BraneMF : integration of biological networks for functional analysis of proteins

Surabhi Jagtap, Abdulkadir Çelikkanat, Aurélie Pirayre, Frédérique Bidard, Laurent Duval, Fragkiskos D. Malliaros

Bioinformatics, 38(24), 2022, 5383–5389
https://doi.org/10.1093/bioinformatics/btac691
Advance Access Publication Date: 2 November 2022
Original Paper

Systems biology

BraneMF: integration of biological networks for functional analysis of proteins

Surabhi Jagtap [1,2], Abdulkadir Çelikkanat[3], Aurélie Pirayre[1], Frédérique Bidard[1], Laurent Duval[1] and Fragkiskos D. Malliaros[2,*]

[1]IFP Energies Nouvelles, 92852 Rueil-Malmaison, France, [2]Université Paris-Saclay, CentraleSupélec, Inria, Centre for Visual Computing, 91190 Gif-Sur-Yvette, France and [3]Technical University of Denmark, DTU Compute, 2800 Kongens Lyngby, Denmark

*To whom correspondence should be addressed.

Associate Editor: Inanc Birol

Received on January 23, 2022; revised on October 5, 2022; editorial decision on October 10, 2022; accepted on November 1, 2022

Abstract

Motivation: The cellular system of a living organism is composed of interacting bio-molecules that control cellular processes at multiple levels. Their correspondences are represented by tightly regulated molecular networks. The increase of omics technologies has favored the generation of large-scale disparate data and the consequent demand for simultaneously using molecular and functional interaction networks: gene co-expression, protein–protein interaction (PPI), genetic interaction and metabolic networks. They are rich sources of information at different molecular levels, and their effective integration is essential to understand cell functioning and their building blocks (proteins). Therefore, it is necessary to obtain informative representations of proteins and their proximity, that are not fully captured by features extracted directly from a single informational level. We propose BraneMF, a novel random walk-based matrix factorization method for learning node representation in a multilayer network, with application to omics data integration.

Results: We test BraneMF with PPI networks of *Saccharomyces cerevisiae*, a well-studied yeast model organism. We demonstrate the applicability of the learned features for essential multi-omics inference tasks: clustering, function and PPI prediction. We compare it to the state-of-the-art integration methods for multilayer networks. BraneMF outperforms baseline methods by achieving high prediction scores for a variety of downstream tasks. The robustness of results is assessed by an extensive parameter sensitivity analysis.

Availability and implementation: BraneMF's code is freely available at: https://github.com/Surabhivj/BraneMF, along with datasets, embeddings and result files.

Contact: fragkiskos.malliaros@centralesupelec.fr

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Comprehensive interpretation of biological mechanisms in an organism requires understanding bio-molecular interactions represented by tightly regulated molecular networks (Subramanian *et al.*, 2020). The advent of high-throughput technologies has introduced enormous disparate omics data in the scenario, and, in parallel, promising avenues are paved for their analysis and interpretation (Yue *et al.*, 2020). Despite the explosion of omics data, functional annotations are yet to be unveiled, for at least 20% of proteins, even in model organisms (Wood *et al.*, 2019). Hence, novel methods to analyze and obtain significant knowledge from heterogeneous data are necessary. Analysis of single omics is limited to correlations, mostly reflecting reactive processes rather than causative ones. Several studies have shown the importance of multi-omics integration over single omics analysis (Subramanian *et al.*, 2020). Such approach can provide insights on the interconnectedness of different bio-molecules (proteins, RNAs, metabolites) and the flow of biological information occurring among them. This will enable us to understand biological mechanisms, for instance, gene regulation (Hu *et al.*, 2020; Jagtap *et al.*, 2021), protein function prediction (PFP) (Cozzetto *et al.*, 2013) or drug–target identification (Luo *et al.*, 2017).

In the past years, network approaches have offered potential for integrative omics analysis, facilitating a new era of systems biology (Di Nanni *et al.*, 2020; Subramanian *et al.*, 2020; Yan *et al.*, 2018). 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. Multilayer

network integration strategies can be classified as early, intermediate or late integration (Gligorijević and Pržulj, 2015; Li *et al.*, 2018). 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 a great value in developing efficient intermediate-level integration approaches (Zhou *et al.*, 2020a), capable of handling heterogeneous data, providing insights into the functional categories of proteins (e.g. representation of system-level inter-relationships within bio-molecules).

In this article, we are inspired by Graph Representation Learning (GRL) algorithms to encode graph structure into compact embedding vectors (Hamilton *et al.*, 2017). Our motivation is further extended toward leveraging closed forms of GRL methods that perform implicit matrix factorization, favoring intrinsic connection and interpretability of graph topology (Levy and Goldberg, 2014; Qiu *et al.*, 2018). We propose BraneMF, a novel intermediate-level network integration framework by effectively combining protein–protein interaction (PPI) networks of heterogeneous data sources. We formally define the task as a multilayer network embedding problem. Given a set of networks, we aim to learn low-dimensional latent node embeddings so that the structure of input networks is properly integrated and preserved in the new space.

