomplished through manual or computational annotation [Coz+13]. The former approach is the gold standard because it is implemented by expert annotators and yields high-quality curated results [SW10]. Nonetheless, this approach is expensive and laborious, and thus, it is difficult to scale. Therefore, due to the availability of omics data, computational annotation methods have been developed to improve the accuracy of the protein function prediction [SW10]. Gene Ontology (GO) [Con04] is the most comprehensive resource. It has all the desirable properties of a functional classification system, including a controlled vocabulary describing the functional properties of biomolecules (e.g., genes, proteins, and RNA). Each ontology belongs to one of three categories: Molecular Function (MF), Cellular Component (CC), and Biological Process (BP).

We model the problem of protein function prediction as a multi-label node classification task. We use the learned features, $\mathbf{\Omega}_d$, to train a Support Vector Machine (SVM) classifier and predict probability scores for each protein. We use the SVM implementation provided in the LIBSVM package [CL11]. To measure the performance of the SVM on the embedding vectors, we adopt a 5-fold Cross Validation (CV) process [CBP16; GBB18]. We split all the annotated proteins into a training set, comprising 80%, and a test set, comprising the remaining 20% ones. We train the SVM on the training set and predict the function of the test proteins. We use the standard Radial Basis Function (RBF) for SVM and perform a nested 5-fold cross-validation within the training set to select the optimal hyperparameters of the SVM via grid search (i.e., δ in the RBF kernel and the weight regularization parameter C). All performance results are averaged over 10 different CV trials. The evaluation metrics micro- Area Under Precision-Recall Curve (AUPR), Macro (M)-AUPR , Accuracy (ACC), and F1 utilized for protein function prediction are mentioned in Section 3.7. The illustration of the protein function prediction task

is shown in Figure 3.9.

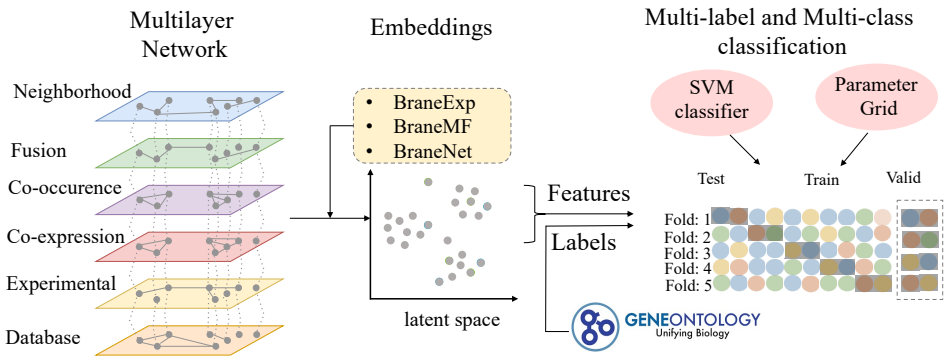


Figure 3.9: **Illustration of the protein function prediction task.** For each protein, a d-dimension embedding vector is computed from STRING PPI networks using the proposed models. These features, along with known GO terms (labels), are given as input to the SVM model. 5-fold cross-validation (CV) is performed over 10 trials.

3.6.3 Clustering of functionally related proteins

Proteins interacting in PPI networks are seen to be physically or functionally related [Saf+14]. After performing integration, we would like to investigate the clustering of related proteins in the embedding space. We expect the embeddings to preserve/enhance this relatedness among proteins. First, we utilize k-means clustering [AV07] and then evaluate the obtained clusters using the YeastEnrichr tool [Kul+16; Sub+05]. The illustration of the clustering task is shown in Figure 3.10.

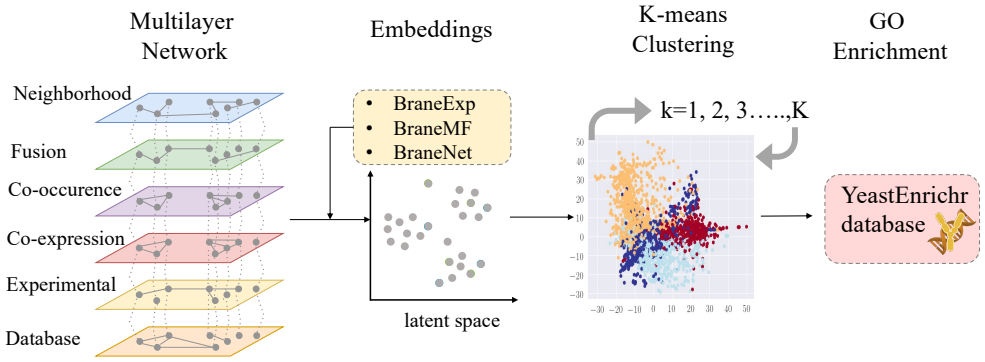


Figure 3.10: **Illustration of the clustering task.** For each protein, a d-dimension embedding vector is computed from STRING PPI networks using the proposed models. These features are utilized to perform clustering with the k-means algorithm. The functional analysis of the obtained clusters is performed using the YeastEnrichr tool.

3.6.4 Protein Protein Interaction (PPI) prediction

This task is very similar to the link prediction task usually performed to evaluate graph representation learning methods in social network analysis domain [Per+17; GL16; ÇM20]. We estimate the probability of interactions between proteins in a multilayer PPI network. Nevertheless, it is well known that PPI networks are incomplete. Therefore, to predict the unseen interactions, we train a model on a part of PPIs. We divide the PPI interaction in a given multilayer network G into two parts to form training and test sets by randomly removing 50% of the edges (PPIs). We keep the network connected during the process by selecting the largest connected component. Additionally, we obtain the same number of edges that do not exist in G. With the PPI prediction task, we aim to predict missing or new edges. The proposed models learn node embeddings, but for this task, we are interested in having edge embeddings. Therefore, we obtain edge embeddings using the node features that are computed using the proposed models. The node embedding vectors $\mathcal{E}[u]$ and $\mathcal{E}[v]$ of nodes u and v are converted into edge feature vectors by applying the coordinate-wise operations as given below:

Consider nodes u and v. To compute features for a candidate edge between node u and v, we first extract node features $\mathbf{\Omega}[u]$ and $\mathbf{\Omega}[v]$ respectively. Then, we perform coordinate-wise operations (Hadamard product, cosine distance) as follows:

1. Hadamard product:
$$\mathbf{\Omega}[u]_i \times \mathbf{\Omega}[v]_i \tag{3.18}$$
2. Cosine distance:
$$1 - \frac{\mathbf{\Omega}[u]_i . \mathbf{\Omega}[v]_i}{\|\mathbf{\Omega}[u]_i\| \ \|\mathbf{\Omega}[v]_i\|},$$

where $\mathbf{\Omega}[u]_i$ is the i^{th} index the embedding vector $\Omega[u]$ of node u. We train the *Logistic Regression* (LR) model with L2 regularization. We report the Area Under Curve (AUC) score of the operators showing the best performance for each method. The illustration of PPI prediction is shown in Figure 3.11.

3.6.5 Network reconstruction

A key challenge in biology is to understand complex molecular interactions among genes and proteins. Despite of the large available omics datasets, the complete interactome of the cellular processes is still understudied. We aim to reconstruct the interactome using heterogeneous omics data sources. Network reconstruction is a process that involves inferring a network by performing integration on a multilayer network. The inferred network is expected to reflect the edges in all layers of the input network. In this study, we aim to reconstruct a protein-protein interaction network using the trained embeddings.

Firstly, expecting to capture the interaction patterns for each protein from all layers in the input network. Secondly, we expect to reconstruct the reference network to evaluate if the embeddings can recover the known interactions. Lastly,

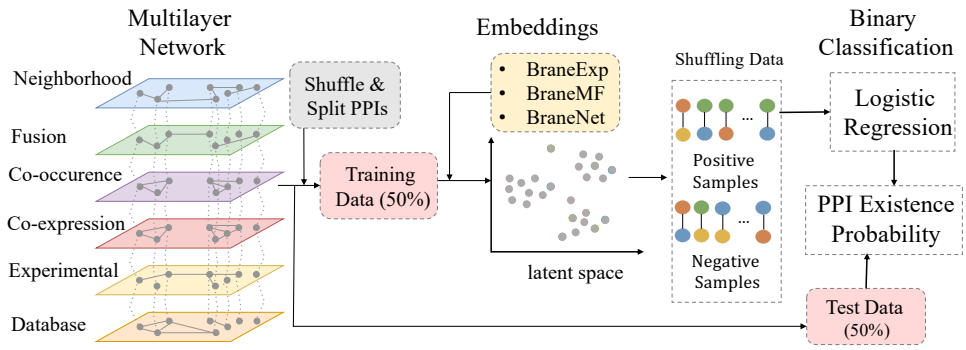


Figure 3.11: **Illustration of protein-protein interaction (PPI) prediction.** 50% of PPI interactions from STRING multilayer PPI network are removed to form the training and test datasets. Embeddings are learned for each protein using the training data as input. From the learned embeddings, edge features are computed for PPI present in the input graph (label = 1) and the edges that do not exist in the input graph (label = 0). A *Logistic Regression* (LR) model is trained, and the PPI existence probability is calculated. The prediction is performed on the test data, and the performance is evaluated by computing AUC.

we are also interested in investigating newly inferred edges. We consider that if embedding vectors for a gene pair are close in embedding space, they are likely to share some hidden information. So, to identify such pairs, we compute euclidean distance or dot product from embedding vectors for each gene-gene pair. Then, based on the distribution of distances, we apply a threshold to select the interactions with the least distance. We assume that if the embedding vectors for a protein pair are close in the embedding space, they are likely to be related and share some hidden information. To identify such pairs, we compute the scalar product from embedding vectors for each protein-protein pair. Further, we evaluate the reconstruction by comparing the inferred PPIs with the reference integrated network from the STRING database. The illustration of the network reconstruction task is shown in Figure 3.12.

3.7 Evaluation Metrics

Evaluation metrics play an important role in achieving the prediction model during the learning phase. Hence, the selection of appropriate evaluation metrics is crucial for discriminating and obtaining the optimal classifiers [HS15]. Evaluation metrics have been employed in two stages, which are the training stage (i.e., the learning process of the model) and the testing stage (i.e., prediction evaluation) [HS15]. In the binary classification problem, the evaluation of the prediction performance can be defined based on the confusion matrix, as shown in Table 3.4.

- tp: True positive
- tn: True negative

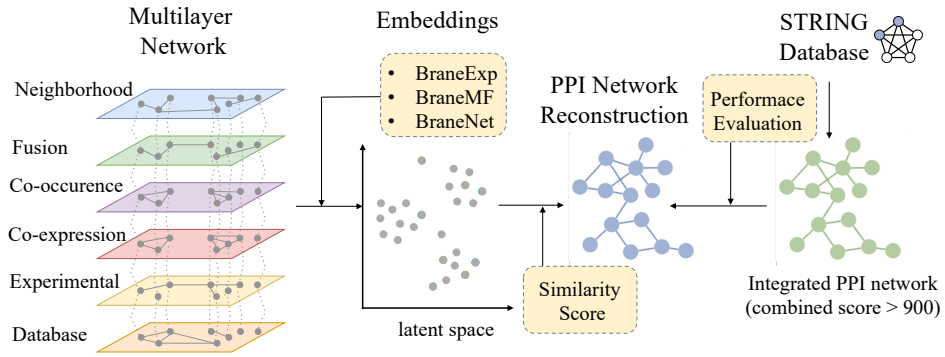


Figure 3.12: **Illustration of the network reconstruction task.** The Protein embeddings are computed by integrating the multilayer STRING PPI network. The PPI network is reconstructed using the similarity score defined by computing the cosine similarity for each protein-protein interaction. The inferred PPI network is compared with the reference PPI of *yeast* obtained from the STRING database.

- fp: False positive
- fn: False negative

tp and tn denote the number of positive and negative instances that are correctly classified. Meanwhile, fp and fn denote the number of misclassified negative and positive instances, respectively. From Table 3.4, several commonly utilized metrics can be generated as shown in Table 3.5 to evaluate the performance of the classifier focusing on different aspects of the evaluation. Due to multiclass classification problems, a few of the metrics listed in Table 3.5 have been extended for multi-class classification evaluations (see the last four metrics) [HS15].

The Area Under the Precision-Recall curve (*AUPR*) and the Area Under the Receiver Operating Curve (*AUROC*) are the popular ranking type metrics that are utilized to construct an optimized learning model and also for comparing learning algorithms.

The AUPR curve is calculated as the area under the Precision and Recall (PR) curve. A PR curve shows the trade-off between Precision and Recall across different thresholds. The x-axis of a PR curve is the recall, and the y-axis is the precision. A PR curve starts at the upper left corner, i.e., the point (recall = 0, precision = 1), which corresponds to a threshold of 1. Whereas a PR curve ends at the lower right, where recall = 1 and precision is low. This corresponds to a threshold of 0. The points that create the PR curve are obtained by calculating the precision and recall for different thresholds between 1 and 0.

The area under the receiver operating characteristic (AUROC) is a performance metric that you can use to evaluate classification models. AUROC is thus a performance metric for “discrimination": it tells you about the model’s ability to

discriminate between different classes. The AUROC is calculated as the area under the ROC curve. A ROC curve shows the trade-off between true positive rate (TPR) and false positive rate (FPR) across different decision thresholds. A ROC curve always starts at the lower left-hand corner, i.e., the point (FPR = 0, TPR = 0) which corresponds to a threshold of 1. A ROC curve always ends at the upper right-hand corner, i.e., the point (FPR = 1, TPR = 1) which corresponds to a threshold of 0. The points in between, which create the curve, are obtained by calculating the TPR and FPR for different thresholds between 1 and 0.

	Actual Positive Class	**Actual Negative Class**
Predicted Positive Class	True positive (tp)	False negative (fn)
Predicted Negative Class	False positive (fp)	True negative (tn)

Table 3.4: **Confusion Matrix for Binary Classification.**

Matthews Correlation Coefficient (MCC): MCC is another way to evaluate performance. It measures the differences between the actual values and the predicted ones. The advantages of MCC over Precision-based metrics are shown in recent articles [CJ20]. For the edges with similarity score $\delta_{i,j}$ higher than a threshold($\theta \in \{0.1, 0.2, \ldots, 0.9\}$), we compute MCC,

$$\frac{\text{True Negative (TN)} \times \text{TP} - \text{False Negative (FN)} \times \text{FP}}{\sqrt{(\text{TP}+\text{FP})(\text{TP}+\text{FN})(\text{TN}+\text{FP})(\text{TN}+\text{FN}))}}. \tag{3.19}$$

Metrics	Formula	Definition
Accuracy (acc)	$\frac{tp+tn}{tp+tn+fp+fn}$	measures the ratio of correct predictions over the total number of instances evaluated
Sensitivity (sn)	$\frac{tp}{tp+fn}$	measures the fraction of positive patterns that are correctly classified; True Positive Rate (TPR)
Specificity (sp)	$\frac{tn}{tn+fp}$	measures the fraction of negative patterns that are correctly classified; True Negative Rate (TNR)
Precision (p)	$\frac{tp}{tp+fp}$	measures the positive patterns that are correctly predicted from the total predicted patterns in a positive class
Recall (r)	$\frac{tp}{tp+tn}$	measures the fraction of positive patterns that are correctly classified
F-1 score ($f1$)	$\frac{2\times p\times r}{p+r}$	represents the harmonic mean between recall and precision values
Averaged Accuracy (ACC)	$\frac{\sum_{i=1}^{y}\frac{tp_i+tn_i}{tp_i+tn_i+fp_i+fn_i}}{y}$	average effectiveness of all y classes
Averaged Precision (P)	$\frac{\sum_{i=1}^{y}\frac{tp_i}{tp_i+fp_i}}{y}$	average of per-class precision
Averaged Recall (R)	$\frac{\sum_{i=1}^{y}\frac{tp_i}{tp_i+tn_i}}{y}$	average of per-class recall
Averaged F-1 ($F1$) score	$\frac{\sum_{i=1}^{y}\frac{2\times p_M\times r_M}{p_M+r_M}}{y}$	average of per-class f-1 score

Table 3.5: **Metrics for classification evaluations**. Each class $i \in Y$; tp_i - true positive for Y_i; ; fp_i - false positive for Y_i; fn_i – false negative for Y_i; tp_i - true negative for Y_i; and M macro-averaging

Next, we define the metric utilized for the evaluation of the performance in clustering task 3.6.3 using the Enrichment Score (ES).

Enrichment Score (ES)

Consider a gene set G_k, where $k = 1,\dots,K$. G_k consists of a list of n_k genes (g_{kj}), i.e., $G_k = \{g_{kj} : j = 1,\dots,n_k\}$. Each gene in the set is represented in the ranked list L. The set of genes outside of G_k is defined as as $\bar{G}_k = \{\bar{g}_{kj} : 1,\dots,n - n_k\}$. The Enrichment Score (ES) for a given gene set G_k is given as:

$$ES = \sup_{1 \le i \le n} \left(F_i^{G_k} - F_i^{\bar{G}_k} \right),$$

where sup(·) is the supremum, i represents the position in L, and

$$F_i^{G_k} = \frac{\sum_{t=1}^{i} |s_t|^{\alpha} . \mathbb{1}_{gene_t \in G_k}}{\sum_{t=1}^{n} |s_t|^{\alpha} . \mathbb{1}_{gene_t \in G_k}},$$

$$F_i^{\bar{G}_k} = \frac{\sum_{t=1}^{i} |s_t|^{\alpha} . \mathbb{1}_{gene_t \in \bar{G}_k}}{n - n_k},$$

where $\mathbb{1}$ is the indicator function for the membership in a given gene set. s_t is given by correlation of g_{kj} and weighted by α [Sub+05; Iri+09].

4

Results and Discussion

This chapter is dedicated to the results of downstream tasks that are explained in the previous chapter. First, we provide the parameter selection strategy adopted for the proposed models as well as baseline models for each downstream task. Secondly, we test the performance of the proposed models for each downstream task, comparing them to the baseline models.

4.1 Parameter Selection

All multilayer network integration methods that are based on machine learning and mathematical models require tuning a certain set of parameters to learn protein features. From the model description given by each method in their respective research article, we highlight parameters that could be tuned to improve the model performance. The remaining parameters that have little impact on performance have been set to default values. In Table 4.1, we provide the required parameters that are tuned for each method. Note that the representation in Table 4.1 is simplified to show the dependency of baseline methods on the different parameters. For some methods, a direct comparison of parameters is not possible since they may share different parameter spaces. We adopt this approach to simplify the parameter selection strategy.

1. Embedding size (d): the size of the protein feature vector. Its dimensionality is typically much lower than that of the ambient space. We chose $d \in \{128, 256, 512, 1024\}$.

2. Walk length (w): it is a parameter to select the length of a node set one would like to obtain while performing random walks on a graph. For instance, a *walk* of length 5 is defined as "proteinA proteinB proteinC proteinD proteinX". We chose $w \in \{15, 20\}$.

