

---

# BraneMF: Integration of Biological Networks for Functional Analysis of Proteins

## Supplementary Material

---

Surabhi Jagtap<sup>1,2</sup>, Abdulkadir Çelikkanat<sup>3</sup>, Aurélie Pirayre<sup>1</sup>, Frédérique Bidard<sup>1</sup>, Laurent Duval<sup>1</sup> and Fragkiskos D. Malliaros<sup>2</sup>

<sup>1</sup>IFP Energies nouvelles, 1 et 4 avenue de Bois-Préau, 92852 Rueil-Malmaison, France

<sup>2</sup>Université Paris-Saclay, CentraleSupélec, Inria, Centre for Visual Computing, 91190 Gif-Sur-Yvette, France

<sup>3</sup>DTU Compute, Technical University of Denmark, 2800 Kongens Lyngby, Denmark

**Source code:** the implementation of BraneMF is freely available at:  
<https://github.com/Surabhivj/BraneMF> along with datasets, embeddings, and result files.

## 1 Pseudocode of BraneMF

---

### Algorithm 1 BraneMF

---

**Input:** Multilayer graph  $\mathcal{G} = \{\mathcal{G}_l\}_{l=1}^L$ ;

Parameters: window size:  $w$ , embedding dimension:  $d$ , and weighting factor:  $\gamma$

**Output:**  $d$ -dimension protein features,  $\Omega_d$

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

**return**  $\Omega_d$

---

## 2 Functional derivative of $\mathcal{O}$

To obtain  $\mathbf{U}$  and  $\mathbf{V}$ , we solve the objective function  $\mathcal{O}$  in Eq. (2) using the derivation given below:

$$\frac{\partial \mathcal{O}}{\partial \mathbf{U}} = - \left( \sum_{l=1}^L (\mathbf{M}^{(l)} - \mathbf{U}\Sigma^{(l)}\mathbf{V}^\top) \right) \mathbf{V}\Sigma^{(l)} + \alpha \mathbf{U} + \beta (\mathbf{U}\mathbf{V}^\top - \mathbf{I})\mathbf{V}^\top,$$
$$\frac{\partial \mathcal{O}}{\partial \mathbf{V}^\top} = \Sigma^{(l)}\mathbf{U}^\top \left( \sum_{l=1}^L (\mathbf{M}^{(l)} - \mathbf{U}\Sigma^{(l)}\mathbf{V}^\top) \right) + \alpha \mathbf{V}^\top + \beta \mathbf{U}^\top (\mathbf{U}\mathbf{V}^\top - \mathbf{I}). \quad (\text{S1})$$

### 3 Datasets

#### 3.1 Yeast PPI networks

Network	Nodes	Edges	Density	Evidence
Neighborhood	4,900	7,656	0.008741	Gene order and sequence homology
Fusion	4,900	492	0.003943	Orthology and fusion
Cooccurrence	4,900	1,231	0.003861	Orthology
Coexpression	4,900	54,317	0.006562	Gene expression data
Experimental	4,900	48,190	0.005600	Biochemical, biophysical genetic experiments
Database	4,900	29,231	0.005946	Human curation

Table S1: **Overview of the datasets used in our study.** We select the network edges whose combined score exceeds 900.

#### 3.2 Functional annotations

The functional annotations were downloaded from the Gene Ontology database [4] (May 2022 update). Each category of GO is represented in levels (i.e., level I, II, and III). A lower level (i.e., Level I) represents more specific terms whereas a higher one (i.e., Level III) represents more general terms. Table S2 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 S2: **Overview of the Gene Ontology (GO) terms used for prediction.** Level I:  $10 < \text{proteins per term} < 30$ ; Level II:  $30 < \text{proteins per term} < 100$ ; and Level III:  $100 < \text{proteins per term} < 300$ .

## 4 Baseline models

We have selected nine multi-layer network integration methods to benchmark BraneMF. The brief introduction and the details of their implementation is given below. We test all the methods including BraneMF with the datasets of a well-studied organism, i.e., yeast (tax id: 4392). Our implementation takes as input the taxonomy id of each organism and fetches the recently available data from the STRING database [16]. Moreover, we perform extensive parameter tuning for each baseline method. The details of parameter choices for each method are given in Sec. 4.2 of the supplementary material. We report the best performance for each model in the main results, while the remaining results are provided in the following sections of this supplementary material.

### 4.1 Description and implementation

1. SNF [17]: Similarity network fusion (SNF) is a computational method for data integration. SNF first constructs similarity network of samples (e.g., patients) for each available data type and then efficiently fuses them into one network. We have applied SNF to integrate our datasets. Since we already have PPI networks, we have skipped the network construction step. We obtain the integrated (fused) network from SNF with the best parameters given by the authors. Moreover, we need features for each node of this integrated network. To obtain features we have performed SVD on the graph and obtain node features as the first  $d$  columns of the left singular matrix  $U$ , where  $d$  is the embedding size.

Implementation: [http://compbio.cs.toronto.edu/SNF/SNF/Software\\_files/SNFtool\\_v2.1.tar.gz](http://compbio.cs.toronto.edu/SNF/SNF/Software_files/SNFtool_v2.1.tar.gz)  
[http://compbio.cs.toronto.edu/SNF/SNF/Software\\_files/SNFmatlab\\_v2.1.zip](http://compbio.cs.toronto.edu/SNF/SNF/Software_files/SNFmatlab_v2.1.zip)

2. Mashup [3]: Mashup extracts a compact vector representation of the network that could explain topological patterns of nodes in multiple heterogeneous interaction networks.

Mashup is closely related to BraneMF except for some perceptible differences:

- Firstly, node proximities in Mashup are captured using a random walk with restart (RWR) matrix. In BraneMF, we have chosen to utilize PPMI matrices that are computed from standard random walk (RW) matrices without restart probabilities. Specifically, in Mashup, the RWR for a node  $i \in \mathcal{V}$  is defined by

$$\mathbf{s}_i^{t+1} = (1 - p_r)\mathbf{P}\mathbf{s}_i^t + p_r\mathbf{e}_i, \quad (S2)$$

where  $p_r$  is the return probability and  $t$  is the number of steps. The matrix  $\mathbf{S} := [\mathbf{s}_1^\infty | \dots | \mathbf{s}_i^\infty | \dots | \mathbf{s}_N^\infty] \in \mathbb{R}^{N \times N}$  is constructed, where each  $\mathbf{s}_i^\infty$  is the column vector for node  $i \in \mathcal{V}$ . Each entry of  $\mathbf{s}_i^\infty$  indicates the probability of visiting  $j$  by starting from node  $i$  over an infinite walk with an additional return probability. For BraneMF, the choice of the PPMI matrix is inspired by the relationship between SkipGram-based random walk embeddings (e.g., DeepWalk) and matrix factorization [13]. This formulation has shown significant improvement in single-layer graph embedding methods [13]. Taking into account this inspiration, we have utilized the concept of jointly factorizing PPMI matrices for multilayer graph embedding. The PPMI matrix is defined in Eq. (1) of the main paper. Besides, the PPMI formulation has a window size parameter ( $T$ ), which is used to control the similarity definition between nodes. Thus, it introduces a flexible framework which can be adapted depending on the structural properties of a given network.

- Secondly, Mashup and BraneMF rely on different computational optimization techniques to obtain integrative embeddings. Mashup has two instances for learning embeddings. (i) A multinomial logistic model for dimension reduction by integrating the diffusion states of each node in each network layer. (ii) Optimization based on singular value decomposition (SVD). The later instance of Mashup is close to BraneMF’s joint matrix factorization. Mashup simply concatenates the normalized RWR matrices, and then it performs SVD [3]. More precisely, the resulting concatenated matrix  $\mathbf{S}$  has dimension of  $NL \times L$ . The logarithm of  $\mathbf{S}$  is given as  $\tilde{\mathbf{S}}$  and truncated SVD on  $\tilde{\mathbf{S}}$  is performed (with a user-specified number of components) to get a low-rank factorization  $\mathbf{U}\Sigma\mathbf{V}^\top$ . On the contrary, we propose an efficient way to combine the spectra of PPMI matrices, whose result can be viewed as the joint spectrum shared by all the graph layers. For a multilayer graph  $\mathcal{G}$  composed of  $L$  layers, the PPMI matrices  $\mathbf{M}^{(l)}$  are computed for each  $l \in \{1, \dots, L\}$ . This set of  $\mathbf{M}^{(l)}$  matrices are jointly decomposed to singular vector  $\Sigma^{(l)}$  and singular value matrices ( $\mathbf{U}$  and  $\mathbf{V}$ ) that are shared across all layers (see Sec. 3.2 in the main paper).