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

2 Related work

A plethora of GRL approaches are based on random walks (Çelikkanat and Malliaros, 2020; Grover and Leskovec, 2016; Nguyen and Malliaros, 2018; Perozzi *et al.*, 2014), matrix factorization (Levy and Goldberg, 2014; Qiu *et al.*, 2018) or neural networks (Hamilton *et al.*, 2017). However, they have mostly been introduced for single-layer networks (Yue *et al.*, 2020). Inspired by Word2Vec-based single-layer network embedding techniques (Grover and Leskovec, 2016; Mikolov *et al.*, 2013; Perozzi *et al.*, 2014), a few GRL methods have been proposed for multilayer networks. Principled Multilayer Network Embedding (PMNE) (Liu *et al.*, 2017) is an extension of a single-layer graph embedding to a multilayer network. Multiplex Network Embedding (MNE) (Zhang *et al.*, 2018) is a multi-layer network embedding method that generates random walks for each layer and then applies the Skip-Gram model (Mikolov *et al.*, 2013) to learn joint embeddings for each node. The final node embeddings are composed of three parts: common embeddings, relation-based embeddings and a transformation matrix. Multi-Net (Bagavathi and Krishnan, 2018) uses random walks, namely classical, diffusive and physical to obtain node sequences (Guo *et al.*, 2016; Solé-Ribalta *et al.*, 2016). Then, it merges the nodes' neighborhoods (context for each node) and learns a d-dimensional feature vector for each node by maximizing the likelihood of occurrence of node neighbors across all layers. Multi-node2vec (Multi-n2v) (Wilson, 2020) extends Node2Vec (Grover and Leskovec, 2016) to multi-layer networks. The model collects a bag of words from each layer by performing 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-Lopez *et al.*, 2021) computes a similarity matrix using random walks with restart (RWR). Then, it applies an optimized version of VERSE (Tsitsulin *et al.*, 2018), a vertex-to-vertex similarity-oriented embedding method to compute the representations. FAME (Liu *et al.*, 2020) decomposes the heterogeneous multiplex network into homogeneous and bipartite sub-networks. It follows the use of a spectral transformation module to automatically aggregate and decouple sub-networks with the exploration of their multi-relational topological signals. Lastly, BraneExp (Jagtap *et al.*, 2021) is based on random walks that are used to define node similarity, using expressive conditional probability functions. It is a network integration framework with the concept of exponential family graph embeddings, which generalizes multilayer random walk-based GRL methods to an instance of exponential family distribution (Çelikkanat and Malliaros, 2020; Rudolph *et al.*, 2016).

For biological networks, similarity network fusion (SNF) (Wang *et al.*, 2014) constructs a similarity network for each data type and then iteratively integrates these networks using a network fusion methodology. Mashup (Cho *et al.*, 2016) 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. deepNF (Gligorijević *et al.*, 2018) is a network fusion method relying on multimodal deep autoencoders. It also takes a collection of PPI networks as input and makes use of an autoencoder (AE). OhmNet (Zitnik and Leskovec, 2017) is an unsupervised feature learning approach for multilayer networks with a predefined hierarchy describing relationships between the layers. To capture the structural properties of networks, deepNF and Mashup are based on RWR. The vectors learned from both methods are then fed into a support vector machine (SVM) classifier to predict functional classes of proteins. deepMNE-CNN (Peng *et al.*, 2021) has been introduced as a multi-network embedding approach which applies a semi-autoencoder based model to learn protein features. Graph2GO (Fan *et al.*, 2020) extends variational graph autoencoders to multilayer networks. It establishes to integrate networks derived from heterogeneous information, including sequence similarity, PPI and protein features, amino acid sequence, sub-cellular location and protein domains.

Most of these network integration frameworks are engrossed toward Gene Ontology (GO) prediction. 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. To do so, we obtain random walk-based Positive Pointwise Mutual Information (PPMI) matrices from the set of networks to capture node neighborhood information. These matrices are used as input for learning embeddings using a joint singular value decomposition (SVD) framework. Moreover, we demonstrate the adequacy of the learned embeddings to solve important downstream machine learning tasks such as protein clustering with their functional enrichment, prediction of protein functions and PPIs.

3 Materials and methods

BraneMF is an integration framework to learn protein features from multiple PPI networks. A schematic representation is given in Figure 1. Firstly, we compute a random walk-based multi-layered 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. In addition, we compare the performance of BraneMF to state-of-the-art baseline methods.

3.1 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 $\mathcal{G}$, 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 used as a similarity measure for a variety of problems in graph theory. More precisely, from some single-layer graph $\mathcal{G}_l$ and a starting node v_i, a random walk performs transitions by selecting a neighborhood node at random at each step. The random walk generation procedure continues for all nodes and for a predefined number of walks of a given length. In this way, node sequences are obtained that are further used to learn node features. This can be achieved by maximizing the likelihood of node co-occurrences within random walks, following ideas from the Skip-Gram model in natural language processing. Nevertheless, for large networks, simulating random walks is computationally expensive and requires additional parameter settings. To alleviate this effect, recent studies have formulated the problem of random walk embedding generation as a matrix factorization task (Levy and Goldberg, 2014; Qiu *et al.*, 2018).