3. Window size (t): the number of nodes (proteins) that will be used to determine the context of each node (protein). For instance, in a *walk* of length

Method	d	t	w	n	σ	e	γ	p_r	r	b
BraneMF	✓	✓	✗	✗	✗	✗	✓	✗	✗	✗
BraneNet	✓	✓	✗	✗	✗	✗	✓	✗	✗	✗
BraneExp	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗
Graph2GO	✓	✗	✗	✗	✓	✓	✗	✗	✗	✗
MultiVERSE	✓	✓	✗	✗	✓	✗	✗	✓	✓	✗
OhmNet	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗
Multi-n2v	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗
MultiNet	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗
DeepNF	✓	✗	✗	✗	✗	✓	✗	✓	✓	✓
Mashup	✓	✗	✗	✗	✗	✗	✗	✓	✗	✗
SNF	✓	✓	✗	✗	✗	✗	✗	✗	✓	✗

Table 4.1: **Overview of parameters considered for tuning**. Green-coloured ticks indicate that the method depends on the respective parameters. Red crosses show that the method does not depend on a particular parameter.

3, such as, "proteinA proteinB proteinC", a window size of 2 would mean your samples are like (proteinA proteinB) or (proteinB proteinC). We chose $t \in \{2, 4, 6, 8, 10\}$.

4. Number of walks (n): this parameter allows the selection of the number of random walks that will be sampled per node (protein). We chose $n \in \{10, 20\}$.

5. Learning rate (σ): it is a hyper-parameter that controls how much we are adjusting the weights in the learning process with respect to the gradient of the loss function. A lower σ represents a smaller step along the downward slope. We chose $\sigma \in \{0.1, 0.01, 0.001, 0.0001\}$.

6. Number of epochs (e): an epoch is one learning cycle where the learner sees the whole training data set and calculates the error rate. We chose $e \in \{60, 80\}$.

7. Restart probability (p_r): this parameter is used in models that rely on Random Walks with Restart (RWR). While performing random walks, at each iteration, the walker can also restart by jumping to any randomly selected node in the graph with a defined restart probability. We chose $p_r \in \{0.8, 0.85, 0.9, 0.95\}$.

8. Gamma (γ): it is a weighting factor used by our BraneMF model. It represents the power to be applied on the singular values ($\bar{\Sigma}$) used for the computation of the embeddings (please see line 5 of Alg. 3). We chose $\gamma \in \{0, 0.25, 0.50, 0.75, 1\}$.

9. Number of samples (r): it is the parameter to choose the number of times we would like to perform Random Walk with Restart (RWR). We chose $r \in \{2, 3, 4, 6\}$.

Method	Parameters
SNF	$r = 6; t = 10$
Mashup	$p_r = 0.8$
deepNF	$b = 64; r = 4; p_r = 0.95; e = 80$
MultiNet	$w = 20; n = 20; t = 10$
Multi-n2v	$w = 20; n = 20; t = 10$
OhmNet	$t = 10; w = 20; n = 10$
MultiVERSE	$t = 10; r = 4; p_r = 0.95, \sigma = 0.01$
Graph2GO	$e = 80; \sigma = 0.01$
BraneExp	$w = 20; n = 10; t = 10$
BraneMF	$t = 10; \gamma = 1$
BraneNet	$t = 10; \gamma = 0.5$

Table 4.2: **Model parameters.** The above table shows the parameters selected for the clustering and protein function prediction tasks.

10. Batch size (b): it is the number of samples that will be used for training at one time. We chose $b \in \{32, 64, 128\}$.

4.2 Clustering of Biological Related Proteins

As mentioned in Section 3.6.3, we perform clustering of node embeddings to obtain groups of similar nodes. Moreover, we are interested in identifying if the obtained clusters share a biological similarity. Therefore, we examine the ability of the learned features to cluster functionally related proteins. We have performed an unsupervised clustering of proteins with the learned embeddings, $\mathbf{\Omega}_d$, using the k-means++ clustering algorithm [AV07] for $k \in \{20, 40, 60, 80, 100\}$. We execute the clustering algorithm 20 times to take into account the randomness in the process. For each of the obtained clusters, we perform Gene Set Enrichment Analysis (GSEA) [Sub+05] using the "GO_Biological_Process_2018" library of the *YeastEnrichr* database [Kul+16] consisting of 1,649 GO terms. A cluster is considered to be enriched if at least one GO term in a cluster is significantly enriched (adjusted P-value < 0.05). For all significantly enriched clusters, the performance is measured by the enrichment score (ES). For the selected parameters (Section 4.1), we report the best performance for each method (Table 4.2).

4.2.1 Comparison to baseline methods

We have compared the performance of clustering to eight baseline methods which have been introduced in Section 2.5. Table 4.3 shows the clustering results measured by the average enrichment score (ES). We have reported the standard deviation across 20 simulations. The definition of the ES metric is provided in

Method	$K = 40$	$K = 60$	$K = 80$	$K = 100$
SNF	7.2 ± 10.2	23.1 ± 6.6	15.2 ± 2.4	43.1 ± 4.1
Mashup	21.5 ± 0.6	30.1 ± 3.0	38.8 ± 0.4	41.9 ± 0.4
deepNF	$\mathbf{25.7 \pm 1.7}$	22.7 ± 1.2	26.3 ± 1.0	26.2 ± 1.1
MultiNet	20.4 ± 2.2	22.5 ± 0.5	45.4 ± 0.0	45.4 ± 0.0
Multi-n2v	16.6 ± 2.4	24.7 ± 0.9	$\underline{46.6 \pm 0.1}$	$\underline{46.6 \pm 0.1}$
OhmNet	15.1 ± 7.2	35.1 ± 0.2	45.1 ± 0.3	45.1 ± 0.4
MultiVERSE	16.4 ± 10.4	13.7 ± 0.9	20.1 ± 0.0	20.4 ± 0.0
BraneExp	21.0 ± 6.7	$\underline{41.9 \pm 1.4}$	45.4 ± 0.0	45.4 ± 0.0
Graph2GO	21.3 ± 1.7	22.5 ± 11.9	25.5 ± 11.9	30.4 ± 7.7
BraneNet	20.1 ± 4.02	37.5 ± 3.3	42.4 ± 5.2	45.6 ± 2.04
BraneMF	$\underline{24.05 \pm 9.3}$	$\mathbf{46.2 \pm 5.5}$	$\mathbf{48.9 \pm 4.28}$	$\mathbf{49.5 \pm 3.38}$

Table 4.3: **Clustering: comparison to baselines.** Performance of the proposed models compared to the baselines, measured by ES standard deviation computed for all 20 runs of k-means++ clustering algorithm. The numbers in bold indicate the best performance and underlined numbers indicate the second-best performance.

Section 3.7. The Enrichr algorithm [Kul+16] first ranks the genes based on a measure of correlation with a continuous phenotype. Then, the entire ranked list is used to assess how the genes of each gene set are distributed across the ranked list. To do this, the algorithm walks down the ranked list of genes, increasing a running-sum statistic when a gene belongs to the set and decreasing it when the gene does not. The enrichment score is the maximum deviation from zero encountered during that walk. The ES reflects the degree to which the genes in a gene set are over-represented at the top or bottom of the entire ranked list of genes. A set that is not enriched will have its genes spread more or less uniformly through the ranked list. An enriched set, on the other hand, will have a larger portion of its genes at one or the other end of the ranked list. A higher value of ES shows that clusters are enriched with a high number of over-represented terms [Kul+16].

In Table 4.3, it is observed that for $K = 60, 80$, and 100 the mean ES score of BraneMF is the highest, and the mean ES of BraneExp and BraneNet are the second highest. For $K = 40$, deepNF is the best-performing method, and BraneMF shows the second-best performance. Comparing BraneMF to the other proposed models, BraneMF's mean ES score exceeds three and four units BraneExp and BraneMF, respectively. However, overall from all the results, it was observed that for each K all the methods have at least one significantly enriched cluster. Nevertheless, this is our preliminary analysis to show the ability of the learned embeddings to cluster functionally related proteins.

From the above results, we ask our second biological question. Can the learned embeddings be used as features to predict protein functions? To answer this, in the next section, we apply the learned embeddings in the protein function prediction task.

4.3 Protein Function Prediction

We now investigate the ability of learned features to predict protein functions. Our multilayer-based integration approach allows us to obtain low-dimensional combined protein features from different PPI networks that can be used for function prediction. In the previous section, we have shown the efficacy of the learned embeddings to capture rich features that are biologically relevant and represent their interaction patterns and protein similarities. Hence, we would like to investigate the reliability of these features to predict protein functions. We model the problem of protein function prediction as a multi-label node classification task. We use the learned features, $\mathbf{\Omega}_d$, to train an SVM classifier and predict the probability scores for the functional annotations of each protein. The functional annotation groups (level I, II, and III) of BP, MF, and CC are used as class labels to train the SVM classifier (see Section 3.2). We use the SVM implementation provided in the LIBSVM package [CL11].

To measure the performance of the SVM on the embedding vectors, we adopt a 5-fold cross-validation (CV) process [CBP16; GBB18]. We split all the annotated proteins into a training set, comprising 80%, and a test set, comprising the remaining 20% ones. We train the SVM on the training set and predict the function of the test proteins. We use the standard radial basis kernel (RBF) for SVM and perform a nested 5-fold cross-validation within the training set to select the optimal hyperparameters (i.e., c and g) of the SVM via grid search. The g parameter defines how far the influence of a single training example reaches, with low values meaning 'far' and high values meaning 'close'. The g parameters can be seen as the inverse of the radius of influence of samples selected by the model as support vectors. The c parameter trades off the correct classification of training examples against the maximization of the decision function's margin. For larger values of c, a smaller margin will be accepted if the decision function is better at classifying all training points correctly. A lower c will encourage a larger margin, therefore, a simpler decision function at the cost of training accuracy. In other words, c behaves as a regularization parameter in the SVM. The grid values for g and c are given as $g \in 0.001, 0.01, 0.1, 1$ and $c \in 0.1, 1, 10, 100$, respectively.

The standard metrics to evaluate multi-class classification performance are the F1 score, Precision, Recall, and Accuracy. Their definitions are provided in Section 3.7. For each CV trial, we compute m-AUPR, M-AUPR, ACC, and F1 scores. The final results are given by averaging over 10 different CV trials. In the next section, we investigate the added value of integration for protein function prediction.

4.3.1 Single layer network *vs* multilayer network

We investigate the added value of integration for protein function prediction. We measure the performance for each level of the respective functional category independently. Accordingly, we show the results for nine datasets in total, namely levels I, II, and III of BP, MF, and CC. We evaluate the performance of the proposed models BraneExp and BraneMF. Note that, BraneNet is an early instance of BraneMF's integration for homogeneous multilayer networks. We compare BraneMF, BraneNet (early integration), and late integration (Section 4.3.2). First, we learn protein features for each network layer using the respective proposed method and then compare the performance of the features learned from individual input networks to the integrated ones. The evaluation of results is done by computing the F1 score (see Section 4.3). The performance of BraneExp and BraneMF in the protein function prediction task using single network embeddings and integrated embeddings is shown in Figures 4.1 and 4.2, respectively.

We observe that for both methods, integration outperforms individual network embeddings in the protein function prediction task. Looking at the performance, it is observed that CC has overall higher F1 scores compared to the BP and MF levels. There could be two possibilities. First, the biological significance behind this could be that cellular compartments inform us of the cellular location, which is more specific, e.g., nucleus, cytoplasm, and mitochondria. Nevertheless, biological processes and molecular functions are broader and more overlapping. Second, computationally, the number of classes in CC is lower than in BP. For instance, level I CC has 181 classes, while BP has 855 classes. Also, when we look at the individual network layers, it is observed that the 'Experimental', 'Co-expression', and 'Database' networks demonstrate good performance in all three levels, whereas the 'Fusion' network gives the lowest score. This indicates the importance of the first three networks in the function prediction task, compared to the 'Neighborhood', 'Fusion', and 'Co-occurrence' networks.

From the above experiment, we conclude that the integrated embeddings outperform single-layer embeddings in the classification. In the following subsection, we explore early and late integration strategies and compare their performance to BraneMF.

4.3.2 Integration strategies

Multilayer network integration strategies can be classified as early, intermediate, or late integration [LWN18; GP15]. In early integration methods, datasets are combined into a single dataset on which the model is built and the features are learned. In the late network integration strategy, a model for each network is built individually, and these individual network features are then combined. In intermediate integration, the data is combined through a joint model inference. Indeed, there is great value in developing efficient intermediate-level integration approaches [ZLX20], capable of handling heterogeneous data and providing insights into the functional categories of proteins (e.g., representation of

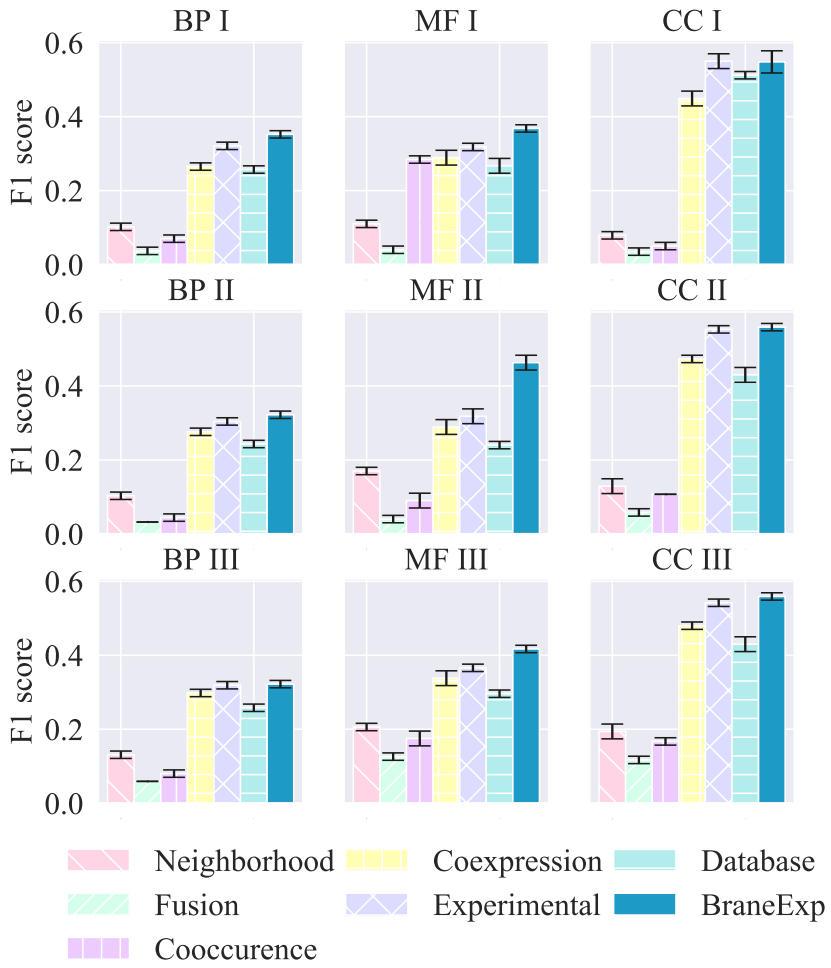


Figure 4.1: **BraneExp: single-layer *vs* multilayer.** Performance of BraneExp applied on individual yeast STRING networks, measured by the F1 score. Parameters: $\gamma = 1, t = 10, d = 128$. Error bars show the standard deviation across 10 CV trials.

system-level inter-relationships within biomolecules).

Early integration is performed before the modeling process, for example, merging all networks into one. On the contrary, late integration is done after the modeling process is applied to each network, and then it concatenates the obtained features. BraneMF is an intermediate integration model where integration is performed in the learning process of embedding computation. To show the effectiveness of the intermediate level of integration, we have compared BraneMF with BraneMF-early and BraneMF-late. In BraneMF-early, the PPMI matrix is computed from the adjacency matrix of the network obtained by taking the union of all six network layers. Then, d-dimensional protein features are learned. In BraneMF-late, the protein features are learned independently for each layer, and the final features are obtained by taking their average. The performance is evaluated by computing the F1 score and Accuracy metrics. As we can observe

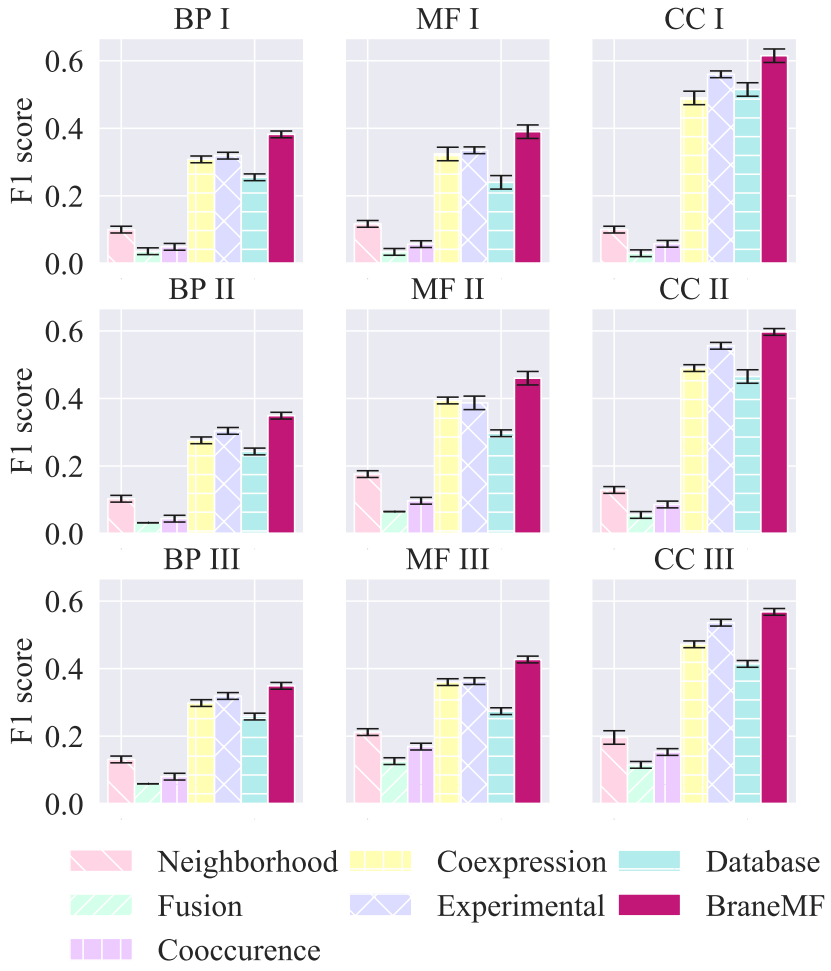


Figure 4.2: **BraneMF: single-layer *vs* multilayer.** We compare the cross-validation performance of BraneMF on individual yeast STRING networks, measured by F1 score. Parameters: $\gamma = 1, t = 10, d = 128$. The error bars show the standard deviation across 10 CV trials. Error bars show the standard deviation across 10 CV trials.

in Figure 4.3, BraneMF outperforms early and late integration strategies for all three levels of BP, MF, and CC. There is an increase of 2% in the accuracy of BP I when compared to BraneMF-early and an increase of 10% compared to the BraneMF-late integration model. Also, the performance of BraneMF for MF and CC is significantly higher than BraneMF-early and BraneMF-late under F1 and ACC scoring schemes. Hence, BraneMF's improvement can be partially attributed to the fact that separately computing the random walk matrices of each individual layer uncovers compressed topological patterns that are difficult to identify in the combined network (BraneMF-early) model where different edge types are not distinguished. Moreover, BraneMF has the advantage over late integration to benefit from capturing inter-layer correlation of modalities at the feature level which is challenging for late integration.