Implementation: <https://groups.csail.mit.edu/cb/mashup/mashup.tar.gz>

3. deepNF [6]: deepNF is a network fusion method based on a multimodal deep autoencoder (MDA) to integrate different heterogeneous networks into a compact, low-dimensional feature representation common to all networks.

Implementation: <https://github.com/VGligorijevic/deepNF>

4. MultiNet [1]: MultiNet is a fast scalable multiplex network embedding framework. First, it computes random walks on all nodes across multiple network layers. Then it merges all sequences of random walks in one document and learns node features by using an optimization that maximizes the likelihood of neighbors of a node across network layers.
5. Multi-node2vec [18]: Multi-node2vec is a fast network embedding model for multilayer/multiplex networks that identifies a continuous and low-dimensional representation for the unique nodes in the network. It is an extension of the Node2Vec model for multilayer networks.

Implementation: <https://github.com/jdwilson4/multi-node2vec>

6. OhmNet [20]: OhmNet is a hierarchy-aware unsupervised learning approach for multi-layer networks. It learns node features from a multi-layer network, where each layer represents a protein-protein interaction network and all these networks are in a tree structure. Since our dataset has no hierarchy, we have used a flat tree where the network hierarchy is flattened; PPI networks are not rendered inside of each other, but instead are rendered as sibling hierarchy elements (i.e. sharing the same parent node) toward features in the common parent in the hierarchy.

Implementation: <https://github.com/mims-harvard/ohmnet>

7. MultiVERSE [12]: MultiVERSE is an extension of the VERSE framework using Random Walks with Restart on Multiplex (RWR-M) and Multiplex-Heterogeneous (RWR-MH) networks. Since all our networks share the same set of nodes, we have used the RWR-M model of MultiVERSE for integrative analysis.

Implementation: <https://github.com/LPioL/MultiVERSE>

8. Graph2GO [5]: A graph-based representation learning method for predicting protein functions. Graph2GO uses the VGAE [9] model to learn embeddings from each layer and perform concatenation to obtain final embeddings.

Implementation : <https://github.com/yanzhanglab/Graph2GO>

9. BraneExp [8]: It is a network integration framework that considers expressive conditional probability models to relate nodes within random walk sequences. The model uses exponential family distributions to capture interactions between nodes in random walks that traverse nodes within and across input network layers, learning node features using exponential family graph embeddings.

Implementation : <https://github.com/Surabhivj/BRANE-EXP>

## 4.2 Parameter selection

All multi-layer 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 in performance have been set to the default values. In Table S3, we provide the required parameters for each method that are tuned. Note that the representation in Table S3 is simplified to show the dependency of baseline methods to the different parameters. For some methods, a direct comparison of parameters is not possible since they may share different parameter space. We adopt this approach to simplify the parameter selection strategy.

1. Embedding size ( $d$ ): size of 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 ( $l$ ): it is a parameter to select the length of a node set you 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  $l \in \{15, 20\}$ .
3. Window size ( $w$ ): the number of nodes (proteins) that will be used to determine the context of each node (protein). For instance, in a *walk* of length 3, such as, “proteinA proteinB proteinC”, a window size of 2 would mean your samples are like (proteinA proteinB) or (proteinB proteinC). We chose  $w \in \{2, 4, 6, 8, 10\}$ .
4. Number of walks ( $n$ ): this parameter allows to select the number of random walks that will be sampled per node (protein). We chose  $n \in \{10, 20\}$ .

Method	$d$	$w$	$l$	$n$	$\sigma$	$e$	$\gamma$	$r$	$k$	$b$
SNF	✓	✓	✗	✗	✗	✗	✗	✗	✓	✗
Mashup	✓	✗	✗	✗	✗	✗	✗	✓	✗	✗
DeepNF	✓	✗	✗	✗	✗	✓	✗	✗	✓	✓
MultiNet	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗
Multi-n2v	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗
OhmNet	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗
MultiVERSE	✓	✓	✗	✗	✓	✗	✗	✓	✓	✗
Graph2GO	✓	✗	✗	✗	✓	✓	✗	✗	✗	✗
BraneExp	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗
BraneMF	✓	✓	✗	✗	✗	✗	✓	✗	✗	✗

Table S3: **Overview of parameters considered for tuning.** Green coloured ticks indicate that the method depends on the respective parameters. Red crosses show that method does not depend on a particular parameter.

- Learning rate ( $\sigma$ ): 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  $\sigma$  represents a smaller step along the downward slope. We chose  $\sigma \in \{0.1, 0.01, 0.001, 0.0001\}$ .
- 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  $e \in \{60, 80\}$ .
- Restart probability ( $r$ ): 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  $r \in \{0.8, 0.85, 0.9, 0.95\}$ .
- Gamma ( $\gamma$ ): 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. 1). We chose  $\gamma \in \{0, 0.25, 0.50, 0.75, 1\}$ .
- Number of samples ( $k$ ): it is the parameter to choose the number of times we would like to perform Random Walk with Restart (RWR). We chose  $k \in \{2, 3, 4, 6\}$ .
- Batch size ( $b$ ): it is the number of samples that will be used for training at one time. We chose  $b \in \{32, 64, 128\}$ .

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

Table S4: **Parameter selection I.** The table shows the best performing parameters for the clustering and protein function prediction tasks.

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

Table S5: **Parameter selection II.** The table shows the best performing parameters for the PPI prediction and network reconstruction tasks.

## 5 Gene Ontology (GO) enrichment for clusters

### 5.1 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  $\bar{G}_k = \{\bar{g}_{kj} : 1, \dots, n - n_k\}$ . The Enrichment Score (ES) for a given gene set  $G_k$  is given as:

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

where  $\sup(\cdot)$  is the supremum,  $i$  represents the position in list  $L$ , and

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

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

where  $\mathbb{1}$  is the indicator function for the membership in a given gene set.  $s_t$  is given by the correlation of  $g_{kj}$  and weighted by  $\alpha$  [15].

### 5.2 Z-Score

The Z-score for gene set  $G_k$  is given by:

$$\text{Z-score} = \sqrt{n_k \bar{t}}, \text{ with } \bar{t} = \frac{1}{n_k} \sum_{i \in G_k} t_i,$$

where  $n_k$  is number of genes in  $G_k$  and  $t_i$  is the  $t$ -statistic that is referred to signal to noise ratio for each gene.

The detailed description of the scoring metrics are given in [15, 7]

### 5.3 Results

Method	$k = 40$	$k = 60$	$k = 80$	$k = 100$
Z-score				
SNF	$-1.73 \pm 0.07$	$-1.80 \pm 0.03$	$-1.77 \pm 0.01$	$-1.81 \pm 0.02$
Mashup	$-1.85 \pm 0.14$	$-1.88 \pm 0.12$	$-1.84 \pm 0.03$	$-1.92 \pm 0.02$
deepNF	$-2.10 \pm 0.07$	$-2.09 \pm 0.06$	$-2.03 \pm 0.01$	$-2.02 \pm 0.03$
MultiNet	$-1.87 \pm 0.03$	$-1.99 \pm 0.02$	$-1.97 \pm 0.02$	$-2.05 \pm 0.02$
Multi-n2v	$-2.62 \pm 0.22$	$-2.05 \pm 0.06$	$-1.92 \pm 0.04$	$-1.96 \pm 0.02$
OhmNet	$-1.96 \pm 0.03$	$-1.92 \pm 0.02$	$-2.03 \pm 0.01$	$-1.99 \pm 0.01$
MultiVERSE	$-1.93 \pm 0.04$	$-1.97 \pm 0.02$	$-2.00 \pm 0.03$	$-2.06 \pm 0.02$
BraneExp	$-1.98 \pm 0.04$	$-1.98 \pm 0.04$	$-1.99 \pm 0.02$	$-2.00 \pm 0.02$
Graph2GO	$-2.06 \pm 0.04$	$-1.89 \pm 0.04$	$-2.02 \pm 0.03$	$-2.04 \pm 0.02$
BraneMF	$-1.92 \pm 0.04$	$-1.93 \pm 0.03$	$-1.98 \pm 0.03$	$-2.01 \pm 0.03$

Table S6: **GO enrichment analysis of clusters.** Performance of BraneMF compared to the baselines, measured by Z-scores. Standard deviation is computed for all 20 runs of  $k$ -means clustering. Parameters:  $\gamma = 1, w = 10, d = 128$  for BraneMF, Sec. 4 for baselines.