Focusing on a specific instance of such approaches, the DeepWalk method first generates a corpus $\mathcal{W}$ by performing random walks on a graph (Perozzi *et al.*, 2014). 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 $v_1, v_2, \ldots, v_N$, where N is the length of the random walk. The context of node v_i is given as the surrounding nodes in a $2w$-sized window $\{v_{i-w}, \ldots, v_{i-1}, v_{i+1}, \ldots, v_{i+w}\}$, $w < N$. Following Levy and Goldberg (2014) and Qiu *et al.* (2018), the closed form expression of the DeepWalk matrix for any graph $\mathcal{G}_l$ is given by:

$$\underbrace{\log\left(\frac{\#(v,c)|\mathcal{W}|}{\#(v).\#(c)}\right) - \log b}_{\text{Skip-Gram}} = \underbrace{\log\left(\frac{\text{vol}(\mathcal{G}_l)}{bw}\left[\frac{1}{w}\sum_{r=1}^{w}\mathbf{P}^r\right]\mathbf{D}^{-1}\right)}_{\text{DeepWalk matrix}}.$$

On the left-hand side, $\#(v,c), \#(v)$ and $\#(c)$ denote respectively the number of times node-context pair (v, c), node v and context 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 $\mathcal{G}_l$, and the power matrix $\mathbf{P}$ defined as $\mathbf{D}^{-1}\mathbf{A}$. Here, $\text{vol}(\mathcal{G}_l)$ is the volume (size) of $\mathcal{G}_l$. For a detailed theoretical explanation, refer to Section 2.2 in Qiu *et al.* (2018). Inspired by this formulation, we have chosen to obtain the set of PPMI matrices $\mathbf{M} = \{\mathbf{M}^{(l)}\}_{l=1}^{L}$ using the closed form expression of the DeepWalk matrices $\mathbf{M}^{(l)}$ for a multilayer graph $\mathcal{G}$:

$$\mathbf{M} = \left\{\log\left(\frac{\text{vol}(\mathcal{G}_l)}{bw}\left[\frac{1}{w}\sum_{r=1}^{w}\left(\mathbf{P}^{(l)}\right)^r\right]\left(\mathbf{D}^{(l)}\right)^{-1}\right)\right\}_{l=1}^{L}. \quad (1)$$

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 Section 2, the PPMI matrix for deepNF and Mashup is computed using 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.

3.2 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

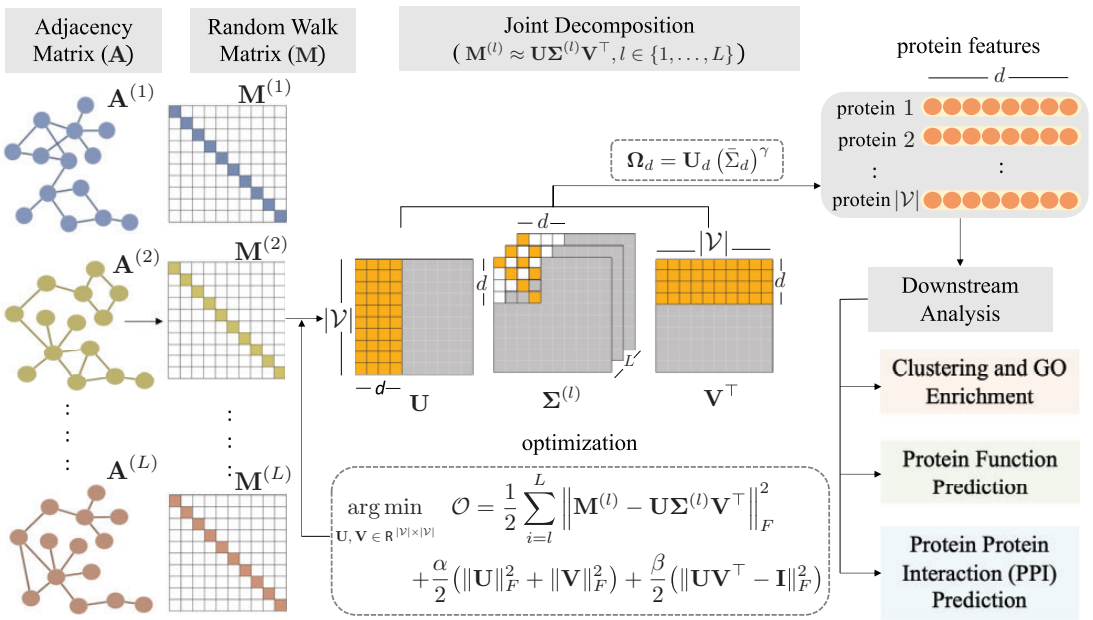

Fig. 1. Schematic representation of BraneMF. The framework takes as input a set of PPI networks represented by their adjacency matrices $\mathbf{A}^{(l)}, l \in \{1,2,\ldots,L\}$. For each PPI network, the random walk matrix $\mathbf{M}^{(l)}$ is computed. For integrative analysis, we learn protein features 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 $\mathbf{U}_d(\mathbf{\Sigma}_d)^\gamma$, where d is the embedding dimension and γ is a factor that scales the magnitude of the singular values. The learned protein features are then utilized for functional analysis of proteins