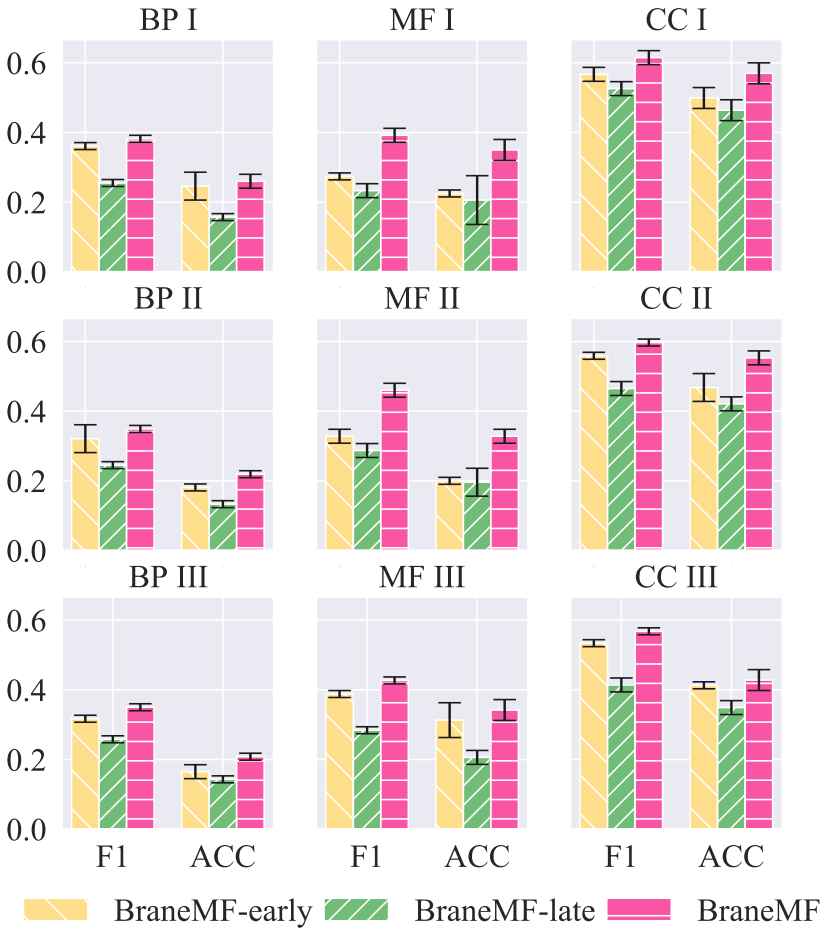


Figure 4.3: **Integration strategies: early, late and intermediate**. Performance of BraneMF compared to early and late integration, measured by the F1 score. Parameters: $\gamma = 1, t = 10, d = 128$. The error bars show the standard deviation across 10 CV trials. Error bars show the standard deviation across 10 CV trials.

4.3.3 Comparison to baseline methods

We compare the performance of the proposed models in the protein function prediction task to eight baseline methods which are introduced in Section 2.5. The results for BP, MF, and CC for levels I, II, and III are shown in Tables 4.4, 1, and 2 respectively [1]. The performance is measured by m-AUPR, M-AUPR, F1, and ACC scores. The definitions of the evaluation metrics are described in Section 3.7. We have reported the average scores with standard deviation across 5 CV trials.

For levels I, II, and III of the BP dataset, BraneMF shows comparable performance to the baseline methods. Regarding the F1 scores, BraneMF outperforms by

[1] Tables 1 and 2 are given in Appendix I.

4%, 2%, and 1% points Graph2GO and Multi-n2v, the second-best performing models. Similarly, BraneMF achieves higher performance for m-AUPR, M-AUPR, and ACC scores. Similarly, for MF and CC datasets, BraneMF retains its best performance. For MF, BraneExp is the second best-performing model for all three levels, while in CC datasets, Graph2GO and BraneExp show good performance. Overall, we observe that protein function prediction based on BraneMF substantially outperforms other integration methods in assigning a previously unseen protein to its known functional categories in a CV experiment.

Note that, the embedding size (d) for all the methods is empirically selected as 128. We further investigate the effect of d on protein function prediction for the proposed models and the baselines. The respective results are shown in Figure 1[2]. The performance is measured by the F1 score. For BraneMF, we see no significant change in performance with respect to d. However, for Mashup, SNF, deepNF, and MultiVERSE, the classification performance is increased with d.

[2]Figure 1 is shown in Appendix I

Method	m-AUPR	M-AUPR	F1	ACC
		BP I		
SNF	0.176 ± 0.01	0.224 ± 0.01	0.199 ± 0.01	0.150 ± 0.00
Mashup	0.362 ± 0.02	0.240 ± 0.01	0.277 ± 0.00	0.161 ± 0.01
deepNF	0.427 ± 0.02	0.260 ± 0.01	0.341 ± 0.01	0.211 ± 0.02
MultiNet	0.415 ± 0.02	0.257 ± 0.01	0.335 ± 0.01	0.212 ± 0.03
Multi-n2v	0.417 ± 0.02	0.250 ± 0.01	0.331 ± 0.01	0.201 ± 0.02
OhmNet	0.361 ± 0.02	0.255 ± 0.01	0.321 ± 0.01	0.063 ± 0.01
MultiVERSE	0.353 ± 0.02	0.223 ± 0.02	0.312 ± 0.01	0.117 ± 0.01
BraneExp	0.454 ± 0.02	$\underline{0.280 \pm 0.01}$	$\underline{0.352 \pm 0.01}$	0.220 ± 0.08
Graph2GO	$\underline{0.458 \pm 0.02}$	0.279 ± 0.01	0.340 ± 0.01	$\underline{0.249 \pm 0.02}$
BraneMF	$\mathbf{0.504 \pm 0.02}$	$\mathbf{0.303 \pm 0.01}$	$\mathbf{0.382 \pm 0.01}$	$\mathbf{0.260 \pm 0.02}$
		BP II		
SNF	0.220 ± 0.01	0.260 ± 0.01	0.220 ± 0.01	0.140 ± 0.01
Mashup	0.385 ± 0.02	0.337 ± 0.01	0.260 ± 0.01	0.130 ± 0.01
deepNF	0.464 ± 0.01	0.381 ± 0.01	0.309 ± 0.01	0.154 ± 0.01
MultiNet	0.458 ± 0.02	0.378 ± 0.01	0.323 ± 0.01	0.178 ± 0.01
Multi-n2v	$\underline{0.494 \pm 0.01}$	$\underline{0.406 \pm 0.01}$	$\underline{0.329 \pm 0.01}$	0.171 ± 0.01
OhmNet	0.382 ± 0.01	0.325 ± 0.01	0.285 ± 0.01	0.027 ± 0.01
MultiVERSE	0.387 ± 0.02	0.329 ± 0.01	0.293 ± 0.01	0.093 ± 0.01
BraneExp	0.474 ± 0.02	0.391 ± 0.01	0.322 ± 0.01	$\underline{0.204 \pm 0.03}$
Graph2GO	0.487 ± 0.02	0.398 ± 0.02	0.317 ± 0.01	0.185 ± 0.03
BraneMF	$\mathbf{0.524 \pm 0.02}$	$\mathbf{0.424 \pm 0.02}$	$\mathbf{0.349 \pm 0.01}$	$\mathbf{0.219 \pm 0.02}$
		BP III		
SNF	0.167 ± 0.00	0.224 ± 0.01	0.153 ± 0.01	0.052 ± 0.01
Mashup	0.484 ± 0.02	0.450 ± 0.01	0.289 ± 0.01	0.144 ± 0.01
deepNF	0.535 ± 0.01	0.478 ± 0.01	0.318 ± 0.01	0.157 ± 0.01
MultiNet	0.555 ± 0.01	0.496 ± 0.01	$\underline{0.343 \pm 0.01}$	$\underline{0.189 \pm 0.01}$
Multi-n2v	0.560 ± 0.02	0.504 ± 0.01	0.341 ± 0.01	0.185 ± 0.02
OhmNet	0.439 ± 0.01	0.411 ± 0.01	0.300 ± 0.01	0.010 ± 0.00
MultiVERSE	0.455 ± 0.02	0.422 ± 0.01	0.315 ± 0.01	0.079 ± 0.01
BraneExp	0.537 ± 0.01	0.495 ± 0.01	0.330 ± 0.01	0.183 ± 0.02
Graph2GO	$\underline{0.568 \pm 0.01}$	$\underline{0.509 \pm 0.01}$	0.329 ± 0.01	0.162 ± 0.01
BraneMF	$\mathbf{0.585 \pm 0.01}$	$\mathbf{0.526 \pm 0.01}$	$\mathbf{0.350 \pm 0.01}$	$\mathbf{0.208 \pm 0.01}$

Table 4.4: **Protein function prediction (BP)**. The above table shows the results of PPI prediction for $d = 128$.

4.4 Network Reconstruction

We perform network reconstruction using the methodology described in Section 3.12. The performance of network reconstruction is evaluated using the reference PPI network from the STRING database [Szk+20]. This integrated STRING PPI network is weighted using a combined score [Szk+20]. We select the edges with a combined score greater than 900, and the reference network is obtained for 4,900 proteins and 63,309 PPIs. In practice, biological networks show small-world properties, where nodes are linked by a short chain of acquaintances. These properties could be extracted by focusing on important edges in the graph. In our context of binary inference, the Precision metric computes the accuracy of retrieving correctly inferred edges. Therefore, to evaluate the performance of graph inference and to retrieve such relevant information, we measure the Precision at top k inferred edges (Precision@k), that corresponds to the number of correctly inferred edges among the top k ones. For all the selected parameter settings, we calculate Precision@k for all the proposed and baseline models. The selected parameters are shown in Table 4.5.

Moreover, we remove the biased of considering only top edges by measuring the Area Under the Receiver Operating Characteristic (AUROC) curve and the Area under the Precision-Recall (AUPR) curve. The AUROC curve summarizes the trade-off between the true positive rate and the false positive rate for a predictive model using different probability thresholds. It is a plot of the false positive rate (x-axis) versus the true positive rate (y-axis) for a number of different candidate threshold values between 0.0 and 1.0. The true positive rate (sensitivity) describes how good the model is at predicting the positive class when the actual outcome is positive. The false positive rate (1-specificity) is the false-alarm ratio that calculates actual negatives that, the model has predicted incorrectly (Section 3.7).

The AUPR curve summarizes the trade-off between the true positive rate and the positive predictive value for a predictive model using different probability thresholds. Reviewing both Precision and Recall is useful in case of an imbalance in the observations between the two classes (Section 3.7). Typically, the number of negative classes is large as compared to the positive ones and, we are less interested to calculate the performance of the model to predict the negative class correctly. Precision and Recall metrics are only concerned with the correct prediction of the positive class. A Precision-Recall curve is a plot of the Precision (y-axis) and the Recall (x-axis) for different thresholds. A good Precision-Recall curve has higher values of AUPR.

4.4.1 Single layer network *vs* multilayer network

Similar to Section 4.3.1, we first investigate the added value of integration. We reconstruct the reference network using embeddings obtained from each layer and compare the performance of network reconstruction with the integrated embeddings. The respective results are shown in Figure 4.4. We observe

Method	Parameters
SNF	$r = 6; t = 10$
Mashup	$p_r = 0.95$
deepNF	$b = 64; r = 4; p_r = 0.95; e = 80$
MultiNet	$w = 20; n = 20; t = 10$
Multi-n2v	$w = 10; n = 20; t = 2$
OhmNet	$t = 2; w = 15; n = 10$
MultiVERSE	$t = 2; r = 2; p_r = 0.8, \sigma = 0.01$
Graph2GO	$e = 80; \sigma = 0.01$
BraneExp	$w = 15; n = 20; t = 10$
BraneMF	$t = 2; \gamma = 0.5$
BraneNet	$t = 2; \gamma = 0.5$

Table 4.5: **Model parameters.** The table shows the best-performing parameters for all models.

that integration outperforms network reconstruction when compared with single-layer embeddings. It could also be seen that embeddings learned from the 'Co-expression' and 'Experimental' networks could reconstruct the reference network more accurately than other input networks. This indicates the importance of the 'Co-expression', 'Experimental' networks in the network reconstruction task, compared to the 'Neighborhood', 'Fusion', 'Database', and 'Co-occurrence' networks.

4.4.2 Comparison to baseline methods

We perform network reconstruction for the proposed models and for the baseline methods using the parameters described in Section 4.5. We learn embeddings for all the eight baseline methods (Section 2.5) for $d \in \{128, 256, 512, 1024\}$. The results for Precision@k are shown in Figure 4.5. It is observed that all models nearly achieve 100% of Precision up to the top 1,000 edges. For $d = 128, 256$ and 512, BraneMF is the best-performing method, and Multi-node2vec (Multi-n2v) is the second best-performing method.

We compute the AUPR and AUROC for the proposed models and compare the performance with the baseline methods. The respective results are shown in Table 3 [3]. It is observed that the performance of various methods, namely SNF, Mashup, deepNF, Graph2GO, BraneMF, and BraneNet, increases with the embedding dimension. Whereas the methods MultiNet, Multi-n2v, OhmNet, and BraneExp achieve the best performance at lower dimensions. BraneExp has 5% higher AUPR

[3]Table 3 is given in Appendix III.

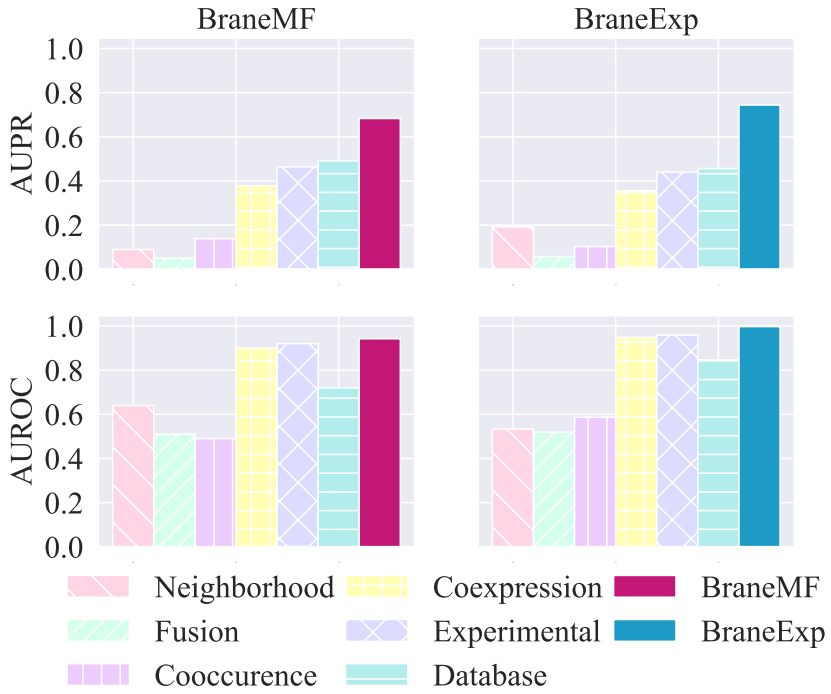


Figure 4.4: **Network reconstruction.** The figure shows the added value of integration for reconstructing the STRING PPI network. The performance is measured by AUPR and AUROC. The parameters for computing embeddings for each layer were kept the same as the integrated ones (Table 4.5) and $d = 128$.

score than BraneMF and Multi-n2v, and 6% higher AUROC score than BraneMF.

4.5 Gene Regulatory Network (GRN) Inference

To evaluate the performance of GRN inference, the reference Gene Regulatory Network (GRN for *yeast* was obtained from YEASTRACT database [Mon+20]. The reference GRN has 10,257 TF-target interactions for 114 TFs and 3,813 target genes. Integrated embeddings were learned from the co-expression network and TF-target network. The construction of input networks and the description of the task is provided in Section 3.8. Moreover, the GRN inference task is similar to the network reconstruction task, where we infer the network using the computed embeddings and evaluate its performance by reconstructing the reference network using these embeddings. We chose the same metrics of the network reconstruction task to evaluate GRN inference, i.e., Precision@k, AUROC, and AUPR. We learn the embeddings for the proposed and baseline methods for $d \in \{128, 256, 512, 1024\}$ using the same parameters as mentioned in Table 4.5.

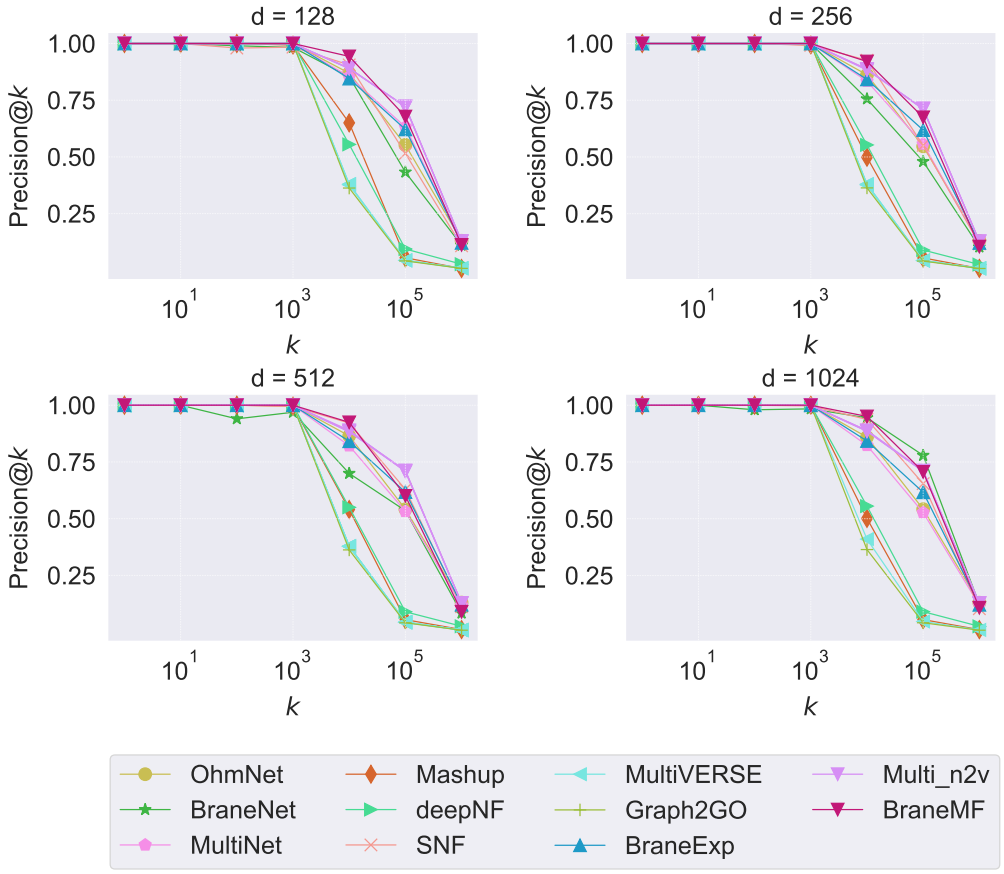


Figure 4.5: **Network reconstruction.** The figure shows the performance of the proposed and baseline models for reconstructing the STRING reference network. The performance is measured by computing the Precision@k. The x-axis represents the number of top k scoring edges inferred by the methods and the y-axis shows the Precision at these top k edges. All the methods are compared for $d \in \{128, 256, 512, 1024\}$.