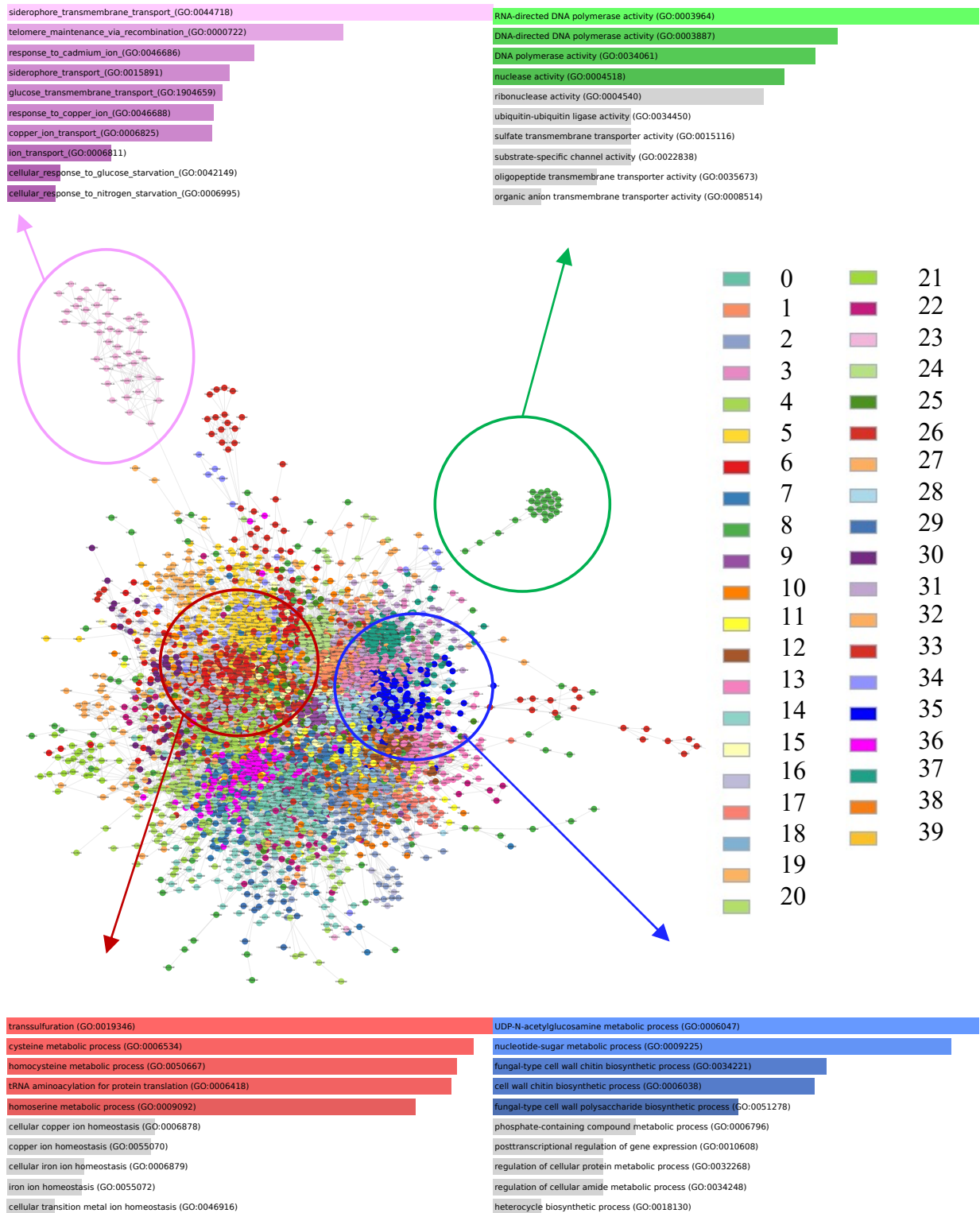


Figure S1: **Visual representation of clusters and their GO enrichment.** We present an example of clustering results of the first simulation of the  $k$ -means++ algorithm for  $k = 40$ . The obtained clusters are then mapped to the integrated STRING network (combined score > 900) visualised using Cytoscape [14] with the default layout. Node colours represent the 40 different clusters. To show the GO enrichment results, we have selected four clusters and their respective enrichment is shown by yeast 'enrichr' barplot results (sorted by  $p$ -value). [10]. Note that, to compare with the baselines, we perform  $k$ -means++ 20 times and compute the ES and Z-scores of significantly enriched clusters for each simulation. Finally, we report the mean ES and Z-score with their standard deviation across 20 simulations (Table S6).

## 6 Protein function prediction

### 6.1 Evaluation metrics

We use the following metrics to evaluate the prediction performance:

- The micro-averaged area under the Precision-Recall curve (m-AUPR) is calculated by vectorizing the matrices of the protein and predicted scores of its function and known binary annotations, then computing the AUPR by using these two vectors.
- The Macro-AUPR (M-AUPR) is computed using the AUPR for each function separately, and then averaging these values across all functions.
- Accuracy (ACC) measures the percentage of test proteins that were correctly predicted.
- Micro-averaged F1 score (F1) is computed by assigning the top three predictions to each protein with micro average parameter, and computing the F1 score [6].