of the curse of dimensionality, these features are not compatible for 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 (Eq. 1), on which joint matrix factorization is eventually performed. In order to learn the spectrum of one layer in graph $\mathcal{G}$, the singular values and singular vectors of its PPMI matrix $\overline{\mathbf{M}}$ can be obtained using SVD, as $\overline{\mathbf{M}} = U\Sigma\mathbf{V}^\top$, where $\mathbf{U}$ and $\mathbf{V}$ correspond to the left and right singular vector matrices, and $\mathbf{\Sigma}$ is the diagonal singular value matrix. In the case of a multilayer graph $\mathcal{G}$ composed by 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: $\mathbf{M}^{(l)} \approx \mathbf{U}\mathbf{\Sigma}^{(l)}\mathbf{V}^\top$, $l \in \{1, \ldots, L\}$. The correspondence above keeps, where $\mathbf{U}$ and $\mathbf{V}^\top$ are orthogonal matrices containing the joint singular vectors and $\mathbf{\Sigma}^{(l)} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$ contains the corresponding singular values in the layer l. The minimization of the following objective function $\mathcal{O}$ yields $\mathbf{U}$ and $\mathbf{V}$:

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

where $\mathbf{I} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$ is the 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 Eq. (2) to get $\mathbf{U}$ and $\mathbf{V}^\top$. Since Eq. (2) 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 (Dong *et al.*, 2012). The derivatives of $\mathcal{O}$ with respect to $\mathbf{U}$ and $\mathbf{V}^\top$ are given in Supplementary Eq. (S1). For non-convex optimization, a good initialization is important, and we suggest to compute the SVD of the mean for all matrices $\mathbf{M}^{(l)}$, and initialize $\mathbf{U}$, $\mathbf{\Sigma}$ and $\mathbf{V}^\top$ with the resulting matrices: $\mathbf{U}$ is the set of joint singular vectors, namely a joint spectrum shared by all layers in $\mathcal{G}$, $\overline{\mathbf{\Sigma}}$ is the joint singular value matrix computed by taking the average of $\mathbf{\Sigma}^{(l)}$ matrices. The integrated embeddings $\mathbf{\Omega}_d \in \mathbb{R}^{n\times d}$ are obtained by multiplying the first d columns of $\mathbf{U}$, $\mathbf{U}_d \in \mathbb{R}^{n\times d}$, scaled by the γth power of the singular value magnitudes, $(\overline{\mathbf{\Sigma}}_d)^\gamma \in \mathbb{R}^{d\times d}$:

$$\mathbf{\Omega}_d = \mathbf{U}_d(\overline{\mathbf{\Sigma}}_d)^\gamma. \quad (3)$$

The stopping condition is defined by the convergence behavior of the cost function—the difference between its values for two consecutive iterations. This optimization process is similar to (Dong *et al.*, 2012), 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 specificities of each layer. An overview of BraneMF is given in Supplementary Algorithm S1.

4 Results

4.1 Experimental setup

To substantiate our methodology, we apply it over six yeast STRING networks. Their relationships are mainly defined by 'Neighborhood', 'Fusion', 'Co-occurrence', 'Co-expression', 'Experimental' and 'Database'. A brief overview of the input data and the gene ontology (GO) terms are given in Supplementary Tables S1 and S2, respectively. Let $\mathcal{G}$ be a multilayer network constructed over $|\mathcal{V}| = 4900$ proteins. BraneMF depends on three parameters: embedding size d, Eq. (3), window size w, Eq. (1), and weighting factor γ, Eq. (3). Given this set as input to BraneMF, d-dimensional latent features, $\mathbf{\Omega}_d \in \mathbb{R}^{|\mathcal{V}|\times d}$, are learned. We have selected nine baselines in our empirical analysis. Their description and parameter selection are provided in Supplementary Section S4. Further, we investigate the usefulness of the learned embeddings for various omics inference tasks. Firstly, we perform clustering using the protein features and by GO enrichment analysis we show functional relatedness of proteins in these clusters (Section 4.2). Secondly, we design protein function prediction as a multi-label node classification problem by training a SVM model. We predict biological process (BP), molecular function (MF) and cellular component (CC) (Section 4.3). Next, we perform PPI prediction by learning protein features using datasets of 2015 (deepNF and Mashup benchmark datasets) aiming to predict unseen PPIs which are updated by 2021 (Szklarczyk *et al.*, 2021). Moreover, we use the computed embeddings from our dataset to reconstruct the integrated yeast PPI network provided in STRING database (Section 4.4). Note that, for each d, we select one embedding file per model that provides the best performance for each downstream task. The respective parameters selection is given in Supplementary Tables S4 and S5. Lastly, we have also performed parameter uncertainty analysis, examining the impact of the different choices for d, γ and w. Details are provided in Supplementary Section S8. The experiments and their respective evaluations are described below.