4.5.1 Single layer network *vs* multilayer network

Similar to the network reconstruction task, we have investigated the GRN inference using single-layer embeddings versus integrated ones. The respective results are shown in Figure 4.6. It is observed that integrated embeddings outperform gene regulatory network inference. In both sub-figures, the embeddings learned from the co-expression network (transcriptomics) could infer GRN edges more accurately than the TF target network (genomics). However, when integrated, these embeddings could infer the top 1,000 edges accurately with the respective dataset using both BraneMF and BraneExp.

4.5.2 Comparison to baseline methods

We perform GRN inference for the proposed models and the baseline methods using the parameters described in Table 4.5. We learn embeddings for all the

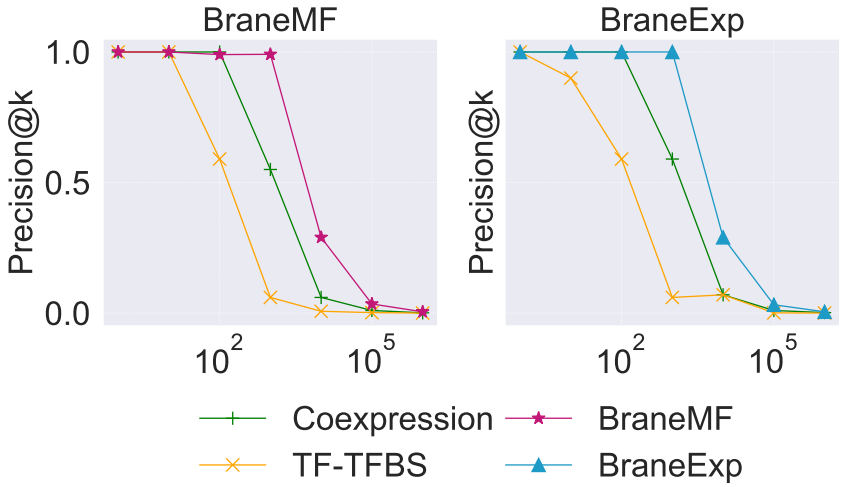


Figure 4.6: **GRN inference.** The figure shows the added value of the integration of the gene co-expression network and TF-target network. The performance is measured by the AUPR and AUROC curves. The parameters for computing embeddings for each layer were kept the same as for the case of network integration (Table 4.5) and $d = 128$.

eight baseline methods (Section 2.5) for $d \in \{128, 256, 512, 1024\}$. The results for Precision@k are shown in Figure 4.7. It is observed that all models except OhmNet, SNF, Mashup, and BraneNet achieve 100% of Precision up to the top 1,000 edges. The AUPR and AUROC curves are shown in Table 4[4]. SNF, Mashup, deepNF, Graph2GO, BraneMF, and BraneNet all exhibit improved performance with respect to dimension size. On the other hand, MultiNet, Multi-n2v, OhmNet, and BraneExp work well at smaller dimensions. Overall, BraneExp achieves the best performance for AUPR and AUROC metrics. It has a 5% higher AUPR score than BraneMF and Multi-n2v and a 6% higher AUROC score than BraneMF.

4.6 Protein-Protein Interaction (PPI) Prediction

The details for Protein-Protein Interaction (PPI) Prediction task are described in Section 3.6.4. In this task, our goal is to predict the missing (unseen) PPIs (edges) between proteins (nodes) using the learned features. We use PPIs from the 2015 and 2021 STRING networks to form training and test sets, respectively. We form the positive training set from PPIs that did not change from 2015 to 2021, and the positive test set from the PPIs that did not exist in 2015 but gained existence in 2021. The same number of PPIs that do not exist in both networks are sampled to generate negative instances for each training and test sets, respectively. We first learn embedding using proposed models and baseline methods for 2015 dataset. We use these embeddings to predict the new edges in 2021 PPI networks. The learned embeddings of protein u and v, given as $\mathbf{\Omega}_d[u]$ and $\Omega_d[v]$, are converted

[4]Table 4 is shown in Appendix III.

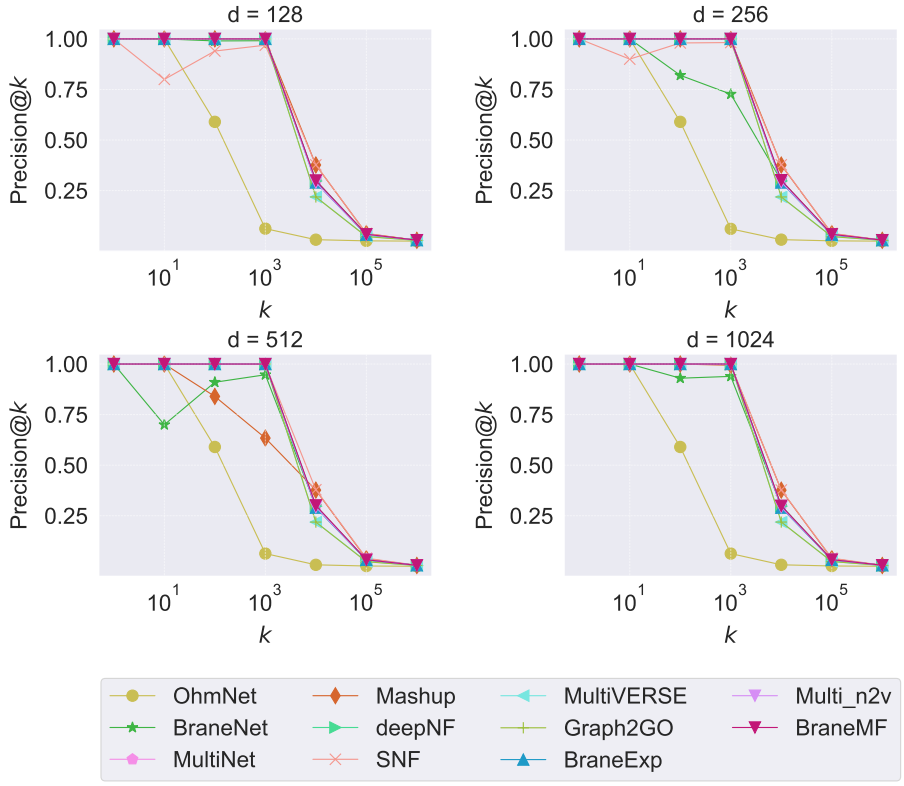


Figure 4.7: **GRN inference I.** The figure shows the performance of the proposed models and baseline models for reconstructing the yeast gene regulatory network. The performance is measured by computing Precision@k. The x-axis represents the number of top k scoring edges inferred by methods, and the y-axis shows the Precision at these top k edges. All the methods are compared for $d \in \{128, 256, 512, 1024\}$.

into edge feature vectors by applying the coordinate-wise Hadamard product or cosine similarity operations [GL16]. Definitions of these operations are given in Section 3.6.4. We perform the prediction task using logistic regression classifier with L2 regularization. The performance of PPI prediction is evaluated based on the AUROC and AUPR metrics. The results are shown in Table 4.6. We observe that BraneMF has competitive and consistent behavior across almost all evaluation metrics for the PPI prediction, achieving 1.5% higher performance (AUPR-H) than BraneExp, which is the second-best performing model. deepNF and Mashup also perform well under specific evaluation metrics.

4.7 Yeast Multi-omics Data Integration

As a case study, we selected the multi-omics data from yeast that contains transcriptomics, epigenetics, and metabolomics data. We learn integrated embeddings using BraneNet and BraneExp for wild-type yeast strains during a heat-shock time course of 0, 20, and 120 minutes. Note that BraneMF is exclusive to embed multi-

Method	AUPR-H	AUROC-H	AUPR-C	AUROC-C
SNF	0.637	0.628	0.575	0.559
Mashup	0.757	0.743	<u>0.712</u>	**0.707**
deepNF	0.764	0.747	0.490	0.480
Multi-Net	0.735	0.724	0.490	0.480
Multi-n2v	0.526	0.528	0.511	0.509
OhmNet	0.513	0.514	0.516	0.516
MultiVERSE	0.500	0.501	0.501	0.501
BraneExp	<u>0.777</u>	**0.760**	0.683	0.680
Graph2GO	0.721	<u>0.757</u>	0.502	0.498
BraneMF	**0.783**	0.747	**0.725**	<u>0.682</u>

Table 4.6: **PPI prediction performance.** Performance of BraneMF, compared to the baseline methods, measured by the AUROC and AUPR for the edge features computed by coordinate-wise operations given by Hadamard product (-H) and cosine similarity (-C). Bold: best score, underlined: second best score.

layer networks that share the same type of nodes. To learn embeddings, BraneNet could be an instance of BraneMF for heterogeneous multilayer networks. Here, BraneNet and BraneExp learn features for differentially expressed biomolecules showing heat stress response. We demonstrate the applicability of the learned features for targeted omics inference tasks: transcription factor (TF)-target prediction, Integrated Omics Network Factor (ION) inference, and module identification. The performance of BraneNet and BraneExp is compared with existing network integration methods. For this experiment, BraneNet is used as the base model, whereas BraneExp is used in the evaluation with other baselines, namely, MOSS [Gon+22], deepNF [GBB18], MultiNet [BK18], and OhmNet [ZL17].

4.7.1 Data description

Yeast multi-omics datasets are obtained by the same yeast sample presenting three basic layers of the transcriptional circuit, including one type of epigenetic modification (H4K12ac mark for identification of active promoters obtained from ChIP-Seq), gene expression (RNA-seq), and targeted metabolomics (NMR) [Nuñ+20]. The dataset is comprised of 7,126 genes, 1,970 H4K12ac peaks, and 37 metabolites. To obtain this data, the yeast culture flask was grown at 30°C until the exponential phase. This culture was split into three different flasks. One flask was maintained at 30°C and labeled as 0 minute (t0). The other two flasks were incubated at 39°C for 20 minutes (t20) and 120 minutes (t120), respectively. Aliquots from all three flasks (t0, t20, and t120) were collected for ChIP-seq (epigenomics), RNA-seq (transcriptomics), and NMR (metabolomics). This process was repeated four times to generate four biological replicates. The datasets were pre-processed using various bioinformatics tools [Nuñ+20]. These consistent datasets, dedicated to the study of the heat stress response, appear as a good candidate to test and evaluate our proposed omics data integration methodology. We recall the experimental

setup and summarize the workflow in Figure 4.8.

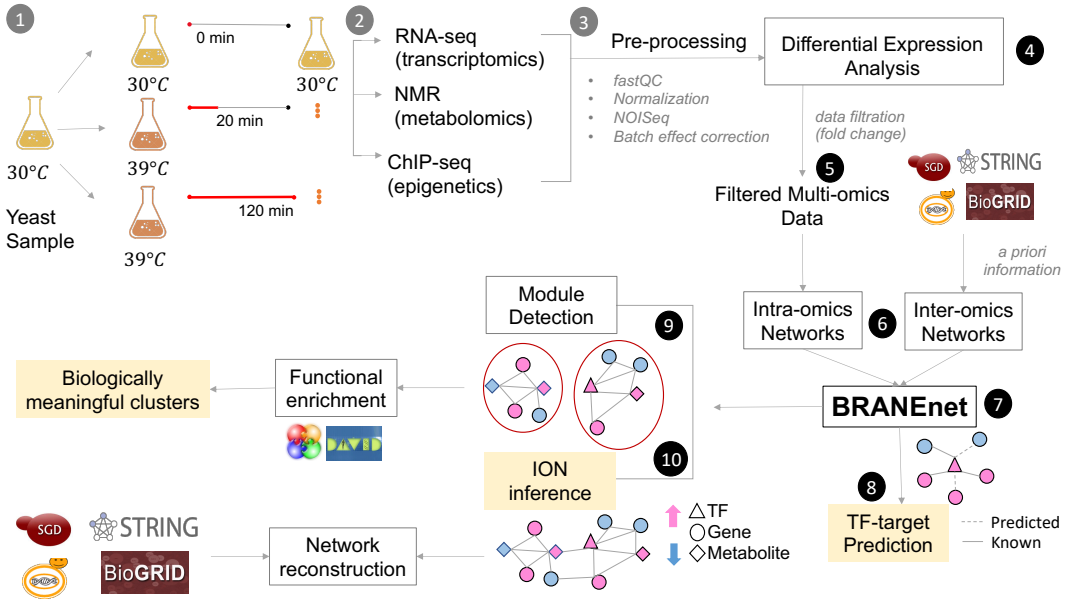


Figure 4.8: **Experimental design and BraneNet processing workflow**. The setup of wet-lab experiments (steps 1, 2, and 3) are taken from the data descriptor article [Nuñ+20]. Steps 4, 5, and 6 perform dataset collection and prepossessing before integration. (7) Learn embeddings using BraneNet. (8-10) Downstream bioinformatics tasks.

4.7.2 Differential expression analysis

Genes, metabolites, TFs are hereafter referred to as biomolecules. More generally, genes are regulated by TFs. Therefore, we separate TFs (genes coding for TFs) and non-TFs (genes not coding for TFs) from transcriptomics data. Now, to obtain differentially expressed biomolecules, we first take the average of control samples in the four biological replicates. Then, we compute the log2 of Fold Change (log2FC) for each bio-molecule in eight test samples (four in t20 and four in t120) by taking its ratio against the average of four control samples (t0). For each test sample, we select non-TFs if log2FC is higher than 2 (over-expressed) or lower than -2 (under-expressed). However, it is well known that expression TFs do not vary considerably as compared to non-TFs [Dal+12]. Therefore, we lower the threshold of log2FC for genes encoding for TFs (TFs). TFs were considered as differentially expressed if log2FC is higher than 1 (over-expressed) or lower than -1 (under-expressed). For metabolites and H4K12ac peaks, we choose the log2FC threshold similar to TFs. If a log2FC value is meaningful with respect to the above thresholds in at least one biological replicate, we consider the corresponding bio-molecule as differentially expressed.

4.7.3 Construction of intra-omics and inter-omics networks

Intra-omics networks are constructed using the same type of biomolecules, for example, gene-gene co-expression or metabolite-metabolite correlation networks. These networks are built on data obtained from multi-omics experiments, for instance, genomics, epigenetics, transcriptomics, proteomics, and metabolomics. We obtain differentially co-expressed biomolecules by computing the Pairwise Pearson correlation coefficient (ρ) [ZH05] of log2FC profiles described above, i.e., log2FC for the eight samples (four in t20 and four in t120). Two intra-omic elements were said to be correlated if the absolute value of ρ is higher than 0.8. These intra-omic correlation networks are represented as a set of adjacency matrices.

Inter-omics networks link biomolecules of different types. They are constructed using biological *a priori* information showing the presence of TF binding sites or H4K12ac epigenetic marks in the promoter of the gene, biochemical reactions within genes, and metabolites. This information can be acquired from various bioinformatics databases such as SGD [Che+98], YEASTRACT (Yeast Search for Transcriptional Regulators And Consensus Tracking) [Tei+18]), YeastPathways [Che+98], and BioCyc [Kar+19]. This *a priori* knowledge bridges the gap to relate two different omics types, for instance, gene-metabolite, TF-target, and gene-epigenetic mark. For each differentially expressed bio-molecule of one type (e.g., gene), we obtained its relationship with a bio-molecule of another type (e.g., TF and metabolite).

4.7.4 Downstream tasks

The embeddings learned from BraneNet and BraneExp can be used for various downstream tasks, for instance, TF-target prediction, ION inference, identification of biomarkers (e.g., heat stress-responsive genes/TFs), identification of biologically related clusters, and visualization. Their details are shown below.

TF-target prediction

To predict TF-targets, we adapt the traditional link prediction task [LK07] to TF-target networks. We use the largest connected component of the TF-target network. Then, we split the targets of each TF into two parts to form positive training and test sets by randomly removing 50% of them. The same number of TF-target pairs that do not exist are sampled to generate negative instances for each training and test sets. The learned embeddings $\mathbf{\Omega}_d$ are used to compute edge features. In particular, the embeddings of node i and j of size d, given by $\mathbf{\Omega}_d[i]$ and $\mathbf{\Omega}_d[j]$ respectively, are converted into edge feature vectors using element-wise operations [GL16; ÇM20] (i) *average*: $(\mathbf{\Omega}_d[i] + \mathbf{\Omega}_d[j])/2$; (ii) *weighted* L2: $|(\mathbf{\Omega}_d[i] - \mathbf{\Omega}_d[j])|^2$. Now, for each positive and negative test and training dataset generated above, edge features are computed. Then, we perform prediction using the logistic regression classifier with L2 regularization [Ped+11]. The performance is measured using the area under the Precision-Recall curve (AUPR) [FK15]. The

performance of BraneNet and BraneExp for TF-target prediction is compared with baseline methods.

ION inference

To infer an ION from the learned embeddings, the pairwise similarity score for nodes i and j is defined as:

$$\theta_{i,j} = \mathbf{\Omega}_d[i] \cdot \mathbf{\Omega}_d[j] = \sum_{i=1}^{d} \mathbf{\Omega}_d[i]\mathbf{\Omega}_d[j]. \tag{4.1}$$

To validate this network, we compare it with the gold-standard (GS) network of yeast that is built by combining networks from multiple databases, such as BIOGRID [Oug+21], STRING [Szk+20], and YEASTRACT [Mon+20]. The performance of ION inference is measured by computing the Matthews Correlation Coefficient (MCC) and the Precision@k (Section 3.7).

Module detection

Interestingly in biological networks, the clustering or community structure property is present, under which the graph topology is organized into modules commonly called communities or clusters. To obtain these modules, we first select the top-scoring edges ($\theta = 0.7$). Then we find clusters using a greedy modularity maximization algorithm [CNM04b]. We select the obtained modules having more than 10 nodes. To validate the obtained clusters, we investigated their biological meaningfulness by performing functional annotation enrichment analysis [Den+03; Bin+09].

Comparison to baseline methods

We compare the performance of link prediction and ION inference using the embeddings learned by BraneNet and BraneExp with existing multilayer network embedding methods. We choose OhmNet [ZL17], MultiNet [BK18], deepNF [GBB18] and MOSS [Gon+22] as our baseline methods. These network integration methods are not specifically developed for multi-omics integration considering biological *a priori* knowledge to learn node embeddings. Therefore, we adapt the existing methods for learning embeddings and performing downstream tasks by keeping the same empirical parameter settings as of BraneNet. Apart from deepNF, all baseline models mainly depend on the window size (t) and embedding dimension (d). For deepNF, we choose to keep the default model architecture configuration proposed by the authors [GBB18].