### 6.2 Added value of integration in comparison to individual networks

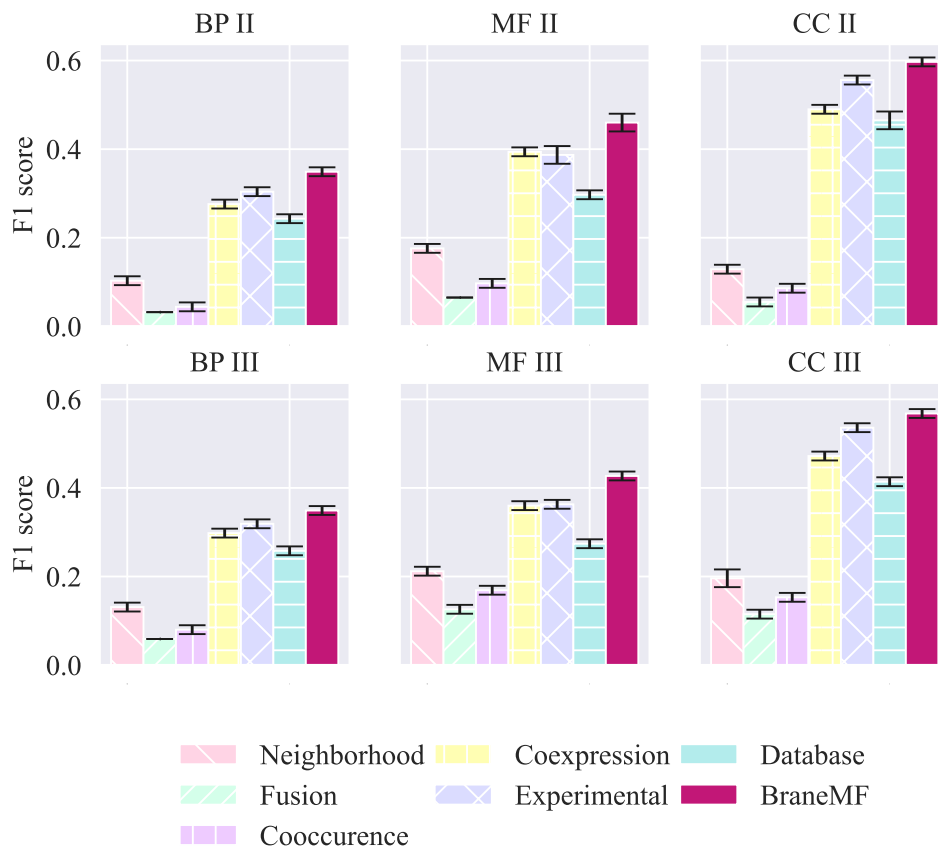


Figure S2: **Integrating multiple networks outperforms individual networks in protein function prediction.** We compare the cross-validation performance of BraneMF on individual yeast STRING networks, measured by F1 score. Parameters:  $\gamma = 1$ ,  $w = 10$ ,  $d = 128$ . The error bars show the standard deviation across 10 CV trials.



### 6.3 Classification performance for early, intermediate, and late integration

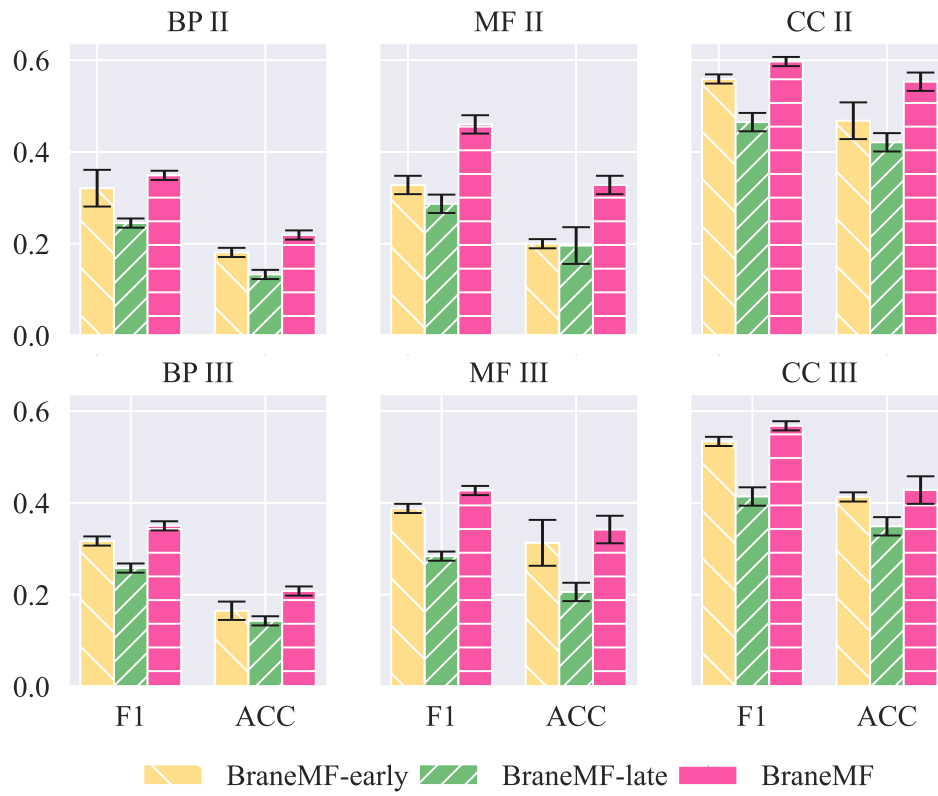


Figure S3: **Integration strategies.** Performance of BraneMF compared to early and late integration, measured by the F1 score. Parameters:  $\gamma = 1, w = 10, d = 128$ . The error bars show the standard deviation across 10 CV trials.

### 6.4 Protein function prediction compared to baseline methods

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	<u>0.280 ± 0.01</u>	<u>0.352 ± 0.01</u>	0.220 ± 0.08
Graph2GO	<u>0.458 ± 0.02</u>	0.279 ± 0.01	0.340 ± 0.01	<u>0.249 ± 0.02</u>
BraneMF	<b>0.504 ± 0.02</b>	<b>0.303 ± 0.01</b>	<b>0.382 ± 0.01</b>	<b>0.260 ± 0.02</b>
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	<u>0.494 ± 0.01</u>	<u>0.406 ± 0.01</u>	<u>0.329 ± 0.01</u>	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	<u>0.204 ± 0.03</u>
Graph2GO	0.487 ± 0.02	0.398 ± 0.02	0.317 ± 0.01	0.185 ± 0.03
BraneMF	<b>0.524 ± 0.02</b>	<b>0.424 ± 0.02</b>	<b>0.349 ± 0.01</b>	<b>0.219 ± 0.02</b>
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	<u>0.343 ± 0.01</u>	<u>0.189 ± 0.01</u>
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	<u>0.568 ± 0.01</u>	<u>0.509 ± 0.01</u>	0.329 ± 0.01	0.162 ± 0.01
BraneMF	<b>0.585 ± 0.01</b>	<b>0.526 ± 0.01</b>	<b>0.350 ± 0.01</b>	<b>0.208 ± 0.01</b>

Table S7: **Protein function prediction (BP)**. Performance of BraneMF, compared to baseline methods for BP using m-AUPR, M-AUPR, F1 and ACC scores. Parameters:  $\gamma = 1, w = 10, d = 128$  for BraneMF, Sec. 4 for baselines. The standard deviation is computed based on 10 CV trials.

Method	m-AUPR	M-AUPR	F1	ACC
MF I				
SNF	0.192 ± 0.01	0.120 ± 0.01	0.104 ± 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	<u>0.306 ± 0.01</u>
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	<u>0.410 ± 0.02</u>	<u>0.256 ± 0.01</u>	<u>0.368 ± 0.01</u>	0.303 ± 0.11
Graph2GO	0.404 ± 0.02	0.243 ± 0.02	0.355 ± 0.01	0.287 ± 0.11
BraneMF	<b>0.457 ± 0.04</b>	<b>0.278 ± 0.02</b>	<b>0.392 ± 0.02</b>	<b>0.350 ± 0.03</b>
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	<u>0.463 ± 0.03</u>	<u>0.368 ± 0.02</u>	<u>0.436 ± 0.02</u>	<u>0.294 ± 0.10</u>
Graph2GO	0.455 ± 0.02	0.359 ± 0.01	0.420 ± 0.01	0.292 ± 0.04
BraneMF	<b>0.518 ± 0.02</b>	<b>0.404 ± 0.02</b>	<b>0.460 ± 0.02</b>	<b>0.328 ± 0.02</b>
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	<u>0.517 ± 0.03</u>	<u>0.454 ± 0.02</u>	<u>0.417 ± 0.02</u>	<b>0.345 ± 0.02</b>
Graph2GO	0.501 ± 0.02	0.448 ± 0.02	0.396 ± 0.01	0.338 ± 0.02
BraneMF	<b>0.541 ± 0.03</b>	<b>0.473 ± 0.02</b>	<b>0.427 ± 0.01</b>	<u>0.342 ± 0.02</u>

Table S8: **Protein function prediction (MF)**. Performance of BraneMF, compared to baseline methods for MF using m-AUPR, M-AUPR, F1 and ACC scores. Parameters:  $\gamma = 1, w = 10, d = 128$  for BraneMF, Sec. 4 for baselines. The standard deviation is computed based on 10 CV trials.