4.2 Clustering and GO enrichment of proteins

Proteins are building blocks of a biological system that facilitate cellular processes. Their discovery, functional annotation and characterization are of great importance (Wood *et al.*, 2019). Therefore, we examine the ability of the learned features $\mathbf{\Omega}_d$ to cluster proteins of similar functions. Due to the large number of proteins (i.e. 4900), we choose higher numbers for clusters, with $k \in \{40, 60, 80, 100\}$. Our choice of clustering algorithm is the k-means++ (Arthur and Vassilvitskii, 2007). We execute it 20 times to take into account the randomness in the algorithm. For each of the obtained clusters, we perform Gene Set Enrichment Analysis (GSEA) (Subramanian *et al.*, 2005) using the 'GO_Biological_Process_2018' and 'GO_Molecular_Function_2018' libraries of the *YeastEnrichr* database (Kuleshov *et al.*, 2016) consisting of 1649 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 the significantly enriched clusters, the performance is measured by the enrichment score (ES) and Z-score (Subramanian *et al.*, 2005). The definition of these metrics is given in Supplementary Section S5. The final scores are calculated by computing the average over the 20 simulations. Similarly, we evaluate the clusters obtained for the baseline methods.

GSEA results with ES and Z-score are shown in Table 1 and Supplementary Table S6. The performance of BraneMF measured by ES is higher for $k = 60$, 80 and 100 compared to the baselines.

Table 1. GO enrichment analysis (ES)

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	**25.7±1.7**	22.7±1.2	26.3±1.0	26.2±1.1
Multi-Net	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	<u>46.6±0.1</u>	<u>46.6±0.1</u>
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	<u>41.9±1.4</u>	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
BraneMF	<u>24.05±9.3</u>	**46.2±5.5**	**48.9±4.28**	**49.5±3.38**

Note: Performance of BraneMF compared to the baselines, measured by the ES. Standard deviation is computed for 20 runs of k-means clustering. Bold: best score, underlined: second best score. Parameters: $\gamma = 1, w = 10, d = 128$ for BraneMF; Supplementary Section S4 for baselines.

For $k = 40$, deepNF achieves higher performance than BraneMF. Overall, this demonstrates that clusters from BraneMF's features can group proteins into more significantly enriched biological processes for higher values of k. Moreover, Multi-n2v and BraneExp are the second best performing methods. The visual representation of the obtained clusters and their respective enrichment is given in Supplementary Figure S1. This is our preliminary analysis to show the ability of the learned embeddings to cluster functionally related proteins. Nevertheless, the extensive analysis with different values of k could reveal the optimal k for each model. In the next section, we apply the learned embeddings to protein function prediction.

4.3 Protein function prediction

In this section, we investigate the reliability of learned 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 a SVM classifier and predict probability scores for each protein. We use the SVM implementation provided in the LIBSVM package (Chang and Lin, 2011). To measure the performance of the SVM on the embedding vectors, we adopt a 5-fold cross validation (CV) process (Cho *et al.*, 2016; Gligorijević *et al.*, 2018). We split all the annotated proteins into a training set, comprising of 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 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 m-AUPR, M-AUPR, ACC and F1 used for protein function prediction are mentioned in Supplementary Section S6.1.

We first investigate the added value of integration for protein function prediction. To do this, we have learned protein features for each input network and performed classification. We 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. The results for level I for BP, MF and CC are shown in Figure 2, while the results for levels II and III are shown in Supplementary Figure S2. We observe that integration outperforms individual network protein function prediction. Also, 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.

In addition, we have explored three different network integration strategies namely early, intermediate and late. 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 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 in Figure 3 and Supplementary Figure S3, BraneMF outperforms the rest 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

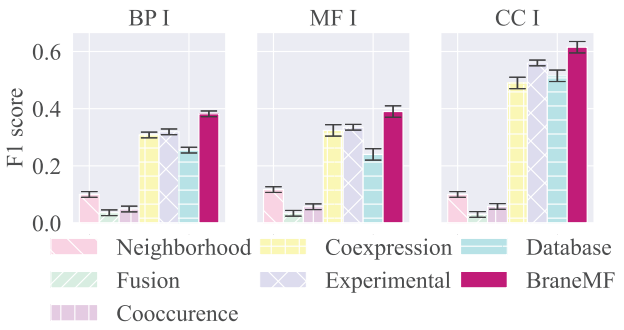

Fig. 2. Integrating multiple networks outperforms individual network. Performance of BraneMF applied on individual yeast STRING networks, measured by the F1 score. Parameters: $\gamma = 1, w = 10, d = 128$. Error bars show the standard deviation across 10 CV trials

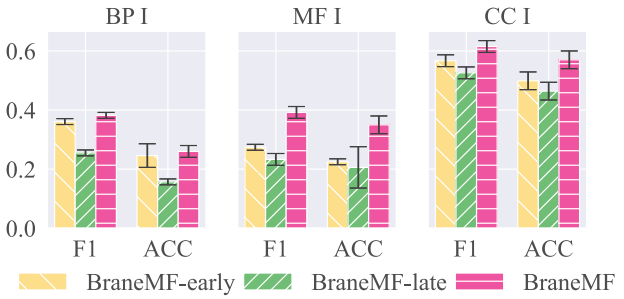

Fig. 3. Integration strategies. Performance of BraneMF compared to early and late integration measured by F1 score. Parameters: $\gamma = 1, w = 10, d = 128$. The error bars show the standard deviation across 10 CV trials

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 that is challenging for late integration.