4.7.5 Results and discussion

We present the results of BraneNet applied to the yeast multi-omics dataset [Nuñ+20]. We have identified differentially expressed (DE) biomolecules as mentioned in Section 4.7.2. We have obtained 333 DE genes (non-TF) out of which

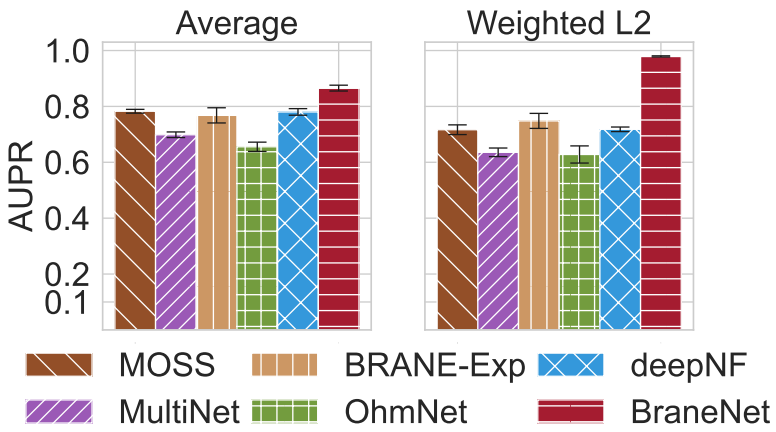


Figure 4.9: **Transcription factor (TF)-target prediction.** The figure shows the performance of BraneNet for TF-target prediction compared to baseline methods. The performance is measured by computing the AUPR score for both *average* and *weighted L2* coordinate-wise operations. The error bars show the standard deviation of the AUPR score for 10 runs.

310 are upregulated and 23 are downregulated, 55 DE TFs (50: over-expressed; 5: under-expressed), 30 DE metabolites (28: increased concentration; 2: decreased concentration). For the epigenetics data, we have observed that no H4K12ac peaks were differentially expressed. Therefore, we discard ChIP-Seq data and use only variable genes, TFs, and metabolites for the study of heat shock response. We then compute intra- and inter-omics networks as described in Section 4.7.3. To construct inter-omics relationships, we obtain known TF-target interactions from the YEASTRACT database [Tei+18]. Gene-metabolite and TF-metabolite associations were given by the participation of genes or TFs in the production and consumption of metabolites in biochemical reactions. This information was acquired from the YeastPathways database [Che+98]. Embeddings are then learned for each node, as discussed in Section 3.4. We use these embeddings to study different aspects of multi-omics data integration, namely, TF-target Prediction, ION inference, and module detection.

TF-target prediction

To perform TF-target prediction we compute node embeddings using BraneNet ($t = 3$, $b = 1$, and $d = 128$) and BraneExp ($w = 10$, $n = 20$, $t = 3$ and $d = 128$). For the same value of d, we learn node embeddings using each baseline method. The edge features are computed using the operators mentioned in Section 4.7.4. TF-target prediction is then performed using logistic regression, and its performance is measured using the AUPR score. Since we randomly remove 50% of targets for each TF, we repeat this process 10 times and report the average AUPR scores with standard deviation computed across 10 runs. The results for proposed models compared to the baseline models are summarized in Figure 4.9. The average AUPR of BraneNet is 10% improved compared to *average* (87%) to *weighted L2* (97.9%) operators. For the empirical parameter settings, the performance of BraneNet is

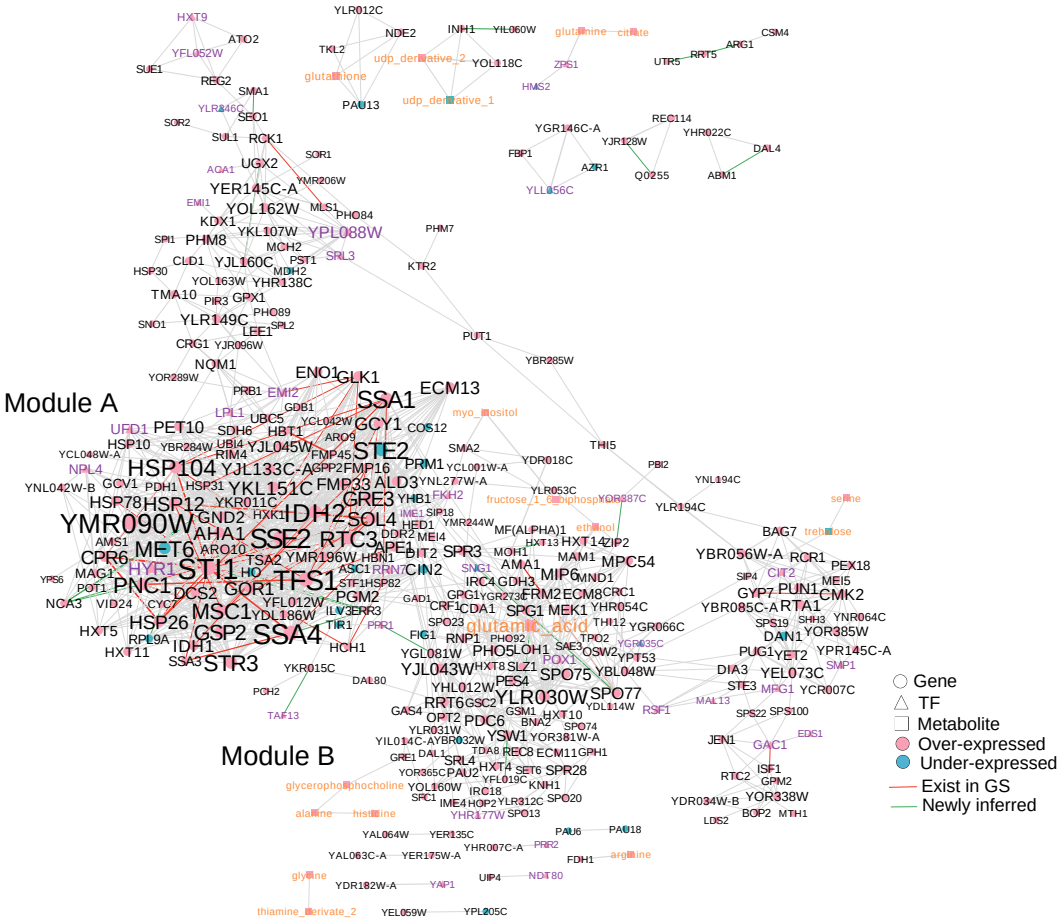
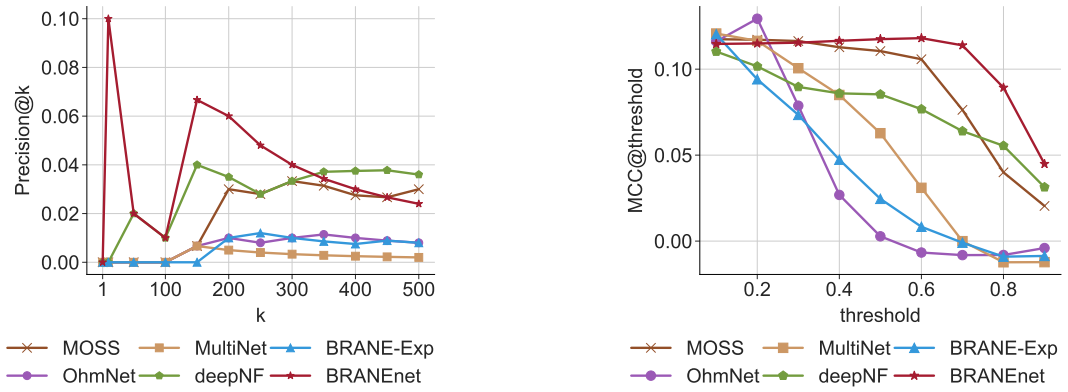


Figure 4.10: **ION visualization.** The figure shows the ION for yeast during time-dependent heat stress inferred using BraneNet. Node color, node shape, and edge color represent the information shown in legends. Since this network is the result of 250 simulations, the edge width is proportional to the number of times the edge occurred during each simulation. The label size of each node is proportional to its degree in the above network.

higher in both operators as compared to the baseline methods. *Weighted L2* score of BraneNet's is 20% higher than BraneExp, the second best performing model. Whereas BraneNet's *average* score is 10% higher than MultiNet, the second best performing model for *average*. The standard deviation of BraneNet for 10 runs is notably lower (except MultiNet with *average*) than all the other methods. Overall from Figure 4.9, we observe that BraneNet outperforms the baseline methods for both operators (i.e., *average* and *weighted L2*).

Integrated Omics Network (ION) inference

We infer an ION using the embeddings learned by BraneNet. To validate the performance of ION inference, we reconstruct the gold-standard (GS) using the learned embeddings. The performance is measured by computing Precision@k and MCC (Mathews Correlation Coefficient). We choose to study the top 500



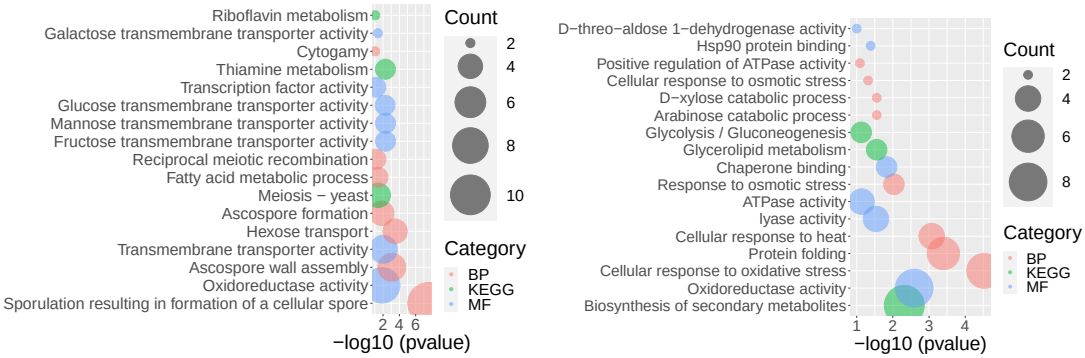
(a) Precision@k (b) MCC@threshold

Figure 4.11: **Network reconstruction.** (a) Precision@k for top 500 edges compared to baseline methods. The x-axis and y-axis represent the top k edges and the Precision@k edges, respectively. (b) MCC@threshold compared to baseline methods. The x-axis and y-axis represent the threshold of θ and the MCC@threshold, respectively.

edges of the inferred ION. We compare the performance of our model to the performance of the baseline methods used. The results are shown in Figure 4.11. The results of Precision@k show that BraneNet scores, up to top 320 edges, are higher than deepNF. BraneNet outperforms MOSS, OhmNet, MultiNet, deepNF, and BraneExp (Figure 4.11a). The results of MCC@threshold shows that BraneNet's performance for different thresholds (θ) is higher than OhmNet, MultiNet, deepNF, MOSS and BraneExp. As shown in Figure 4.11b, the MCC of BraneNet was gradually improved with increasing threshold and began to drop quite sharply at 0.6. For Precision@k metrics, deepNF is the second best performing model, whereas, for MCC@threshold MOSS is the second best performing method.

The network inferred by BraneNet with $\theta = 0.7$ is shown in Figure 4.10. Node color represents over- (pink) and under- (blue) expressed biomolecules. Node shape and label color represent gene (circle, black), TF (triangle, purple), and metabolites (square, orange). Edges existing in GS are given in red, whereas newly inferred edges are given in green. Edge width is represented by the similarity scores, while the node label size is proportional to its degree.

Using the inferred ION, we narrow down the search space from all differentially expressed biomolecules and identify potential biomarkers in heat stress response. We rank nodes based on their degree. Table 4.7 shows the obtained 21 biomolecules that could be potential biomarkers in heat stress response. We have investigated the participation of these genes during the heat-shock response in published literature. Using the BraneNet integrated tool, we are able to recover information from 11 different heat shock response studies. The references of these articles are given in Table 4.7. We have also validated our results by comparing



(a) Module A (b) Module B

Figure 4.12: **Functional enrichment of modules A and B.** The y-axis represents the list of significantly enriched terms, while the x-axis show their significance value (–log10 (p-value)). Different colors of circles indicate types of functional annotations. Biological process (BP) is given in pink, molecular function (MF) is given in blue, and KEGG pathways are shown in green. The size of the circle represents the number of differentially expressed genes/TFs.

them to another study of heat shock response [Cas+11]. We could find the potential biomarkers (Table 4.7) in the heat stress-responsive gene clusters that were identified in this study.

Biologically meaningful modules

To identify modules from the inferred ION, we perform community detection using the Clauset-Newman-Moore greedy modularity maximization algorithm [CNM04a]. We select modules with sizes of more than 10 nodes. We have obtained 6 modules. To know if the obtained modules are biologically meaningful, we perform functional enrichment analysis on the two largest modules. We select the terms with p-value lower than 0.05. Their enrichment results are shown in Figure 4.12. We can clearly see that module A is enriched with catabolic processes, including HSP90 and chaperone binding activity-related terms, while module B is enriched with transport and sporulation. The terms enriched in both these clusters have been discussed over the years in yeast heat-shock response studies [Cas+11; MGM12; Ver+12].

Parameter sensitivity analysis

To examine the added value of integration, we have learned node embeddings by considering only one layer of information, i.e., transcriptomics. First, we consider only gene expression data and learn node embeddings. Secondly, we add the *a priori* knowledge to the transcriptomics data and learn node features. We compare the ION performance of using only one layer of information with the integrated embeddings acquired from multiple layers. Figure 4.14 shows that ION reconstruction is improved with the integration. We then investigated the robustness in the performance of BraneNet with learned integrated embeddings.

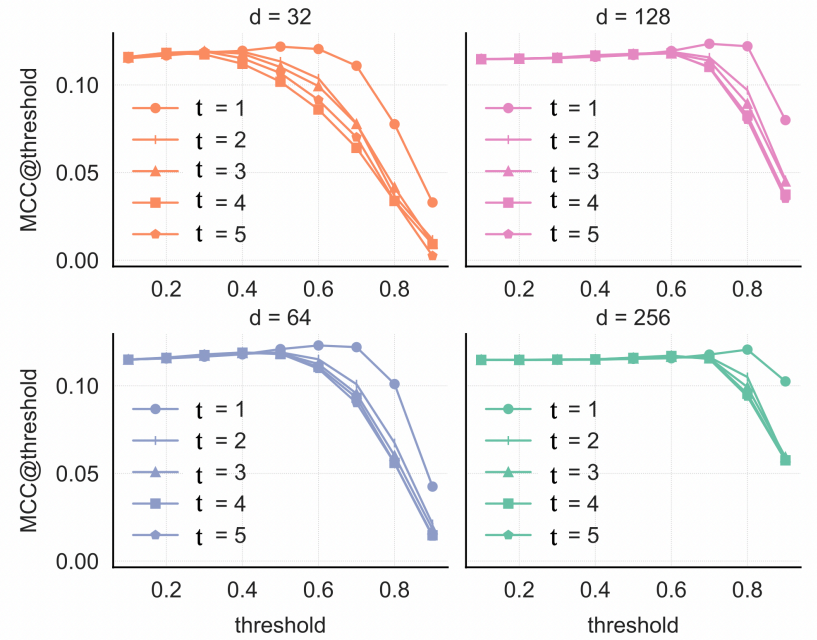


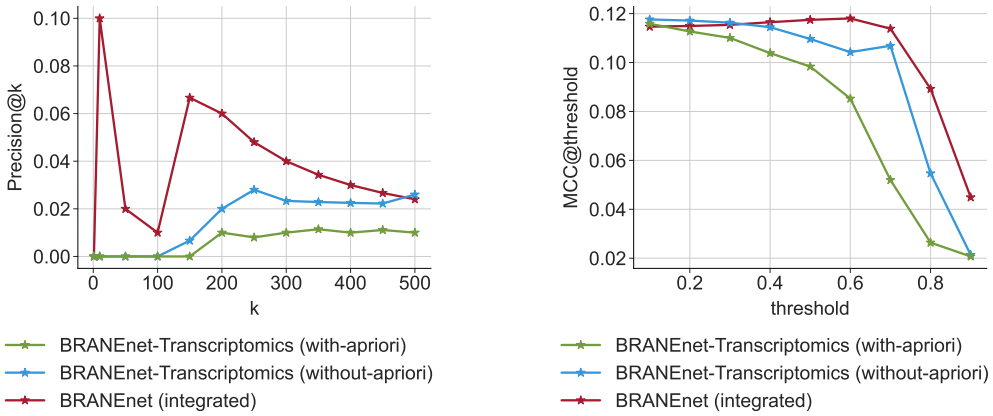
Figure 4.13: **Parameter sensitivity analysis for ION inference.** Node embeddings are computed using $t \in \{1, 2, 3, 4, 5\}$ and $d \in \{32, 64, 128, 256\}$. The performance is measured by computing MCC for different values of θ. The x-axis represents the MCC score at the threshold (θ) given in the y-axis.

We used grid-search to assess the uncertainty in the model outputs that is attributed to different values of the window size t and dimension d. We choose $t \in \{1, 2, 3, 4, 5\}$, $d \in \{32, 64, 128, 256\}$, and perform TF-target prediction and ION inference. The results for TF-target prediction are shown in Table 4.8. The mean AUPR in the given table for *average* and *weighted L2* is 83.8% and 96.1%, respectively, with a standard deviation of 2 percent.

On the other hand, the results for ION inference are shown in Figure 4.13. The performance is measured by MCC at different thresholds. From Figure 4.13, the optimal threshold for θ is between 0.6 and 0.8. We also see that the performance of MCC is increased concerning d and slightly decreased with t. From the parameter sensitivity analysis for both tasks, we see that our model has lower variance in the results with respect to different parameter settings. Therefore for the new datasets, we recommend users to consider the default parameter settings ($t = 3$ and $d = 128$).

Name	↑ / ↓	$\mathcal{D}$	Function	Modules (see Figure 4.10)
STI1	↑	40	Hsp90 cochaperone	A
SSA4	↑	39	Heat shock protein	A
TFS1	↑	46	Inhibitor of carboxy-peptidase Y , Ras GAP	A
YMR090W	↑	50	Unknown function	A
SSE2	↑	36	Hsp110 family member	A
IDH2	↑	37	Oxidative decarboxy-lation of isocitrate	A
SSA1	↑	40	ATPase	A
STE2	↓	36	Receptor for α-factor pheromone	A
HSP104	↑	33	Disaggregase	A
STI1	↑	41	Hsp90 cochaperone	A
MET6	↓	31	Cobalamin-independent methionine synthase	A
STR3	↑	30	Peroxisomal cysta-thionine beta-lyase	A
RTC3	↑	28	Unknown function	A
MSC1	↑	27	Unknown function	A
PNC1	↑	27	Nicotinamidase acid	A
GSP2	↑	30	GTP binding protein	A
GRE3	↑	31	Aldose reductase	A
YLR030W	↑	31	Unknown function	B
SOL4	↑	32	6-phospho-gluconolactonase	A
HSP12	↑	28	Heat shock protein	A
IDH1	↑	25	Oxidative decarboxy-lation of isocitrate	A

Table 4.7: **ION-based identification of potential biomarkers.** The table provides the names, over- (↑) or under- (↓) expressed, node degree in ION ($\mathcal{D}$), function, and BraneNet module information comparison with external studies of potential biomarkers during heat stress response in yeast.