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	<u>0.732 ± 0.03</u>	<u>0.438 ± 0.03</u>	<u>0.564 ± 0.02</u>	<u>0.490 ± 0.03</u>
BraneMF	<b>0.812 ± 0.02</b>	<b>0.470 ± 0.03</b>	<b>0.615 ± 0.02</b>	<b>0.570 ± 0.03</b>
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	<u>0.749 ± 0.02</u>	0.630 ± 0.02	<u>0.559 ± 0.01</u>	0.462 ± 0.04
Graph2GO	<u>0.749 ± 0.01</u>	<u>0.631 ± 0.01</u>	0.549 ± 0.01	<u>0.481 ± 0.04</u>
BraneMF	<b>0.806 ± 0.02</b>	<b>0.666 ± 0.02</b>	<b>0.597 ± 0.01</b>	<b>0.553 ± 0.02</b>
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	<u>0.400 ± 0.02</u>
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	<u>0.706 ± 0.01</u>	<u>0.634 ± 0.01</u>	<u>0.559 ± 0.01</u>	0.378 ± 0.02
Graph2GO	0.701 ± 0.02	0.623 ± 0.02	0.544 ± 0.01	0.387 ± 0.02
BraneMF	<b>0.734 ± 0.01</b>	<b>0.646 ± 0.02</b>	<b>0.568 ± 0.01</b>	<b>0.428 ± 0.03</b>

Table S9: **Protein function prediction (CC)**. Performance of BraneMF, compared to baseline methods for CC using m-AUPR, M-AUPR, F1 and ACC scores. Parameters:  $\gamma = 1, w = 10, d = 128$  for BraneMF, Sec. 4 for baselines. The standard deviation is computed based on 10 CV trials.

## 7 Protein-protein interaction (PPI) prediction

### 7.1 Computation of edge features

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

1. Hadamard product:

$$\Omega[u]_i \times \Omega[v]_i$$

2. Cosine distance:

$$1 - \frac{\Omega[u]_i \cdot \Omega[v]_i}{\|\Omega[u]_i\| \|\Omega[v]_i\|}$$

### 7.2 Network reconstruction

To reconstruct the yeast STRING network using the learned representations, we first compute pairwise node similarities using the normalized inner product (also known as cosine similarity) of the trained embeddings. Then, a candidate PPI network is constructed, whose nodes are proteins and the edges represent the similarity between two proteins, weighted by the similarity score. The total number of possible edges of this reconstructed network is 20.4 million. We evaluate this reconstructed network using yeast STRING PPI of year 2021 (reference network). This network has 1 million edges. For a fair comparison, we narrow down our evaluation to the 1 million top scoring edges of the reference network. The performance is measured using the precision@ $k$ . This metric gives us the number of true edges (edges in the reference network) that are present in the top- $k$  edges of the reconstructed network. We choose to compute precision@ $k$  for each non-negative power of 10 for all the models [11]. Recent studies have suggested that when the dataset is unbalanced (the number PPIs in a PPI network is much smaller than the number of non-PPIs), accuracy could provide an overoptimistic estimation of the classifier’s ability on the majority class (non-PPIs) [2]. Therefore, to have a more comprehensive evaluation, we also measure Matthews correlation coefficient (MCC) for several thresholds. The results for both metrics are shown in Fig. S4. It is observed that BraneMF outperforms all baseline models for both evaluation metrics. Biological networks show small-world properties [19]. These properties can be extracted by focusing on important edges in the graph. For the top 1,000 edges, notably all the baseline methods except SNF, give 100% of Precision. When we increase the number of edges to 1 million, BraneMF continues to show high performance when compared to the baseline models. This demonstrates that BraneMF can capture information accurately even for big networks.

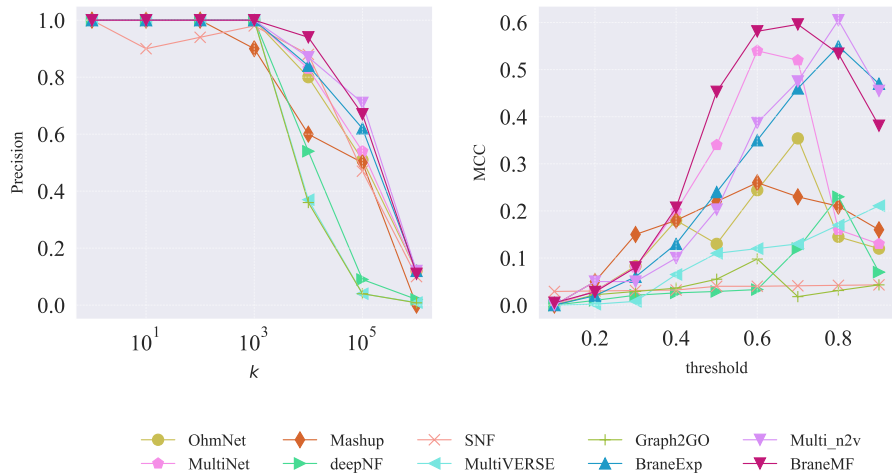


Figure S4: **Network reconstruction.** Performance of BraneMF for the network reconstruction task, compared against baseline models. Left plot: the  $x$ -axis represents the number of top  $k$  PPIs and the  $y$ -axis shows the precision (edges present in the reference network) for these PPIs. Right plot: the  $x$ -axis represents the threshold for selecting the edges and the  $y$ -axis shows the MCC for the respective threshold.

## 8 Uncertainty analysis

To assess the goodness of our results, we have performed various uncertainty analysis tests. Since protein function prediction is the most critical task, we chose to perform parameter sensitivity analysis for it. Our protein features depend on the setting of the  $\gamma$ ,  $w$ , and  $d$  parameters. Therefore, we set the weighting factor  $\gamma$  as  $[0, 0.25, 0.50, 0.75, 1]$ , the window size  $w$  as  $[2, 4, 6, 8, 10]$ , and the dimension  $d$  as  $[128, 256, 512, 1024]$ . For each of the selected values of  $\gamma$  and  $d$ , we learn protein features and perform classification using 5-fold cross-validation across 10 trials. The performance (F1 score and accuracy) w.r.t.  $w$  when  $\gamma = 1$  and  $d = 128$  is shown in Figure S5. We observe that the F1 score for  $w = 6, 8$ , and  $10$  is higher than the one for  $w = 2$  and  $4$ . However, the accuracy for  $w = 2$  is higher for BP and MF. Additionally,  $w = 6$  and  $10$  show higher performance in the accuracy of CC compared to other values of window sizes. In our main results, we keep  $w = 10$  considering all tasks, however for new datasets we recommend setting it to  $6, 8$  or  $10$  depending on the downstream task. The results for different values of  $\gamma$  for level I, II, and III of BP, MF, and CC are shown in Figure S6. We have observed that the performance when  $\gamma = 1$  is higher for all the runs compared to other values of  $\gamma$ . Besides,  $0.5$  is the second best performing value of  $\gamma$ . For new datasets, we recommend to test the performance for  $\gamma = 0.5$  and  $1$ .

Lastly, the embedding dimension is one of the essential parameters considered to test the performance of the model and compare against the baselines. We compare the performance of BraneMF with all baseline models for  $d = \{128, 256, 512, 1024\}$ . We have used a common heuristic approach to pick the range of the embedding size as a power of 2. Powers of 2 will increase cache utilization during data movement, thus reducing bottlenecks. For other networks with fewer or more nodes, the choice of the embedding dimension is mostly done empirically. The trade-off is between accuracy and computational concerns. An embedding vector of higher dimensionality may increase the accuracy in a downstream task, since the vectors can capture more aspects of the nodes. But more dimensions could require a higher computing time and memory resources. The results for the classification performance (F1 score) with respect to  $d$  for level I, II, and III of BP, MF, and CC are shown in Figure S7. We observe that the performance of BraneMF is quite stable compared to other methods for all values of  $d$ . BraneMF performs well even at lower dimensions. The performance of Mashup, SNF, deepNF, Graph2GO, and MultiVERSE increases with  $d$ , while BraneMF, BraneExp, MultiNet, OhmNet, and Multi-node2vec perform better at lower dimensions than higher ones.