We also compare the performance of BraneMF to the baseline methods with the validation strategies described earlier. Table 2 and Supplementary Tables S7–S9 show the classification results for level I, II and III of BP, MF and CC, respectively. 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. For instance, the F1 score for BraneMF (BP I) is 38.2%, that is 3 points higher than BraneExp and 4.2 points higher than Graph2GO, the second best performing methods. Whereas BraneMF correctly assigned 26% of proteins (on average) to BP I category, in contrast to 24.9% for Graph2GO and 22% for BraneExp. Similarly, BraneMF consistently outperforms the baselines for level II and level III.

4.4 PPI prediction

The interactome is the complete map of PPIs that can occur in an organism. It is still an open question whether a complete interactome of any organism will ever be discovered by experimental techniques (Keskin *et al.*, 2016). Therefore, predictive methods have become more popular in systems biology to reveal the wiring patterns of proteins. Effective integration of PPIs from different data sources (experimental and/or computational) can help us to have a near complete set of interactome (Keskin *et al.*, 2016). 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. The

Table 2. Protein function prediction

Method	BP I	MF I	CC I
SNF	0.199±0.01	0.104±0.00	0.206±0.01
Mashup	0.271±0.0	0.263±0.02	0.520±0.02
deepNF	0.341±0.01	0.342±0.02	0.564±0.02
Multi-Net	0.335±0.01	0.353±0.02	0.532±0.02
Multi-n2v	0.331±0.01	0.323±0.01	0.511±0.01
OhmNet	0.321±0.01	0.300±0.01	0.512±0.01
MultiVERSE	0.312±0.01	0.294±0.01	0.502±0.02
BraneExp	0.352±0.01	0.368±0.01	0.548±0.03
Graph2GO	0.340±0.01	0.355±0.01	0.564±0.02
BraneMF	**0.382±0.01**	**0.392±0.02**	**0.615±0.02**

Note: Performance of BraneMF, compared to baseline methods using the F1 score. Parameters: $\gamma = 1, w = 10, d = 128$ for BraneMF, Supplementary Section S4 for baselines. The standard deviation is computed based on 10 CV trials. Bold: best score, underlined: second best score.

Table 3. PPI prediction performance

Method	AUPR-H	AUROC-H	AUPR-C	AUROC-C
SNF	0.637	0.628	0.575	0.559
Mashup	0.757	0.743	0.712	**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	0.777	**0.760**	0.683	0.680
Graph2GO	0.721	0.757	0.502	0.498
BraneMF	**0.783**	0.747	**0.725**	0.682

Note: 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. Parameters: $\gamma = 1, w = 10, d = 128$ for BraneMF, Supplementary Section S4 for baselines.

learned embeddings (Section 3) of protein u and v, given as $\Omega_d[u]$ and $\Omega_d[v]$, are converted into edge feature vectors by applying the coordinate-wise Hadamard product or cosine similarity operations (Grover and Leskovec, 2016). Definitions of these operations are given in Supplementary Section S7.1. We perform the prediction task using logistic regression classifier with L2 regularization. The performance of PPI prediction is evaluated based on AUROC (area under the Receiver Operating Characteristic curve) and AUPR (area under the Precision-Recall curve). The results are shown in Table 3. 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.

In addition, we reconstructed the yeast STRING network using the learned representations. The details are provided in Supplementary Section S7.2. The respective results are shown in Supplementary Figure S4. Here, we observe that BraneMF outperforms all baseline models for both evaluation metrics. For the top 1000 edges, notably all the baseline methods except SNF, give 100% of Precision. When we increase the number of edges to 1 million, BraneMF and Multi-n2v continue to show higher performance when compared to the baseline models.