(a) Precision@k (b) MCC@threshold

Figure 4.14: **Added value of integration.**(a)**Precision@k for top** 500 **edges.** The x-axis represents top k edges and y-axis represents Precision@k respectively. (b) **MCC@threshold.** The x-axis and y-axis represent threshold of θ and MCC@threshold, respectively.

Average

	d			
t	32	64	128	256
1	0.700	0.880	0.880	0.870
2	0.790	0.820	0.850	0.860
3	0.830	0.850	0.870	0.870
4	0.780	0.810	0.850	0.860
5	0.820	0.840	0.860	0.870

Weighted L2

	d			
t	32	64	128	256
1	0.980	0.982	0.983	0.983
2	0.916	0.945	0.966	0.968
3	0.956	0.967	0.979	0.979
4	0.852	0.938	0.966	0.968
5	0.952	0.968	0.981	0.981

Table 4.8: **Parameter sensitivity analysis for TF-target prediction.** Node embeddings are computed using $t \in \{1,2,3,4,5\}$ and $d \in \{32,64,128,256\}$. The performance is measured by computing the AUPR score for *average* and *weighted L2* coordinate-wise operations.

5

Concluding Remarks

5.1 Summary

Understanding the bio-molecular interactions represented by tightly controlled molecular networks is necessary for a thorough explanation of biological systems in an organism [Sub+20]. Huge amounts of diverse omics data have been introduced into the picture by the development of high-throughput technologies, and concurrently, promising paths have been opened up for their analysis and interpretation[Yue+20]. The value of multi-omics integration over single omics analysis has been demonstrated in numerous research. Such methods can shed light on the relationships between various biomolecules (proteins, RNAs, and metabolites), as well as the exchange of biological knowledge among them[Sub+20]. In the past years, network approaches have offered potential for integrative omics analysis, facilitating a new era of systems biology [Sub+20; Yan+18; Di +20]. Nevertheless, it is necessary to obtain informative representations (e.g., embeddings) for the nodes in the network (bio-molecules) and their proximity. Potentially, this would be possible by modeling biological data as a multilayer network and learning integrated embeddings that could effectively capture richer features and preserve biological information from each individual layer.

In this dissertation, we took inspiration from Graph Representation Learning (GRL) algorithms to encode graph structure into compact embedding vectors [HYL17b]. Our motivation is further extended towards leveraging closed forms of GRL methods that perform implicit matrix factorization, favouring intrinsic connection and interpretability of graph topology [LG14; Qiu+18]. We proposed three models for multilayer network embedding, namely, BraneExp, BraneNet, and BraneMF. Firstly in BraneExp, we took inspiration from expressive conditional probability models that relate nodes within random walk sequences. Hereby, we capitalize on exponential family distributions to capture interactions between nodes in random walks that traverse nodes within and across input network layers. More precisely, we introduced network integration with the concept of exponential family graph embeddings, which generalizes multilayer random walk-based GRL methods to an instance of exponential family conditional distribution.

Secondly, in BraneNet and BraneMF, we aimed to perform integration by leveraging a properly chosen multilayer random walk-based Positive Pointwise Mutual Information (PPMI) matrix. In the case of BraneNet, the model builds a supra-PPMI matrix that contains the normalized transition probability of node traversing in and across the network layers. The flexibility of random walks to traverse within and across layers allows us to capture inter- and intra-layer node neighbourhood information. The embeddings are learned by factorizing this supra-PPMI matrix. BraneMF builds PPMI matrices for each layer. These matrices are then efficiently integrated using joint matrix factorization. Both BraneMF and BraneNet learn embeddings by factorizing random walk-based PPMI matrices. However, the way integration is being done is different. BraneNet could be an early instance of BraneMF's integration strategy. Conceptually, BraneMF performs integration in a more effective manner, as it incorporates the concept of joint matrix factorization. More precisely, BraneMF brings the best of two worlds: expressiveness of well-celebrated random walk-based embedding models (e.g., DeepWalk, node2vec) and the solid formulation of matrix factorization—going further by extending them to integrate multiple input data sources.

Moreover, for all the proposed methods, we define the objective function in a way that is independent of downstream machine learning tasks, and the embeddings are learned in a purely unsupervised way. We have demonstrated the wide applicability of the proposed methods in exploiting functional analysis of proteins in PPI networks by studying the quality of clusteredness of functionally related proteins, the accuracy of predicting protein functions, and the inference of interactions in the reconstruction of the yeast interactome. Besides, while comparing against several baseline models, our methods have shown competitive performance in all downstream assessments. Nevertheless, our methods are not limited to the downstream tasks explored in this dissertation. They could be leveraged across a wide variety of omics integration tasks.

Lastly, we conclude that our models are simpler, depend on fewer parameters, and produce results comparable, if not better, to more complex methods. Although our formulation is expressive enough to capture these representations, its multiscale properties have certain limitations. For instance, BraneExp needs extensive simulation of random walks, which could be difficult while handling large networks. We rectified this limitation in BraneNet and BraneMF. Yet, BraneNet and BraneMF lack to capture long-range node dependencies (i.e., higher values of w), which could be interesting to study [CM20b]. Overall our three proposed models learn one global representation that coalesces all possible scales of network relationships. Hence, different scales of the representation are not independently accessible.

5.2 Perspectives

As future work, the current models could be improved concerning the limitations discussed in the above section. Moreover, we intend to conflate additional protein associations, such as post-transcriptional and post-translation regulation information, that may impact the functional relationships of proteins in the real world. Besides, it is also possible to take into account protein (node) features such as biochemical properties and protein sequences in the learning process [Zho+20]. These data types can provide insights towards more accurate predictions for functional analysis of proteins. The functionality and applicability of the proposed models are beyond embedding proteins, thus not limited to biological networks. Our proposed models are versatile in nature and can provide an effective, unified, and scalable network integration framework with diverse applications.

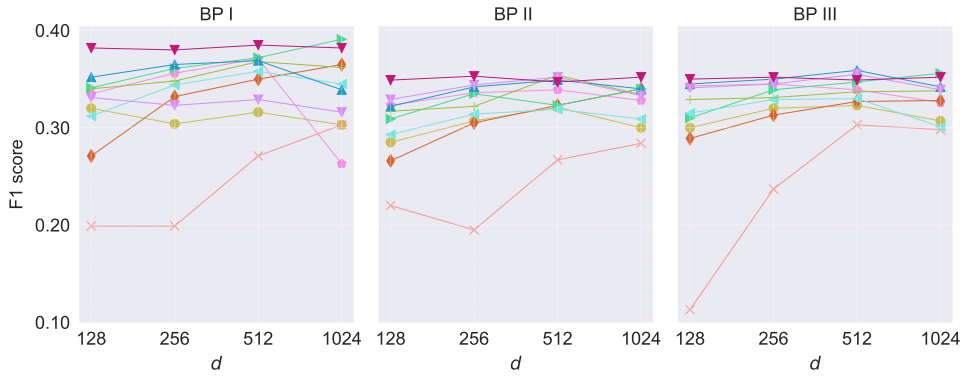
Appendix I

Method	m-AUPR	M-AUPR	F1	ACC
		MF I		
SNF	0.192 ± 0.01	0.120 ± 0.01	104 ± 0.00	0.142 ± 0.01
Mashup	0.255 ± 0.02	0.192 ± 0.01	0.263 ± 0.02	0.183 ± 0.04
deepNF	0.388 ± 0.03	0.235 ± 0.02	0.342 ± 0.02	0.273 ± 0.02
MultiNet	0.376 ± 0.02	0.249 ± 0.02	0.353 ± 0.02	$\underline{0.306 \pm 0.01}$
Multi-n2v	0.398 ± 0.03	0.203 ± 0.01	0.323 ± 0.01	0.262 ± 0.02
OhmNet	0.293 ± 0.02	0.211 ± 0.01	0.300 ± 0.01	0.020 ± 0.02
MultiVERSE	0.294 ± 0.03	0.192 ± 0.01	0.294 ± 0.01	0.145 ± 0.01
BraneExp	$\underline{0.410 \pm 0.02}$	$\underline{0.256 \pm 0.01}$	$\underline{0.368 \pm 0.01}$	0.303 ± 0.11
Graph2GO	0.404 ± 0.02	0.243 ± 0.02	0.355 ± 0.01	0.287 ± 0.11
BraneMF	$\mathbf{0.457 \pm 0.04}$	$\mathbf{0.278 \pm 0.02}$	$\mathbf{0.392 \pm 0.02}$	$\mathbf{0.350 \pm 0.03}$
		MF II		
SNF	0.185 ± 0.01	0.214 ± 0.01	0.126 ± 0.01	0.123 ± 0.00
Mashup	0.362 ± 0.02	0.310 ± 0.01	0.345 ± 0.02	0.228 ± 0.01
deepNF	0.428 ± 0.02	0.335 ± 0.01	0.396 ± 0.01	0.233 ± 0.01
MultiNet	0.440 ± 0.03	0.350 ± 0.02	0.416 ± 0.02	0.267 ± 0.04
Multi-n2v	0.447 ± 0.05	0.350 ± 0.03	0.398 ± 0.03	0.224 ± 0.06
OhmNet	0.342 ± 0.02	0.285 ± 0.01	0.334 ± 0.01	0.038 ± 0.01
MultiVERSE	0.363 ± 0.03	0.303 ± 0.02	0.348 ± 0.02	0.116 ± 0.01
BraneExp	$\underline{0.463 \pm 0.03}$	$\underline{0.368 \pm 0.02}$	$\underline{0.436 \pm 0.02}$	$\underline{0.294 \pm 0.10}$
Graph2GO	0.455 ± 0.02	0.359 ± 0.01	0.420 ± 0.01	0.292 ± 0.04
BraneMF	$\mathbf{0.518 \pm 0.02}$	$\mathbf{0.404 \pm 0.02}$	$\mathbf{0.460 \pm 0.02}$	$\mathbf{0.328 \pm 0.02}$
		MF III		
SNF	0.155 ± 0.01	0.147 ± 0.01	0.165 ± 0.01	0.037 ± 0.00
Mashup	0.393 ± 0.02	0.347 ± 0.02	0.333 ± 0.01	0.221 ± 0.02
deepNF	0.442 ± 0.03	0.389 ± 0.02	0.367 ± 0.01	0.236 ± 0.02
MultiNet	0.481 ± 0.03	0.419 ± 0.02	0.397 ± 0.02	0.309 ± 0.01
Multi-n2v	0.457 ± 0.03	0.406 ± 0.02	0.333 ± 0.02	0.150 ± 0.05
OhmNet	0.365 ± 0.02	0.343 ± 0.02	0.323 ± 0.01	0.014 ± 0.00
MultiVERSE	0.364 ± 0.02	0.337 ± 0.02	0.329 ± 0.01	0.131 ± 0.01
BraneExp	$\underline{0.517 \pm 0.03}$	$\underline{0.454 \pm 0.02}$	$\underline{0.417 \pm 0.02}$	$\mathbf{0.345 \pm 0.02}$
Graph2GO	0.501 ± 0.02	0.448 ± 0.02	0.396 ± 0.01	0.338 ± 0.02
BraneMF	$\mathbf{0.541 \pm 0.03}$	$\mathbf{0.473 \pm 0.02}$	$\mathbf{0.427 \pm 0.01}$	$\underline{0.342 \pm 0.02}$

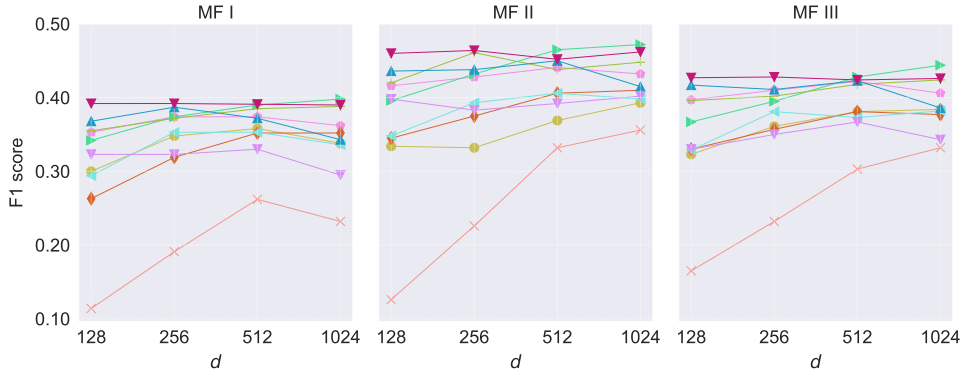
Table 1: **Protein function prediction (MF)** The above table shows the results of PPI prediction for $d = 128$.

Method	m-AUPR	M-AUPR	F1	ACC
		CC I		
SNF	0.178 ± 0.03	0.234 ± 0.02	0.206 ± 0.01	0.048 ± 0.01
Mashup	0.681 ± 0.02	0.414 ± 0.02	0.520 ± 0.02	0.432 ± 0.04
deepNF	0.715 ± 0.02	0.423 ± 0.02	0.564 ± 0.02	0.461 ± 0.02
MultiNet	0.662 ± 0.03	0.394 ± 0.02	0.532 ± 0.02	0.431 ± 0.06
Multi-n2v	0.663 ± 0.02	0.378 ± 0.03	0.511 ± 0.01	0.411 ± 0.02
OhmNet	0.590 ± 0.02	0.380 ± 0.02	0.512 ± 0.01	0.020 ± 0.03
MultiVERSE	0.586 ± 0.04	0.365 ± 0.02	0.502 ± 0.02	0.249 ± 0.08
BraneExp	0.694 ± 0.05	0.418 ± 0.03	0.548 ± 0.03	0.472 ± 0.03
Graph2GO	$\underline{0.732 \pm 0.03}$	$\underline{0.438 \pm 0.03}$	$\underline{0.564 \pm 0.02}$	$\underline{0.490 \pm 0.03}$
BraneMF	$\mathbf{0.812 \pm 0.02}$	$\mathbf{0.470 \pm 0.03}$	$\mathbf{0.615 \pm 0.02}$	$\mathbf{0.570 \pm 0.03}$
		CC II		
SNF	0.258 ± 0.02	0.323 ± 0.02	0.258 ± 0.02	0.040 ± 0.00
Mashup	0.681 ± 0.08	0.604 ± 0.02	0.505 ± 0.03	0.414 ± 0.03
deepNF	0.733 ± 0.01	0.617 ± 0.01	0.550 ± 0.01	0.431 ± 0.01
MultiNet	0.724 ± 0.02	0.610 ± 0.02	0.555 ± 0.01	0.453 ± 0.02
Multi-n2v	0.723 ± 0.01	0.592 ± 0.01	0.523 ± 0.01	0.458 ± 0.02
OhmNet	0.640 ± 0.02	0.544 ± 0.02	0.513 ± 0.01	0.093 ± 0.03
MultiVERSE	0.628 ± 0.02	0.529 ± 0.02	0.504 ± 0.01	0.249 ± 0.01
BraneExp	$\underline{0.749 \pm 0.02}$	0.630 ± 0.02	$\underline{0.559 \pm 0.01}$	0.462 ± 0.04
Graph2GO	$\underline{0.749 \pm 0.01}$	$\underline{0.631 \pm 0.01}$	0.549 ± 0.01	$\underline{0.481 \pm 0.04}$
BraneMF	$\mathbf{0.806 \pm 0.02}$	$\mathbf{0.666 \pm 0.02}$	$\mathbf{0.597 \pm 0.01}$	$\mathbf{0.553 \pm 0.02}$
		CC III		
SNF	0.364 ± 0.02	0.374 ± 0.02	0.342 ± 0.01	0.041 ± 0.01
Mashup	0.620 ± 0.01	0.555 ± 0.01	0.471 ± 0.01	0.345 ± 0.01
deepNF	0.655 ± 0.01	0.564 ± 0.01	0.508 ± 0.01	0.357 ± 0.02
MultiNet	0.688 ± 0.02	0.603 ± 0.03	0.546 ± 0.01	$\underline{0.400 \pm 0.02}$
Multi-n2v	0.660 ± 0.02	0.595 ± 0.02	0.491 ± 0.01	0.286 ± 0.05
OhmNet	0.588 ± 0.02	0.523 ± 0.03	0.493 ± 0.02	0.091 ± 0.01
MultiVERSE	0.598 ± 0.01	0.535 ± 0.02	0.496 ± 0.01	0.224 ± 0.01
BraneExp	$\underline{0.706 \pm 0.01}$	$\underline{0.634 \pm 0.01}$	$\underline{0.559 \pm 0.01}$	0.378 ± 0.02
Graph2GO	0.701 ± 0.02	0.623 ± 0.02	0.544 ± 0.01	0.387 ± 0.02
BraneMF	$\mathbf{0.734 \pm 0.01}$	$\mathbf{0.646 \pm 0.02}$	$\mathbf{0.568 \pm 0.01}$	$\mathbf{0.428 \pm 0.03}$

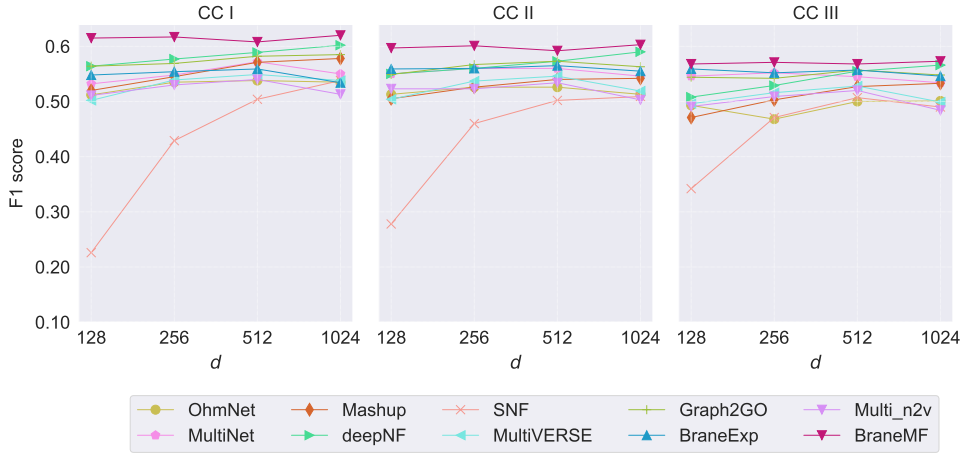
Table 2: **Protein function prediction (CC)** The above table shows the results of PPI prediction for $d = 128$.



(a) Biological Process: Level I, II and III



(b) Molecular Function: Level I, II and III



(c) Cellular Component: Level I, II and III

Figure 1: **Effect of parameter d on the classification**. The figure shows the effect of parameter d on the classification performance compared with the baseline methods. The x-axis represents the dimension of the protein features. The y-axis shows the F1 score of classification performed for respective datasets.