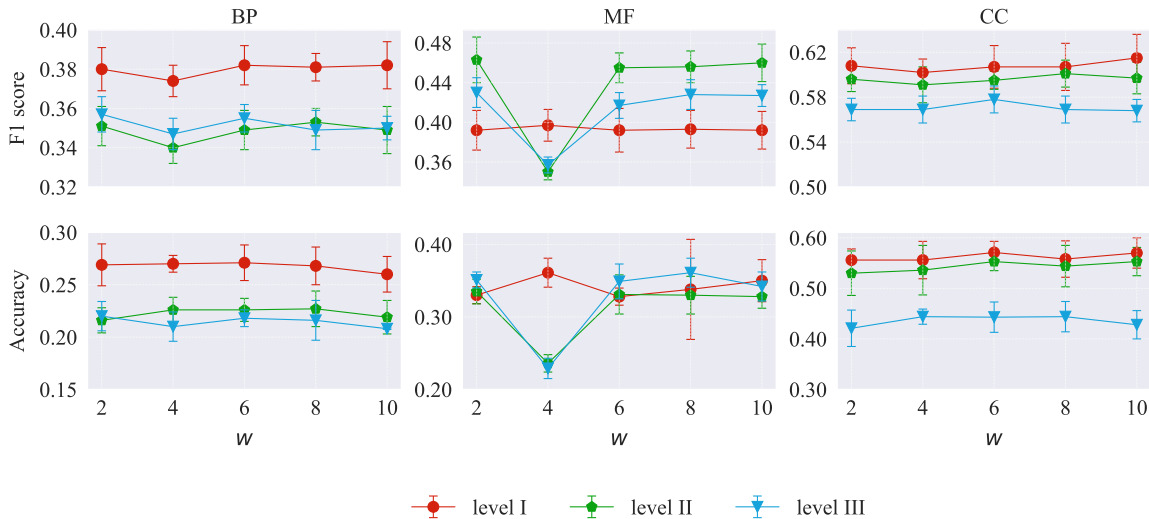


Figure S5: **Cross-validation performance.** Performance (F1 score and accuracy) of BraneMF with  $w \in [2, 4, 6, 8, 10]$ ,  $\gamma = 1$  and  $d = 128$ , in 5-fold cross validation for function prediction, compared to baseline network integration methods. Error bars represent standard deviations across 10 CV trials.

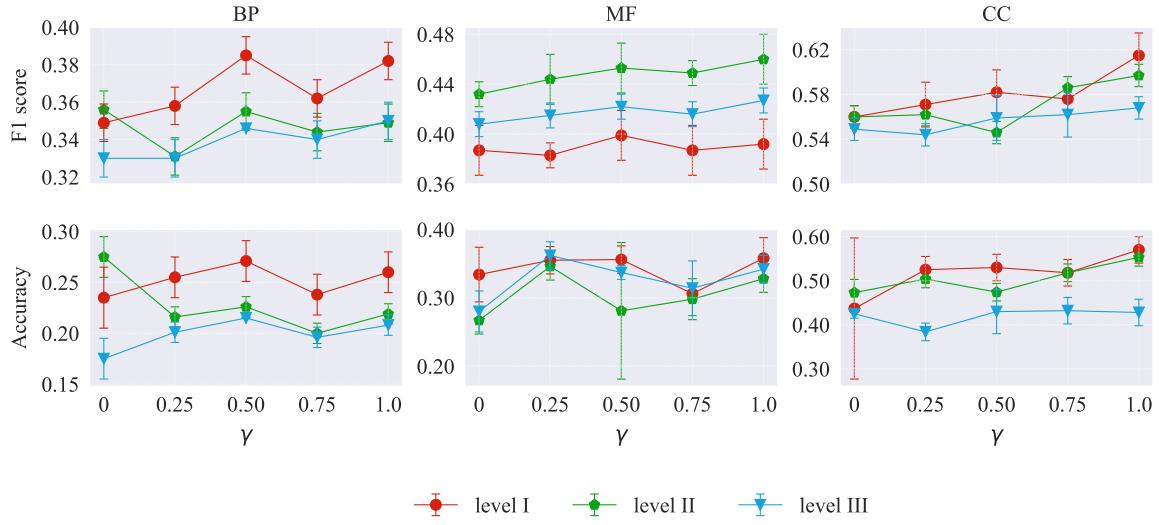
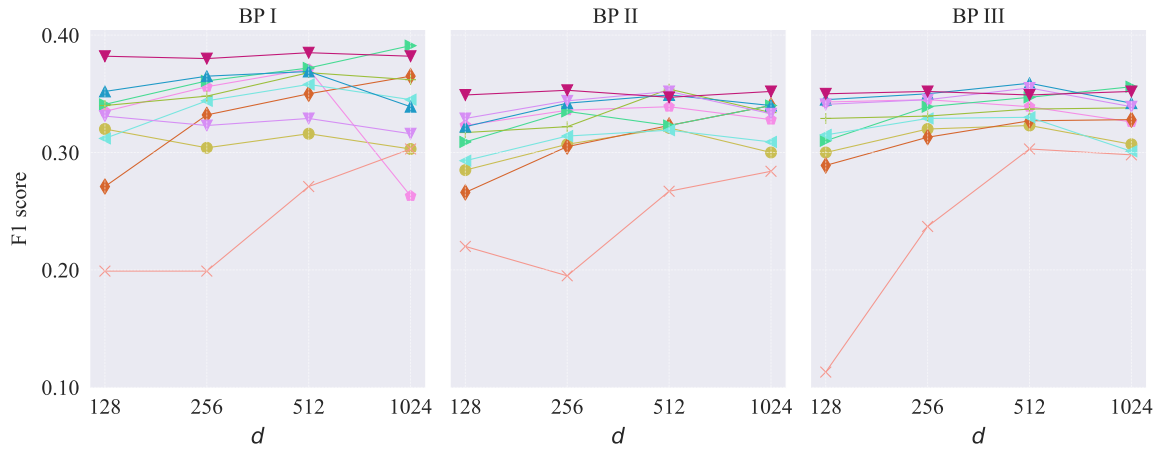
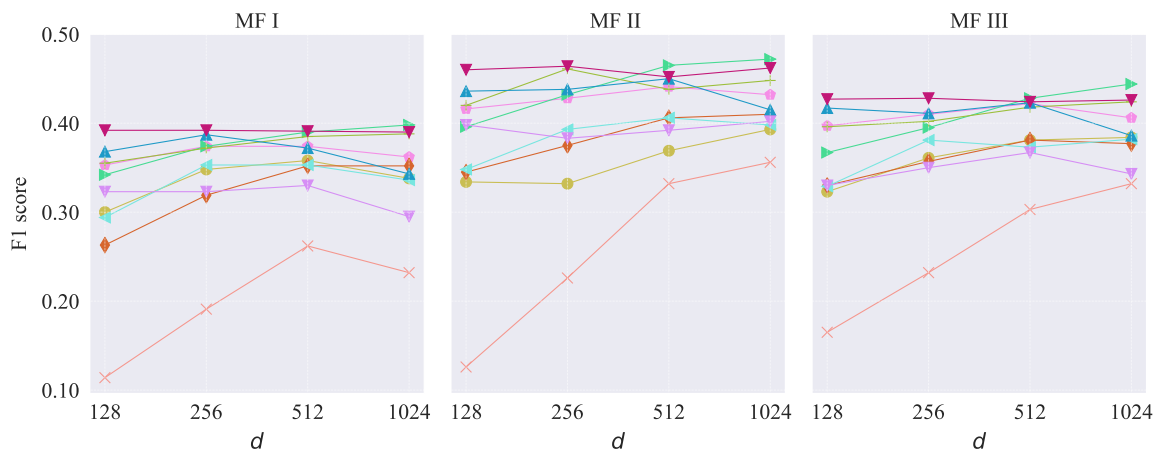