5 Discussion

The wide availability of omics data has driven the need for the development of novel integrative methods that can properly analyze and interpret them. We have presented BraneMF, an integrative framework for analyzing the topology of multiple PPI networks toward extracting relevant protein features from heterogeneous data sources. BraneMF performs integration of multilayer biological networks with the concept of joint matrix factorization that generalizes random walk-based network embedding models. More precisely, BraneMF brings the best of two worlds: expressiveness of the well-celebrated random walk-based embedding models [e.g. DeepWalk (Perozzi *et al.*, 2014), node2vec (Grover and Leskovec, 2016)] and the solid formulation of matrix factorization—going further by extending them to integrate multiple sources. We have demonstrated the wide applicability of BraneMF 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 nine other baseline models, BraneMF has shown competitive performance in all downstream assessments. In a modeling framework that integrates multiple sources, it is imperative to define the uncertainty of the model's predictions. We have performed sensitivity analysis for three parameters, namely γ, w and d that BraneMF depends on. The embedding size range is consistent with the current literature. The selected sizes of the embedding vectors are $\{128, 256, 512, 1024\}$ for BraneMF and all the baseline models. Also, w and γ are given as $\{2, 4, 6, 8, 10\}$ and $\{0, 0.25, 0.50, 0.75, 1\}$, respectively. The details are presented in Supplementary Section S8. We observed that BraneMF performs relatively consistently, even with smaller dimension sizes, i.e. 128. For other networks with a smaller or larger number of nodes, the embedding dimensions used are mostly selected empirically. The trade-off is between accuracy and computational time. Large embedding sizes may potentially increase the performance in the downstream tasks, since the vectors could capture extended aspects of a node. Yet, higher dimensions drastically affect computing time and parametrization effort. Therefore, for smaller networks we believe $d = 128$ could be an ideal choice, while for larger networks, such as the PPI networks for human (with approximately 25 000 genes), the size can be increased from 128 to 512 or 1024 depending on the task and computing capacity.

In summary, we conclude that BraneMF is simpler, depends on less parameters, and produces results comparable, if not better, to more complex methods (e.g. deepNF). Although our formulation is expressive enough to capture these representations, its multiscale properties have certain limitations. First, the model learns one global representation that coalesces all possible scales of network relationships. Hence, different scales of the representation are not independently accessible. In addition, our approach lacks to capture long-range node dependencies (i.e. higher values of w) which could be interesting to study (Chanpuriya and Musco, 2020). As future work, 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 realworld. Besides, it is also possible to upgrade BraneMF to take into account protein (node) features such as biochemical properties and protein sequences in the learning process (Zhou *et al.*, 2020b). These data types can provide insights toward more accurate predictions for functional analysis of proteins. The functionality and applicability of BraneMF are beyond embedding proteins thus not limited to biological networks. BraneMF is a versatile tool that provides an effective, unified and scalable network integration framework with diverse applications.

Funding

This work was supported in part by ANR (French National Research Agency) under the JCJC project GraphIA [ANR-20-CE23-0009-01].

Conflict of Interest: none declared.

References

Arthur,D. and Vassilvitskii,S. (2007) k-means++: the advantages of careful seeding. In: *Proceedings of the Annual ACM-SIAM Symposium Discrete Algorithms, New Orleans, USA*, pp. 1027–1035.

Bagavathi,A. and Krishnan,S. (2018) Multi-Net: a scalable multiplex network embedding framework. In: Aiello,L. *et al.* (eds.) *Proceedings of the International Conference Complex Network Application,* Cambridge, UK, Vol. **813**, pp. 119–131. Springer, Cham.

Çelikkanat,A. and Malliaros,F.D. (2020) Exponential family graph embeddings. *Proc. Conf. AAAI Artif. Intell.*, **34**, 3357–3364.

Chang,C.-C. and Lin,C.-J. (2011) LIBSVM: a library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, **2**, 1–27.

Chanpuriya,S. and Musco,C. (2020) InfiniteWalk: deep network embeddings as Laplacian embeddings with a nonlinearity. In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, California, USA*, pp. 1325–1333.

Cho,H. *et al.* (2016) Compact integration of multi-network topology for functional analysis of genes. *Cell Syst.*, **3**, 540–548.e5.

Cozzetto,D. *et al.* (2013) Protein function prediction by massive integration of evolutionary analyses and multiple data sources. *BMC Bioinformatics*, **14**, S1.

Di Nanni,N. *et al.* (2020) Network diffusion promotes the integrative analysis of multiple omics. *Front. Genet.*, **11**, 106.

Dong,X. *et al.* (2012) Clustering with multi-layer graphs: a spectral perspective. *IEEE Trans. Signal Process*, **60**, 5820–5831.

Fan,K. *et al.* (2020) Graph2GO: a multi-modal attributed network embedding method for inferring protein functions. *GigaScience*, **9**, 8.

Gligorijević,V. and Pržulj,N. (2015) Methods for biological data integration: perspectives and challenges. *J. R. Soc. Interface*, **12**, 20150571.

Gligorijević,V. *et al.* (2018) deepNF: deep network fusion for protein function prediction. *Bioinformatics*, **34**, 3873–3881.

Grover,A. and Leskovec,J. (2016) node2vec: scalable feature learning for networks. In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, USA*, pp. 855–864.

Guo,Q. *et al.* (2016) Lévy random walks on multiplex networks. *Sci. Rep.*, **6**, 37641.

Hamilton,W.L. *et al.* (2017) Representation learning on graphs: methods and applications. *IEEE Data Eng. Bull.*, **40**, 52–74.

Hu,X. *et al.* (2020) Integration of single-cell multi-omics for gene regulatory network inference. *Comput. Struct. Biotechnol. J.*, **18**, 1925–1938.