Appendix II

Method	$d = 128$	$d = 256$	$d = 512$	$d = 1024$
		AUPR		
SNF	0.427	0.479	0.543	0.524
Mashup	0.444	0.449	0.365	0.465
deepNF	0.602	0.607	0.603	0.608
MultiNet	0.542	0.461	0.444	0.438
Multi-n2v	0.690	0.676	0.674	0.670
OhmNet	0.532	0.525	0.520	0.520
MultiVERSE	0.257	0.257	0.256	0.256
Graph2GO	0.587	0.587	0.587	0.591
BraneExp	**0.744**	**0.741**	**0.739**	**0.738**
BraneMF	0.690	0.651	0.652	0.680
BraneNet	0.683	0.682	0.683	0.686
		AUROC		
SNF	0.967	0.977	0.964	0.937
Mashup	0.853	0.873	0.869	0.854
deepNF	0.941	0.947	0.943	0.942
MultiNet	0.990	0.986	0.985	0.984
Multi-n2v	0.993	0.993	0.993	0.993
OhmNet	0.995	0.995	**0.994**	**0.994**
MultiVERSE	0.526	0.527	0.525	0.528
Graph2GO	0.752	0.752	0.752	0.756
BraneExp	**0.997**	**0.997**	0.993	0.993
BraneMF	0.942	0.948	0.954	0.955
BraneNet	0.971	0.970	0.962	0.987

Table 3: **Network reconstruction II.** The figure shows the performance of the proposed models and baseline models for reconstructing the STRING reference network. The performance is measured by computing AUPR and AUROC. All the methods are compared for $d \in \{128, 256, 512, 1024\}$.

Appendix III

Method	$d = 128$	$d = 256$	$d = 512$	$d = 1024$
		AUPR		
SNF	0.157	0.159	0.160	0.160
Mashup	0.161	0.161	0.139	0.158
deepNF	0.233	0.233	0.233	0.233
MultiNet	0.201	0.201	0.201	0.201
Multi-n2v	0.214	0.214	0.214	0.215
OhmNet	0.238	0.198	0.227	0.229
MultiVERSE	0.157	0.157	0.156	0.156
Graph2GO	0.170	0.170	0.169	0.169
BraneExp	**0.300**	**0.299**	**0.299**	**0.299**
BraneMF	0.246	0.246	0.245	0.245
BraneNet	0.238	0.238	0.237	0.236
		AUROC		
SNF	0.591	0.568	0.544	0.549
Mashup	0.611	0.589	0.514	0.505
deepNF	0.561	0.576	0.584	0.590
MultiNet	0.599	0.598	0.619	0.592
Multi-n2v	0.577	0.548	0.611	0.584
OhmNet	0.592	0.509	0.548	0.539
MultiVERSE	0.526	0.527	0.525	0.528
Graph2GO	0.489	**0.636**	0.535	0.595
BraneExp	**0.625**	0.617	**0.621**	0.621
BraneMF	0.593	0.609	0.595	**0.629**
BraneNet	0.592	0.609	0.578	0.589

Table 4: **GRN inference II.** The figure shows the performance of proposed models and baseline models for reconstructing the yeast gene regulatory network. The performance is measured by computing AUPR and AUROC. All the methods are compared for $d \in \{128, 256, 512, 1024\}$.

References

[Cri70] Francis Crick. "Central dogma of molecular biology". In: *Nature* 227.5258 (1970), pp. 561–563.

[DC92] David DeMers and Garrison Cottrell. "Non-linear dimensionality reduction". In: *Advances in neural information processing systems* 5 (1992).

[Che+98] J Michael Cherry, Caroline Adler, Catherine Ball, Stephen A Chervitz, Selina S Dwight, Erich T Hester, Yankai Jia, Gail Juvik, TaiYun Roe, Mark Schroeder, et al. "SGD: Saccharomyces genome database". In: *Nucleic acids research* 26.1 (1998), pp. 73–79.

[BA99] Albert-László Barabási and Réka Albert. "Emergence of Scaling in Random Networks". In: *Science* 286.5439 (1999), pp. 509–512.

[Pag+99] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. *The PageRank citation ranking: Bringing order to the web.* Tech. rep. Stanford InfoLab, 1999.

[RS00] Sam T Roweis and Lawrence K Saul. "Nonlinear dimensionality reduction by locally linear embedding". In: *science* 290.5500 (2000), pp. 2323–2326.

[Wes+01] Douglas Brent West et al. *Introduction to graph theory*. Vol. 2. Prentice hall Upper Saddle River, 2001.

[Hub+02] Tim Hubbard, Daniel Barker, Ewan Birney, Graham Cameron, Yuan Chen, L Clark, Tony Cox, J Cuff, Val Curwen, Thomas Down, et al. "The Ensembl genome database project". In: *Nucleic acids research* 30.1 (2002), pp. 38–41.

[Den+03] Glynn Dennis, Brad T Sherman, Douglas A Hosack, Jun Yang, Wei Gao, H Clifford Lane, and Richard A Lempicki. "DAVID: database for annotation, visualization, and integrated discovery". In: *Genome biology* 4.9 (2003), pp. 1–11.

[BO04] Albert-Laszlo Barabasi and Zoltan N Oltvai. "Network biology: understanding the cell's functional organization". In: *Nature reviews genetics* 5.2 (2004), pp. 101–113.

[CNM04a] Aaron Clauset, M. E. J. Newman, and Cristopher Moore. "Finding community structure in very large networks". In: *Physical Review E* 70.6 (Dec. 2004).

[CNM04b] Aaron Clauset, Mark EJ Newman, and Cristopher Moore. "Finding community structure in very large networks". In: *Physical review E* 70.6 (2004), p. 066111.

[Con04] Gene Ontology Consortium. "The Gene Ontology (GO) database and informatics resource". In: *Nucleic acids research* 32.suppl_1 (2004), pp. D258–D261.

[Pee05] Dana Pe'er. "Bayesian network analysis of signaling networks: a primer." eng. In: *Science's STKE : signal transduction knowledge environment* 2005 (281 Apr. 2005), pl4.

[Sub+05] Aravind Subramanian, Pablo Tamayo, Vamsi K Mootha, Sayan Mukherjee, Benjamin L Ebert, Michael A Gillette, Amanda Paulovich, Scott L Pomeroy, Todd R Golub, Eric S Lander, et al. "Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles". In: *Proceedings of the National Academy of Sciences* 102.43 (2005), pp. 15545–15550.

[ZH05] Bin Zhang and Steve Horvath. "A general framework for weighted gene co-expression network analysis". In: *Statistical applications in genetics and molecular biology* 4.1 (2005).

[SMR06] Oksana Samko, A David Marshall, and Paul L Rosin. "Selection of the optimal parameter value for the Isomap algorithm". In: *Pattern Recognition Letters* 27.9 (2006), pp. 968–979.

[Sau+06] Lawrence K Saul, Kilian Q Weinberger, Fei Sha, Jihun Ham, and Daniel D Lee. "Spectral methods for dimensionality reduction." In: *Semi-supervised learning* 3 (2006).

[Yan+06] Shuicheng Yan, Dong Xu, Benyu Zhang, Hong-Jiang Zhang, Qiang Yang, and Stephen Lin. "Graph embedding and extensions: A general framework for dimensionality reduction". In: *IEEE transactions on pattern analysis and machine intelligence* 29.1 (2006), pp. 40–51.

[AV07] David Arthur and Sergei Vassilvitskii. "K-Means++: The Advantages of Careful Seeding". In: *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. SODA '07. New Orleans, Louisiana: Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035. ISBN: 9780898716245.

[Hub+07] Wolfgang Huber, Vincent J Carey, Li Long, Seth Falcon, and Robert Gentleman. "Graphs in molecular biology". In: *BMC Bioinformatics* 8.S8 (2007).

[LK07] David Liben-Nowell and Jon Kleinberg. "The link-prediction problem for social networks". In: *Journal of the American society for information science and technology* 58.7 (2007), pp. 1019–1031.

[Hoo08] L Hoopes. "Introduction to the gene expression and regulation topic room". In: *Nature Education* 1.1 (2008), p. 160.

[LH08] Peter Langfelder and Steve Horvath. "WGCNA: an R package for weighted correlation network analysis." eng. In: *BMC bioinformatics* 9 (Dec. 2008), p. 559.

[Bin+09] Gabriela Bindea, Bernhard Mlecnik, Hubert Hackl, Pornpimol Charoentong, Marie Tosolini, Amos Kirilovsky, Wolf-Herman Fridman, Franck Pagès, Zlatko Trajanoski, and Jérôme Galon. "ClueGO: a Cytoscape plug-in to decipher functionally grouped gene ontology and pathway annotation networks". In: *Bioinformatics* 25.8 (2009), pp. 1091–1093.

[CWZ09] Luonan Chen, Rui-Sheng Wang, and Xiang-Sun Zhang. *Biomolecular networks: methods and applications in systems biology*. John Wiley & Sons, 2009.

[DBB09] Lori Dalton, Virginia Ballarin, and Marcel Brun. "Clustering algorithms: on learning, validation, performance, and applications to genomics". In: *Current genomics* 10.6 (2009), pp. 430–445.

[Haa+09] Herman H. H. B. M. van Haagen et al. "Novel protein-protein interactions inferred from literature context." eng. In: *PloS one* 4 (11 Nov. 2009), e7894.

[Iri+09] Rafael A Irizarry, Chi Wang, Yun Zhou, and Terence P Speed. "Gene set enrichment analysis made simple". In: *Statistical methods in medical research* 18.6 (2009), pp. 565–575.

[Kar09] Gerald Karp. *Cell and molecular biology: concepts and experiments*. John Wiley & Sons, 2009.

[Sch+09] Ariel S Schwartz, Jingkai Yu, Kyle R Gardenour, Russell L Finley Jr, and Trey Ideker. "Cost-effective strategies for completing the interactome". In: *Nature methods* 6.1 (2009), pp. 55–61.

[AW10] Hervé Abdi and Lynne J Williams. "Principal component analysis". In: *Wiley interdisciplinary reviews: computational statistics* 2.4 (2010), pp. 433–459.

[SW10] Roy D Sleator and Paul Walsh. "An overview of in silico protein function prediction". In: *Archives of microbiology* 192.3 (2010), pp. 151–155.

[Vin+10] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. "Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion". In: *J. Mach. Learn. Res.* 11 (Dec. 2010), pp. 3371–3408. ISSN: 1532-4435.

[Cas+11] Laia Castells-Roca, José Garcıa-Martınez, Joaquın Moreno, Enrique Herrero, Gemma Bellı, and José E Pérez-Ortın. "Heat shock response in yeast involves changes in both transcription rates and mRNA stabilities". In: *PLOS One* 6.2 (2011), e17272.

[CL11] Chih-Chung Chang and Chih-Jen Lin. "LIBSVM: a library for support vector machines". In: *ACM transactions on intelligent systems and technology (TIST)* 2.3 (2011), pp. 1–27.

[Har+11] Mehrtash T Harandi, Conrad Sanderson, Sareh Shirazi, and Brian C Lovell. "Graph embedding discriminant analysis on Grassmannian manifolds for improved image set matching". In: *CVPR 2011*. IEEE. 2011, pp. 2705–2712.

[Leg+11] Pierre Legrain, Ruedi Aebersold, Alexander Archakov, Amos Bairoch, Kumar Bala, Laura Beretta, John Bergeron, Christoph H Borchers, Garry L Corthals, Catherine E Costello, et al. "The human proteome project: current state and future direction". In: *Molecular & cellular proteomics* 10.7 (2011).

[Ped+11] F Pedregosa, G Varoquaux, A Gramfort, V Michel, B Thirion, O Grisel, and J Vanderplas. "scikit-learn: Machine learning in Python". In: *Journal of Machine Learning Research* 12 (2011).

[Bar+12] Tanya Barrett et al. "NCBI GEO: archive for functional genomics data sets—update". In: *Nucleic acids research* 41.D1 (2012), pp. D991–D995.

[Dal+12] Mark R Dalman, Anthony Deeter, Gayathri Nimishakavi, and Zhong-Hui Duan. "Fold change and p-value cutoffs significantly alter microarray interpretations". In: *BMC bioinformatics*. Vol. 13. SUppl. 2, S11. 2012, pp. 1–4.

[Don+12] Xiaowen Dong, Pascal Frossard, Pierre Vandergheynst, and Nikolai Nefedov. "Clustering with multi-layer graphs: A spectral perspective". In: *IEEE Transactions on Signal Processing* 60.11 (2012), pp. 5820–5831.

[MGM12] Kevin A Morano, Chris M Grant, and W Scott Moye-Rowley. "The response to heat shock and oxidative stress in *Saccharomyces cerevisiae*". In: *Genetics* 190.4 (2012), pp. 1157–1195.

[Ver+12] Jacob Verghese, Jennifer Abrams, Yanyu Wang, and Kevin A Morano. "Biology of the heat shock response and protein chaperones: budding yeast (*Saccharomyces cerevisiae*) as a model system". In: *Microbiology and Molecular Biology Reviews* 76.2 (2012), pp. 115–158.

[Ahm+13] Amr Ahmed, Nino Shervashidze, Shravan Narayanamurthy, Vanja Josifovski, and Alexander J Smola. "Distributed large-scale natural graph factorization". In: *Proceedings of the 22nd international conference on World Wide Web*. 2013, pp. 37–48.

[Coz+13] Domenico Cozzetto, Daniel WA Buchan, Kevin Bryson, and David T Jones. "Protein function prediction by massive integration of evolutionary analyses and multiple data sources". In: *BMC bioinformatics*. Vol. 14. 3. Springer. 2013, pp. 1–11.

[HP13] Saad Haider and Ranadip Pal. "Integrated analysis of transcriptomic and proteomic data". In: *Current genomics* 14.2 (2013), pp. 91–110.

[Ize13] Alan Julian Izenman. "Linear discriminant analysis". In: *Modern multivariate statistical techniques*. Springer, 2013, pp. 237–280.

[KW13] Diederik P Kingma and Max Welling. "Auto-encoding variational bayes". In: *arXiv preprint arXiv:1312.6114* (2013).

[Kub13] Christian P Kubicek. "Systems biological approaches towards understanding cellulase production by Trichoderma reesei". In: *Journal of biotechnology* 163.2 (2013), pp. 133–142.

[Mik+13] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. "Distributed Representations of Words and Phrases and Their Compositionality". In: *NIPS*. 2013, pp. 3111–3119.

[ODS13] Aisling O'Driscoll, Jurate Daugelaite, and Roy D Sleator. "'Big data', Hadoop and cloud computing in genomics". In: *Journal of biomedical informatics* 46.5 (2013), pp. 774–781.

[Yu+13] Donghyeon Yu, MinSoo Kim, Guanghua Xiao, and Tae Hyun Hwang. "Review of biological network data and its applications". In: *Genomics & informatics* 11.4 (2013), p. 200.

[EDH14] Frank Emmert-Streib, Matthias Dehmer, and Benjamin Haibe-Kains. "Gene regulatory networks and their applications: understanding biological and medical problems in terms of networks". In: *Frontiers in cell and developmental biology* 2 (2014), p. 38.

[Eng+14] Stacia R Engel, Fred S Dietrich, Dianna G Fisk, Gail Binkley, Rama Balakrishnan, Maria C Costanzo, Selina S Dwight, Benjamin C Hitz, Kalpana Karra, Robert S Nash, et al. "The reference genome sequence of Saccharomyces cerevisiae: then and now". In: *G3: Genes, Genomes, Genetics* 4.3 (2014), pp. 389–398.

[LG14] Omer Levy and Yoav Goldberg. "Neural word embedding as implicit matrix factorization". In: *Advances in neural information processing systems* 27 (2014), pp. 2177–2185.

[Nes14] Alexey I Nesvizhskii. "Proteogenomics: concepts, applications and computational strategies". In: *Nature methods* 11.11 (2014), pp. 1114–1125.

[PAS14] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. "Deepwalk: Online learning of social representations". In: *Proc. 20th ACM SIGKDD Int. Conf. on knowledge discovery and data mining*. 2014, pp. 701–710.

[Saf+14] Nahid Safari-Alighiarloo, Mohammad Taghizadeh, Mostafa Rezaei-Tavirani, Bahram Goliaei, and Ali Asghar Peyvandi. "Protein-protein interaction networks (PPI) and complex diseases". In: *Gastroenterology and Hepatology from bed to bench* 7.1 (2014), p. 17.

[Wan+14] Bo Wang, Aziz M Mezlini, Feyyaz Demir, Marc Fiume, Zhuowen Tu, Michael Brudno, Benjamin Haibe-Kains, and Anna Goldenberg. "Similarity network fusion for aggregating data types on a genomic scale". In: *Nature methods* 11.3 (2014), p. 333.

[Alb+15] Bruce Alberts et al. "Molecular biology of the cell Sixth edition Ch. 14, 755–756". In: *Garland Science, Taylor and Francis Group* (2015).

[CLX15] Shaosheng Cao, Wei Lu, and Qiongkai Xu. "Grarep: Learning graph representations with global structural information". In: *Proceedings of the 24th ACM international on conference on information and knowledge management*. 2015, pp. 891–900.

[FK15] Peter Flach and Meelis Kull. "Precision-recall-gain curves: PR analysis done right". In: *Advances in neural information processing systems* 28 (2015).

[GP15] Vladimir Gligorijević and Nataša Pržulj. "Methods for biological data integration: perspectives and challenges". In: *Journal of the Royal Society Interface* 12.112 (2015), p. 20150571.

[HS15] Mohammad Hossin and Md Nasir Sulaiman. "A review on evaluation metrics for data classification evaluations". In: *International journal of data mining & knowledge management process* 5.2 (2015), p. 1.

[Pir+15a] Aurélie Pirayre, Camille Couprie, Frédérique Bidard, Laurent Duval, and Jean-Christophe Pesquet. "BRANE Cut: biologically-related a priori network enhancement with graph cuts for gene regulatory network inference". In: *BMC Bioinformatics* 16.1 (2015), pp. 1–12.

[Pir+15b] Aurélie Pirayre, Camille Couprie, Laurent Duval, and Jean-Christophe Pesquet. "Fast convex optimization for connectivity enforcement in gene regulatory network inference". In: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2015, pp. 1002–1006.

[CBL16] Theodosia Charitou, Kenneth Bryan, and David J Lynn. "Using biological networks to integrate, visualize and analyze genomics data". In: *Genetics Selection Evolution* 48.1 (2016), pp. 1–12.

[CBP16] Hyunghoon Cho, Bonnie Berger, and Jian Peng. "Compact integration of multi-network topology for functional analysis of genes". In: *Cell systems* 3.6 (2016), pp. 540–548.

[GL16] Aditya Grover and Jure Leskovec. "node2vec: Scalable feature learning for networks". In: *Proc. 22nd ACM SIGKDD Int. Conf. on knowledge discovery and data mining*. 2016, pp. 855–864.

[Guo+16] Quantong Guo, Emanuele Cozzo, Zhiming Zheng, and Yamir Moreno. "Levy random walks on multiplex networks". In: *Scientific reports* 6.1 (2016), pp. 1–11.