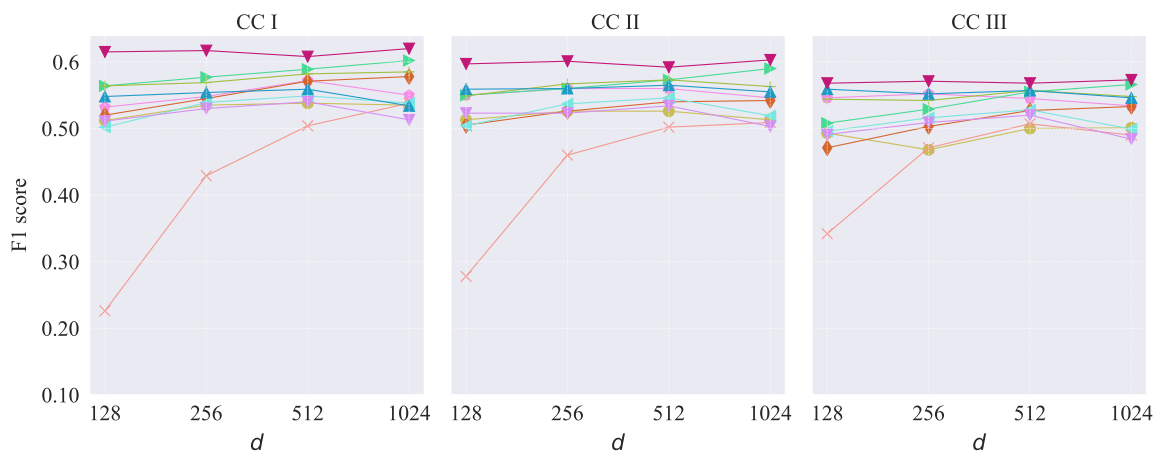
Figure S6: **Cross-validation performance.** Performance (F1 score and accuracy) of BraneMF with  $\gamma \in [0, 0.25, 0.50, 0.75, 1.0]$ ,  $w = 10$  and  $d = 128$ , in 5-fold cross validation for function prediction, compared to baseline network integration methods. Error bars represent standard deviations across 10 CV trials.



(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 S7: **Classification performance benchmark (BraneMF versus baselines)**.  $x$ -axis: protein feature dimension  $d$ ;  $y$ -axis: F1 score.



## 9 Statistical significance analysis

To determine whether the results are statistically significant compared to the baseline methods, we have applied a two sample t-test and calculated the  $p$ -values ( $P$ ). These  $p$ -values show the probability of observing an effect of the same magnitude or more extreme given that the null hypothesis is true. A hypothesis test is designed to show whether the mean of the results of BraneMF is significantly different than the mean of the baseline method. We perform statistical test on results of each level of BP, MF, and CC for the following metrics: m-AUPR, M-AUPR, F1, and ACC. We compare the statistical significance of the BraneMF results compared to individual baseline methods. The resulting  $P$  values are then grouped into 'ns', '\*', '\*\*', '\*\*\*', and '\*\*\*\*' showing the level of significance. Their details are given below:

- $P > 5.00e - 02$  : 'ns'
- $1.00e - 02 < P \leq 5.00e - 02$  : '\*'
- $1.00e - 03 < P \leq 1.00e - 02$  : '\*\*'
- $1.00e - 04 < P \leq 1.00e - 03$  : '\*\*\*'
- $P \leq 1.00e - 04$  : '\*\*\*\*'

The results are shown in Figure S8. For BP II, BP III, CC I, and CC II, BraneMF's results are statistically significant compared to all the baseline models. For BP I, BP II, MF I, MF II, except accuracy, BraneMF's results are statistically significant compared to all the baseline models. Nevertheless, accuracy can be a useful measure if we have the same amount of samples per class but if we have an imbalanced set of samples accuracy is not that useful. Therefore, remaining three metrics have more weight over accuracy. Note that, the statistical significance tests can only reveal the significance of differences in the mean score. However, to actually see if the mean scores of BraneMF are lower or higher than the baselines, please refer to Table S7, S8, and S9.

		Significance										
		****	****	****	****	****	****	****	****	****	****	****
BP I	m-AUPR	45.8727	28.7863	17.0551	20.5932	18.918	27.5783	11.1042	10.2868	28.3633	10.2273	31.4019
	M-AUPR	28.4284	20.0424	14.3986	15.5002	17.4721	17.8064	7.79667	9.83304	23.4913	6.98232	27.3795
	F1 score	39.941	33.8278	17.4975	17.1025	20.8687	21.8165	12.401	14.97	23.4645	7.32966	36.0429
	ACC	30.3883	25.3416	12.725	6.9812	15.2605	38.9018	2.06062	2.15209	31.0448	1.13853	27.5405
		SNF	Mashup	deepNF	Multi-Net	Multi-n2y	OhmNet	BraneExp	Graph2GO	MultiVERSE	BraneMF-early	BraneMF-late

		Significance										
		****	****	****	****	****	****	****	****	****	****	****
BP II	m-AUPR	41.2784	25.1538	13.7844	14.0984	6.453	26.322	8.70607	6.09842	24.2726	5.67653	28.9652
	M-AUPR	38.1189	22.447	15.1042	13.8816	4.51313	26.6708	9.54171	6.1695	24.0421	3.36914	29.2238
	F1 score	36.0687	30.2261	16.438	9.46247	6.97216	24.1308	10.0255	13.4978	22.032	4.58063	32.2659
	ACC	24.1797	22.8933	19.3065	12.3765	13.7593	39.9136	1.82673	5.14495	32.8764	4.59108	23.938
		SNF	Mashup	deepNF	Multi-Net	Multi-n2y	OhmNet	BraneExp	Graph2GO	MultiVERSE	BraneMF-early	BraneMF-late

		Significance										
		****	****	****	****	****	****	****	****	****	****	****
BP III	m-AUPR	60.0749	25.7898	18.272	10.3207	5.85672	34.1506	14.8865	5.12057	30.324	9.69087	32.359
	M-AUPR	52.4955	22.3284	17.8303	11.6771	6.77495	31.9319	10.3124	5.9053	31.1936	10.1606	32.152
	F1 score	56.0528	32.0439	22.5158	3.38881	3.32201	27.6248	12.7786	13.4438	20.5521	18.9882	40.7232
	ACC	53.7342	31.1523	23.6298	7.45511	5.83741	54.3714	6.76973	20.7582	42.4208	14.7876	25.5696
		SNF	Mashup	deepNF	Multi-Net	Multi-n2y	OhmNet	BraneExp	Graph2GO	MultiVERSE	BraneMF-early	BraneMF-late