Jagtap,S. *et al.* (2021) Multiomics data integration for gene regulatory network inference with exponential family embeddings. In: *Proceedings of the European Signal Image Conference,* Dublin, IRE, pp. 1221–1225.

Keskin,O. *et al.* (2016) Predicting protein–protein interactions from the molecular to the proteome level. *Chem. Rev.*, **116**, 4884–4909.

Kuleshov,M.V. *et al.* (2016) Enrichr: a comprehensive gene set enrichment analysis web server 2016 update. *Nucleic Acids Res.*, **44**, W90–W97.

Levy,O. and Goldberg,Y. (2014) Neural word embedding as implicit matrix factorization. *Adv. Neural Inf. Process. Syst.*, **27**, 2177–2185.

Li,Y. *et al.* (2018) A review on machine learning principles for multi-view biological data integration. *Brief. Bioinformatics*, **19**, 325–340.

Liu,W. *et al.* (2017) Principled multilayer network embedding. In: *Proceedings of the IEEE International Conference on Data Mining Workshops (ICDMW), New Orleans, USA*, pp. 134–141.

Liu,Z. *et al.* (2020) Fast attributed multiplex heterogeneous network embedding. In: *Proceedings of the ACM International Conference on Information and Knowledge Management*, pp. 995–1004.

Luo,Y. *et al.* (2017) A network integration approach for drug-target interaction prediction and computational drug repositioning from heterogeneous information. *Nat. Commun.*, **8**, 1–13.

Mikolov,T. *et al.* (2013) Distributed representations of words and phrases and their compositionality. In: *Proceedings of the Annual Conference on Neural Information Processing Systems, California, USA*, Vol. **2**, pp. 3111–3119.

Nguyen,D. and Malliaros,F.D. (2018) BiasedWalk: biased sampling for representation learning on graphs. In: *IEEE International Conference on Big Data, Seattle, WA, USA*, pp. 4045–4053.

Peng,J. *et al.* (2021) Integrating multi-network topology for gene function prediction using deep neural networks. *Brief. Bioinformatics*, **22**, 2096–2105.

Perozzi,B. *et al.* (2014) DeepWalk: online learning of social representations. In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, USA*, pp. 701–710.

Pio-Lopez,L. *et al.* (2021) MultiVERSE: a multiplex and multiplex-heterogeneous network embedding approach. *Sci. Rep.*, **11**, 8794.

Qiu,J. *et al.* (2018) Network embedding as matrix factorization: unifying DeepWalk, LINE, PTE, and node2vec. In: *Proceedings of the ACM International Conference on Web Search and Data Mining, California, USA*, pp. 459–467.

Rudolph,M. *et al.* (2016) Exponential family embeddings. In: *Proceedings of the Annual Conference on Neural Information Processing Systems, New York, USA*, pp. 478–486.

Solé-Ribalta,A. *et al.* (2016) Random walk centrality in interconnected multilayer networks. *Physica D*, **323-324**, 73–79.

Subramanian,A. *et al.* (2005) Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proc. Natl. Acad. Sci. USA*, **102**, 15545–15550.

Subramanian,I. *et al.* (2020) Multi-omics data integration, interpretation, and its application. *Bioinform. Biol. Insights*, **14**, 1–24.

Szklarczyk,D. *et al.* (2021) The STRING database in 2021: customizable protein-protein networks, and functional characterization of user-uploaded gene/measurement sets. *Nucleic Acids Res.*, **49**, D605–D612.

Tsitsulin,A. *et al.* (2018) VERSE: Versatile graph embeddings from similarity measures. In: *Proceedings of the World Wide Web Conference*, pp. 539–548.

Wang,B. *et al.* (2014) Similarity network fusion for aggregating data types on a genomic scale. *Nat. Methods*, **11**, 333–337.

Wilson, J.D. (2020) Analysis of population functional connectivity data via multilayer network embeddings. *Netw. Sci.*, **9**, 99–122.

Wood,V. *et al.* (2019) Hidden in plain sight: what remains to be discovered in the eukaryotic proteome? *Open Biol.*, **9**, 180241.

Yan,J. *et al.* (2018) Network approaches to systems biology analysis of complex disease: integrative methods for multi-omics data. *Brief. Bioinformatics*, **19**, 1370–1381.

Yue,X. *et al.* (2020) Graph embedding on biomedical networks: methods, applications and evaluations. *Bioinformatics*, **36**, 1241–1251.

Zhang,H. *et al.* (2018) Scalable multiplex network embedding. In: *Proceedings of the IJCAI International Joint Conference on Artificial Intelligence, Stockholm*, Vol. **18**, pp. 3082–3088.

Zhou,G. *et al.* (2020a) Network-based approaches for multi-omics integration. *Methods Mol. Biol. (Clifton, N.J.)*, **2104**, 469–487.

Zhou,J. *et al.* (2020b) Graph neural networks: a review of methods and applications. *AI Open*, **1**, 57–81.

Zitnik,M. and Leskovec,J. (2017) Predicting multicellular function through multi-layer tissue networks. *Bioinformatics*, **33**, i190–i198.