[Jud+16] Gaëlle Judes, Khaldoun Rifaı, Marine Daures, Lucas Dubois, Yves-Jean Bignon, Frédérique Penault-Llorca, and Dominique Bernard-Gallon. "High-throughput «Omics» technologies: New tools for the study of triple-negative breast cancer". In: *Cancer letters* 382.1 (2016), pp. 77–85.

[KW16] Thomas N Kipf and Max Welling. "Semi-supervised classification with graph convolutional networks". In: *arXiv preprint arXiv:1609.02907* (2016).

[Kul+16] 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". In: *Nucleic acids research* 44.W1 (2016), W90–W97.

[Ou+16] 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*. 2016, pp. 1105–1114.

[Oye+16] Jelili Oyelade, Itunuoluwa Isewon, Funke Oladipupo, Olufemi Aromolaran, Efosa Uwoghiren, Faridah Ameh, Moses Achas, and Ezekiel Adebiyi. "Clustering algorithms: their application to gene expression data". In: *Bioinformatics and Biology insights* 10 (2016), BBI–S38316.

[Rud+16] Maja Rudolph, Francisco Ruiz, Stephan Mandt, and David Blei. "Exponential Family Embeddings". In: *NIPS*. Curran Associates Inc., 2016, pp. 478–486.

[Sil16] Roberto N Silva. "Perspectives in Genomics The Future of Fungi in 'omics, era". In: *Current genomics* 17.2 (2016), p. 82.

[Sol+16] Albert Solé-Ribalta, Manlio De Domenico, Sergio Gómez, and Alex Arenas. "Random walk centrality in interconnected multilayer networks". In: *Physica D: Nonlinear Phenomena* 323 (2016), pp. 73–79.

[WCZ16] 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*. 2016, pp. 1225–1234.

[HYL17a] Will Hamilton, Zhitao Ying, and Jure Leskovec. "Inductive representation learning on large graphs". In: *Advances in neural information processing systems* 30 (2017).

[HYL17b] William L. Hamilton, Rex Ying, and Jure Leskovec. "Representation Learning on Graphs: Methods and Applications". In: *IEEE Data Eng. Bull.* 40.3 (2017), pp. 52–74.

[KGK17] Jocelyn E Krebs, Elliott S Goldstein, and Stephen T Kilpatrick. *Lewin's genes XII*. Jones & Bartlett Learning, 2017.

[Liu+17] Weiyi Liu, Pin-Yu Chen, Sailung Yeung, Toyotaro Suzumura, and Lingli Chen. "Principled multilayer network embedding". In: *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE. 2017, pp. 134–141.

[Per+17] Yasset Perez-Riverol, Max Kuhn, Juan Antonio Vizcaıno, Marc-Phillip Hitz, and Enrique Audain. "Accurate and fast feature selection workflow for high-dimensional omics data". In: *PloS one* 12.12 (2017), e0189875.

[Pir+17] Aurélie Pirayre, Camille Couprie, Laurent Duval, and Jean-Christophe Pesquet. "BRANE Clust: Cluster-assisted gene regulatory network inference refinement". In: *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 15.3 (2017), pp. 850–860.

[Vel+17] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. *Graph Attention Networks*. 2017.

[ZL17] Marinka Zitnik and Jure Leskovec. "Predicting multicellular function through multi-layer tissue networks". In: *Bioinformatics* 33.14 (2017), pp. i190–i198.

[BK18] Arunkumar Bagavathi and Siddharth Krishnan. "Multi-net: a scalable multiplex network embedding framework". In: *International conference on complex networks and their applications*. Springer. 2018, pp. 119–131.

[GBB18] Vladimir Gligorijevic, Meet Barot, and Richard Bonneau. "deepNF: deep network fusion for protein function prediction". eng. In: *Bioinformatics (Oxford, England)* 34.29868758 (Nov. 2018), pp. 3873–3881. ISSN: 1367-4803.

[LWN18] Yifeng Li, Fang-Xiang Wu, and Alioune Ngom. "A review on machine learning principles for multi-view biological data integration". In: *Briefings in bioinformatics* 19.2 (2018), pp. 325–340.

[Man+18] Claudia Manzoni, Demis A Kia, Jana Vandrovcova, John Hardy, Nicholas W Wood, Patrick A Lewis, and Raffaele Ferrari. "Genome, transcriptome and proteome: the rise of omics data and their integration in biomedical sciences". In: *Briefings in bioinformatics* 19.2 (2018), pp. 286–302.

[NM18] Duong Nguyen and Fragkiskos D Malliaros. "BiasedWalk: Biased sampling for representation learning on graphs". In: *IEEE Int. Conf. on Big Data (Big Data)*. 2018, pp. 4045–4053.

[Ngu+18] Nga Thi Thuy Nguyen, Bruno Contreras-Moreira, Jaime A Castro-Mondragon, Walter Santana-Garcia, Raul Ossio, Carla Daniela Robles-Espinoza, Mathieu Bahin, Samuel Collombet, Pierre Vincens, Denis Thieffry, et al. "RSAT 2018: regulatory sequence analysis tools 20th anniversary". In: *Nucleic acids research* 46.W1 (2018), W209–W214.

[Qiu+18] Jiezhong Qiu, Yuxiao Dong, Hao Ma, Jian Li, Kuansan Wang, and Jie Tang. "Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec". In: *Proceedings of the eleventh ACM international conference on web search and data mining*. 2018, pp. 459–467.

[RS18] Nimrod Rappoport and Ron Shamir. "Multi-omic and multi-view clustering algorithms: review and cancer benchmark". In: *Nucleic acids research* 46.20 (2018), pp. 10546–10562.

[Sae+18] Nasir Saeed, Haewoon Nam, Mian Imtiaz Ul Haq, and Dost Bhatti Muhammad Saqib. "A survey on multidimensional scaling". In: *ACM Computing Surveys (CSUR)* 51.3 (2018), pp. 1–25.

[Tei+18] Miguel C Teixeira, Pedro T Monteiro, Margarida Palma, Catarina Costa, Cláudia P Godinho, Pedro Pais, Mafalda Cavalheiro, Miguel Antunes, Alexandre Lemos, Tiago Pedreira, et al. "YEASTRACT: an upgraded database for the analysis of transcription regulatory networks in *Saccharomyces cerevisiae*". In: *Nucleic acids research* 46.D1 (2018), pp. D348–D353.

[Tsi+18] Anton Tsitsulin, Davide Mottin, Panagiotis Karras, and Emmanuel Müller. "Verse: Versatile graph embeddings from similarity measures". In: *Proceedings of the 2018 world wide web conference*. 2018, pp. 539–548.

[Van+18a] Sipko Van Dam, Urmo Vosa, Adriaan van der Graaf, Lude Franke, and Joao Pedro de Magalhaes. "Gene co-expression analysis for functional classification and gene–disease predictions". In: *Briefings in bioinformatics* 19.4 (2018), pp. 575–592.

[Van+18b] Monique GP Van Der Wijst, Dylan H de Vries, Harm Brugge, Harm-Jan Westra, and Lude Franke. "An integrative approach for building personalized gene regulatory networks for precision medicine". In: *Genome medicine* 10.1 (2018), pp. 1–15.

[Wil+18] James D Wilson, Melanie Baybay, Rishi Sankar, and Paul Stillman. "Fast embedding of multilayer networks: An algorithm and application to group fmri". In: *arXiv preprint arXiv:1809.06437* 96 (2018).

[Yan+18] 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". In: *Briefings in bioinformatics* 19.6 (2018), pp. 1370–1381.

[Zha+18] Hongming Zhang, Liwei Qiu, Lingling Yi, and Yangqiu Song. "Scalable Multiplex Network Embedding." In: *IJCAI*. Vol. 18. 2018, pp. 3082–3088.

[Du+19] Jingcheng Du, Peilin Jia, Yulin Dai, Cui Tao, Zhongming Zhao, and Degui Zhi. "Gene2vec: distributed representation of genes based on co-expression". In: *BMC Genomics* 20.1 (2019), pp. 7–15.

[Kar+19] Peter D Karp, Richard Billington, Ron Caspi, Carol A Fulcher, Mario Latendresse, Anamika Kothari, Ingrid M Keseler, Markus Krummenacker, Peter E Midford, Quang Ong, et al. "The BioCyc collection of microbial genomes and metabolic pathways". In: *Briefings in bioinformatics* 20.4 (2019), pp. 1085–1093.

[Nie19] Jens Nielsen. "Yeast systems biology: model organism and cell factory". In: *Biotechnology journal* 14.9 (2019), p. 1800421.

[Pržl9] Nataša Pržulj. *Analyzing Network Data in Biology and Medicine: An Interdisciplinary Textbook for Biological, Medical and Computational Scientists*. Cambridge University Press, 2019.

[Tin+19] Giulia Tini, Luca Marchetti, Corrado Priami, and Marie-Pier Scott-Boyer. "Multi-omics integration—a comparison of unsupervised clustering methodologies". In: *Briefings in bioinformatics* 20.4 (2019), pp. 1269–1279.

[Zit+19] Marinka Zitnik, Francis Nguyen, Bo Wang, Jure Leskovec, Anna Goldenberg, and Michael M Hoffman. "Machine learning for integrating data in biology and medicine: Principles, practice, and opportunities". In: *Information Fusion* 50 (2019), pp. 71–91.

[Bis20] James Bisgard. *Analysis and Linear Algebra: The Singular Value Decomposition and Applications*. 1st. Student Mathematical Library. Providence, RI, USA: American Mathematical Society, 2020, p. 217. ISBN: 1470463326.

[CM20a] Abdulkadir Celikkanat and Fragkiskos D Malliaros. "Exponential family graph embeddings". In: *Proc. AAAI Conference on Artificial Intelligence*. Vol. 34. 04. 2020, pp. 3357–3364.

[ÇM20] Abdulkadir Çelikkanat and Fragkiskos D. Malliaros. "Exponential Family Graph Embeddings". In: *AAAI*. 2020.

[CM20b] Sudhanshu Chanpuriya and Cameron Musco. "Infinitewalk: Deep network embeddings as Laplacian embeddings with a nonlinearity". In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2020, pp. 1325–1333.

[Cha+20] Cécile Chauvel, Alexei Novoloaca, Pierre Veyre, Frédéric Reynier, and Jérémie Becker. "Evaluation of integrative clustering methods for the analysis of multi-omics data". In: *Briefings in Bioinformatics* 21.2 (2020), pp. 541–552.

[Che+20] Fenxiao Chen, Yun-Cheng Wang, Bin Wang, and C-C Jay Kuo. "Graph representation learning: a survey". In: *APSIPA Transactions on Signal and Information Processing* 9 (2020).

[CJ20] Davide Chicco and Giuseppe Jurman. "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation". In: *BMC genomics* 21.1 (2020), pp. 1–13.

[Di +20] Noemi Di Nanni, Matteo Bersanelli, Luciano Milanesi, and Ettore Mosca. "Network diffusion promotes the integrative analysis of multiple omics". In: *Frontiers in genetics* 11 (2020), p. 106.

[FGZ20] Kunjie Fan, Yuanfang Guan, and Yan Zhang. "Graph2GO: a multimodal attributed network embedding method for inferring protein functions". In: *GigaScience* 9.8 (2020), giaa081.

[For+20] Oriol Fornes, Jaime A Castro-Mondragon, Aziz Khan, Robin Van der Lee, Xi Zhang, Phillip A Richmond, Bhavi P Modi, Solenne Correard, Marius Gheorghe, Damir Baranašić, et al. "JASPAR 2020: update of the open-access database of transcription factor binding profiles". In: *Nucleic acids research* 48.D1 (2020), pp. D87–D92.

[Ham20] William L Hamilton. "Graph representation learning". In: *Synthesis Lectures on Artifical Intelligence and Machine Learning* 14.3 (2020), pp. 1–159.

[Lee+20] Bohyun Lee, Shuo Zhang, Aleksandar Poleksic, and Lei Xie. "Heterogeneous multi-layered network model for omics data integration and analysis". In: *Frontiers in genetics* 10 (2020), p. 1381.

[Liu+20] Zhijun Liu, Chao Huang, Yanwei Yu, Baode Fan, and Junyu Dong. "Fast Attributed Multiplex Heterogeneous Network Embedding". In: *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2020, pp. 995–1004.

[Mon+20] Pedro T Monteiro, Jorge Oliveira, Pedro Pais, Miguel Antunes, Margarida Palma, Mafalda Cavalheiro, Mónica Galocha, Cláudia P Godinho, Luıs C Martins, Nuno Bourbon, et al. "YEASTRACT+: a portal for cross-species comparative genomics of transcription regulation in yeasts". In: *Nucleic acids research* 48.D1 (2020), pp. D642–D649.

[Nuñ+20] Carme Nuño-Cabanes, Manuel Ugidos, Sonia Tarazona, Manuel Martın-Expósito, Alberto Ferrer, Susana Rodrıguez-Navarro, and Ana Conesa. "A multi-omics dataset of heat-shock response in the yeast RNA binding protein Mip6". In: *Scientific data* 7.69 (2020).

[Sub+20] Indhupriya Subramanian, Srikant Verma, Shiva Kumar, Abhay Jere, and Krishanpal Anamika. "Multi-omics Data Integration, Interpretation, and Its Application". eng. In: *Bioinformatics and biology insights* 14.32076369 (Jan. 2020), pp. 1177932219899051–1177932219899051. ISSN: 1177-9322.

[Szk+20] Damian Szklarczyk et al. "The STRING database in 2021: customizable protein–protein networks, and functional characterization of user-uploaded gene/measurement sets". In: *Nucleic Acids Research* 49.D1 (Nov. 2020), pp. D605–D612.

[Yue+20] Xiang Yue, Zhen Wang, Jingong Huang, Srinivasan Parthasarathy, Soheil Moosavinasab, Yungui Huang, Simon M Lin, Wen Zhang, Ping Zhang, and Huan Sun. "Graph embedding on biomedical networks: methods, applications and evaluations". In: *Bioinformatics* 36.4 (2020), pp. 1241–1251.

[ZLX20] Guangyan Zhou, Shuzhao Li, and Jianguo Xia. "Network-Based Approaches for Multi-omics Integration." eng. In: *Methods in molecular biology (Clifton, N.J.)* 2104 (2020), pp. 469–487.

[Zho+20] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. "Graph neural networks: A review of methods and applications". In: *AI Open* 1 (2020), pp. 57–81.

[Can+21] Laura Cantini, Pooya Zakeri, Celine Hernandez, Aurelien Naldi, Denis Thieffry, Elisabeth Remy, and Anaıs Baudot. "Benchmarking joint multi-omics dimensionality reduction approaches for the study of cancer". In: *Nature communications* 12.1 (2021), pp. 1–12.

[Dua+21] Ran Duan, Lin Gao, Yong Gao, Yuxuan Hu, Han Xu, Mingfeng Huang, Kuo Song, Hongda Wang, Yongqiang Dong, Chaoqun Jiang, et al. "Evaluation and comparison of multi-omics data integration methods for cancer subtyping". In: *PLoS computational biology* 17.8 (2021), e1009224.

[HSS21] Zexi Huang, Arlei Silva, and Ambuj Singh. "A broader picture of random-walk based graph embedding". In: *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*. 2021, pp. 685–695.

[Jag+21a] Surabhi Jagtap, Abdulkadir Celikkanat, Aurélie Pirayre, Frederique Bidard, Laurent Duval, and Fragkiskos D Malliaros. "Multiomics Data Integration for Gene Regulatory Network Inference with Exponential Family Embeddings". In: 2021.

[Jag+21b] Surabhi Jagtap, Aurélie Pirayre, Frederique Bidard, Laurent Duval, and Fragkiskos D. Malliaros. "BRANet: Graph-based Integration of Multi-omics Data with Biological a priori for Regulatory Network Inference". In: *17th International Conference on Computational Intelligence Methods for Bioinformatics and Biostatistics (CIBB)*. Online, Italy, Nov. 2021.

[Kso+21] Najla Ksouri, Jaime A. Castro-Mondragón, Francesc Montardit-Tardà, Jacques van Helden, Bruno Contreras-Moreira, and Yolanda Gogorcena. "Motif analysis in co-expression networks reveals regulatory elements in plants: The peach as a model". In: *Plant Phyiology* (2021).

[Oug+21] Rose Oughtred, Jennifer Rust, Christie Chang, Bobby-Joe Breitkreutz, Chris Stark, Andrew Willems, Lorrie Boucher, Genie Leung, Nadine Kolas, Frederick Zhang, et al. "The BioGRID database: A comprehensive biomedical resource of curated protein, genetic, and chemical interactions". In: *Protein Science* 30.1 (2021), pp. 187–200.

[Pen+21] Jiajie Peng, Hansheng Xue, Zhongyu Wei, Idil Tuncali, Jianye Hao, and Xuequn Shang. "Integrating multi-network topology for gene function prediction using deep neural networks". In: *Briefings in bioinformatics* 22.2 (2021), pp. 2096–2105.

[Pic+21] Milan Picard, Marie-Pier Scott-Boyer, Antoine Bodein, Olivier Périn, and Arnaud Droit. "Integration strategies of multi-omics data for machine learning analysis". In: *Computational and Structural Biotechnology Journal* 19 (2021), pp. 3735–3746.

[Pio+21] Léo Pio-Lopez, Alberto Valdeolivas, Laurent Tichit, Élisabeth Remy, and Anaïs Baudot. "MultiVERSE: a multiplex and multiplex-heterogeneous network embedding approach". In: *Scientific Reports* 11.1 (Apr. 2021), p. 8794. ISSN: 2045-2322.

[Gon+22] Agustin Gonzalez-Reymundez, Alexander Grueneberg, Guanqi Lu, Filipe Couto Alves, Gonzalo Rincon, and Ana I Vazquez. "MOSS: multi-omic integration with sparse value decomposition". In: *Bioinformatics* 38.10 (2022), pp. 2956–2958.

[Hoc+22] Rémi Hocq, Surabhi Jagtap, Magali Boutard, Andrew C. Tolonen, Laurent Duval, Aurélie Pirayre, Nicolas Lopes Ferreira, and François Wasels. "Genome-Wide TSS Distribution in Three Related Clostridia with Normalized Capp-Switch Sequencing". In: *Microbiology Spectrum* 10.2 (2022), e02288–21.

[Jag+22a] Surabhi Jagtap, Abdulkadir Çelikkanat, Aurélie Pirayre, Frédérique Bidard, Laurent Duval, and Fragkiskos D Malliaros. "BraneMF: Integration of Biological Networks for Functional Analysis of Proteins". In: *Bioinformatics* (Nov. 2022). ISSN: 1367-4803.

[Jag+22b] Surabhi Jagtap, Aurélie Pirayre, Frédérique Bidard, Laurent Duval, and Fragkiskos D. Malliaros. "BRANEnet: embedding multilayer networks for omics data integration". In: *BMC Bioinformatics* 23.1 (Oct. 2022), p. 429. ISSN: 1471-2105.

[Lov+22] Marta Lovino, Vincenzo Randazzo, Gabriele Ciravegna, Pietro Barbiero, Elisa Ficarra, and Giansalvo Cirrincione. "A survey on data integration for multi-omics sample clustering". In: *Neurocomputing* 488 (2022), pp. 494–508.