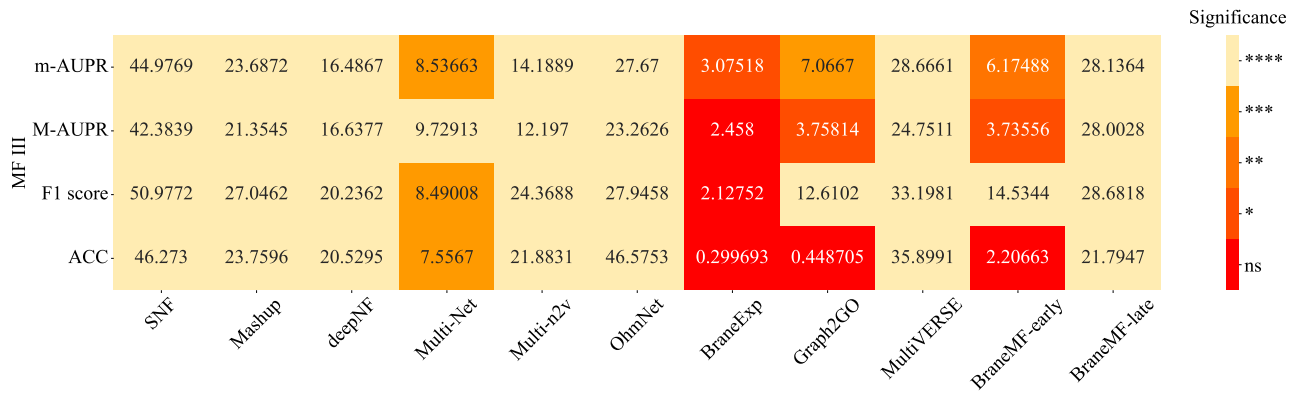
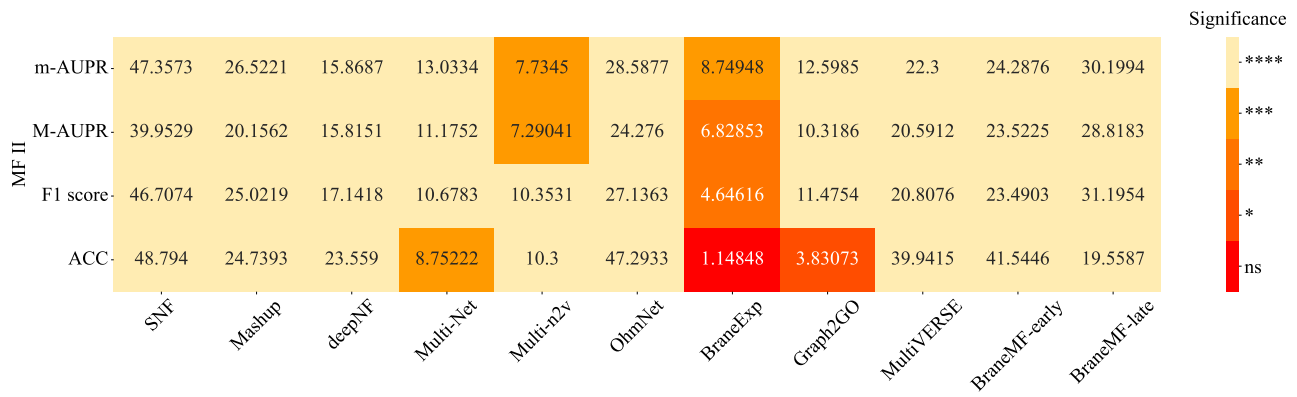
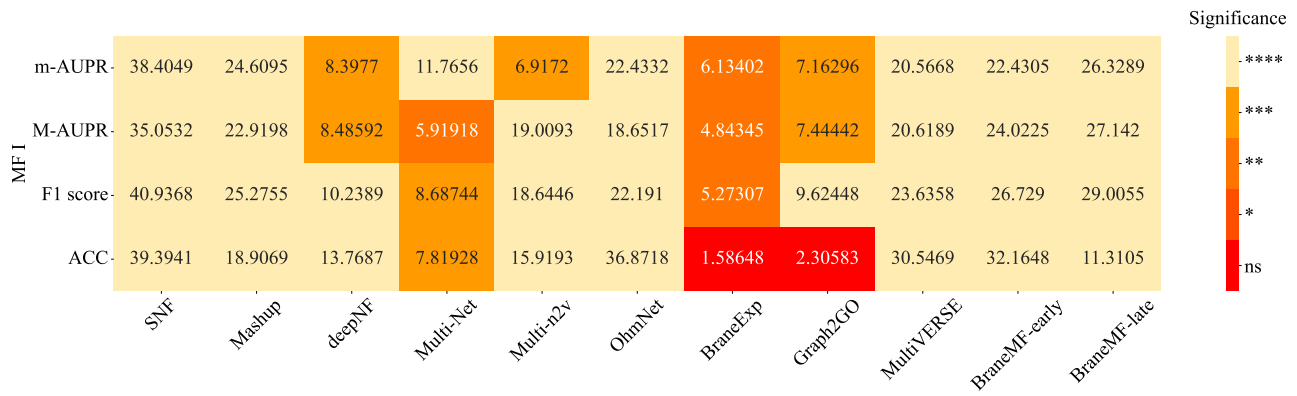




Figure S8: **Statistical significance of BraneMF compared to the baseline methods.** Each heatmap represents the statistical significance of protein function prediction for three levels of BP, MF, and CC measured by m-AUPR, M-AUPR, F1, and accuracy (ACC) scores. The values in the heatmap show the  $-\log_{10}(p\text{-value})$  computed by applying a two-sided t-test for each metric respectively. The colors in the heatmap show the level of significance as described in Section 9.

## References

- [1] Arunkumar Bagavathi and Siddharth Krishnan. Multi-Net: a scalable multiplex network embedding framework. In L. Aiello, C. Cherifi, H. Cherifi, R. Lambiotte, P. Lió, and L. Rocha, editors, *Proc. Int. Conf. Complex Netw. Appl.*, volume 813, pp. 119–131, 2018.
- [2] Davide Chicco and Giuseppe Jurman. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics*, 21(1):1–13, 2020.
- [3] Hyunghoon Cho, Bonnie Berger, and Jian Peng. Compact integration of multi-network topology for functional analysis of genes. *Cell Syst.*, 3(6):540–548, 2016.
- [4] Gene Ontology Consortium. The Gene Ontology (GO) database and informatics resource. *Nucleic acids research*, 32(suppl\_1):D258–D261, 2004.
- [5] Kunjie Fan, Yuanfang Guan, and Yan Zhang. Graph2GO: a multi-modal attributed network embedding method for inferring protein functions. *GigaScience*, 9(8), Aug. 2020.
- [6] Vladimir Gligorijević, Meet Barot, and Richard Bonneau. deepNF: deep network fusion for protein function prediction. *Bioinformatics*, 34(22):3873–3881, 2018.
- [7] Rafael A. Irizarry, Chi Wang, Yun Zhou, and Terence P. Speed. Gene set enrichment analysis made simple. *Statistical methods in medical research*, 18(6):565–575, 2009.
- [8] Surabhi Jagtap, Abdulkadir Çelikkanat, Aurélie Pirayre, Frederique Bidard, Laurent Duval, and Fragkiskos D. Malliaros. Multiomics data integration for gene regulatory network inference with exponential family embeddings. In *Proc. Eur. Sig. Image Proc. Conf.*, pp. 1221–1225, 2021.
- [9] T. N. Kipf and M. Welling. Variational graph auto-encoders. In *Bayesian Deep Learning Workshop (NIPS 2016)*, 2016.
- [10] 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, Michael G. McDermott, Caroline D. Monteiro, Gregory W. Gundersen, and Avi Ma’ayan. Enrichr: a comprehensive gene set enrichment analysis web server 2016 update. *Nucleic Acids Res.*, 44:W90–W97, 2016.
- [11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, and J. Vanderplas. scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2011.
- [12] Léo Pio-Lopez, Alberto Valdeolivas, Laurent Tichit, Élisabeth Remy, and Anaïs Baudot. MultiVERSE: a multiplex and multiplex-heterogeneous network embedding approach. *Sci. Rep.*, 11(1):8794, Apr 2021.
- [13] 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 *Proc. ACM Int. Conf. Web Search Data Min.*, pp. 459–467, 2018.
- [14] Paul Shannon, Andrew Markiel, Owen Ozier, Nitin S. Baliga, Jonathan T. Wang, Daniel Ramage, Nada Amin, Benno Schwikowski, and Trey Ideker. Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome research*, 13(11):2498–2504, 2003.
- [15] 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. *Proc. Nat. Acad. Sci. U.S.A.*, 102(43):15545–15550, 2005.
- [16] Damian Szklarczyk, Annika L. Gable, Katerina C. Nastou, David Lyon, Rebecca Kirsch, Sampo Pyysalo, Nadezhda T. Doncheva, Marc Legeay, Tao Fang, Peer Bork, Lars J. Jensen, and Christian von Mering. The STRING database in 2021: customizable protein–protein networks, and functional characterization of user-uploaded gene/measurement sets. *Nucleic Acids Res.*, 49(D1):D605–D612, 2021.
- [17] 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. *Nat. Meth.*, 11(3):333–340, 2014.

- [18] James D. Wilson, Melanie Baybay, Rishi Sankar, and Paul Stillman. Fast embedding of multilayer networks: An algorithm and application to group fMRI. *PREPRINT*, 2018.
- [19] Zhihua Zhang and Jianzhi Zhang. A big world inside small-world networks. *PloS One*, 4(5):e5686, 2009.
- [20] Marinka Zitnik and Jure Leskovec. Predicting multicellular function through multi-layer tissue networks. *Bioinformatics*, 33(14):i190–i198, 2017.