A Supervised Co-complex Probability Weighting of Yeast Composite Protein Networks using Gradient-boosted Trees for Protein Complex Detection

Anthony Van C. Cayetano and John Justine S. Villar

Scientific Computing Laboratory
Department of Computer Science
University of the Philippines
1101 Diliman, Quezon City, Philippines
{accayetano4,jsvillar1}@up.edu.ph

Abstract. Many studies in the past have proposed various methods to detect protein complexes from protein-protein interaction networks (PPINs) by applying clustering algorithms to the network, relying only on the topology of the PPIN. However, PPINs have a high number of false positives and false negatives, making them unreliable when used alone to detect protein complexes. Moreover, not all proteins in a protein complex interact with each other and not all proteins that interact with each other are from the same complex. Thus, relying alone on the physical interactions of proteins is not ideal for detecting protein complexes.
This study extends the idea of a method by Yong et al. called SWC, where they integrated other heterogeneous data sources into the PPIN to create a composite network and where each edge is weighted according to its posterior co-complex probability. SWC, when combined with various clustering algorithms, resulted in more accurate results in detecting protein complexes.
This study attempts to improve SWC by integrating additional data sources and by using a more advanced machine learning model called *gradient-boosted trees*. The proposed method outperformed SWC in every performance metric, often by a considerable margin in terms of precision-recall AUC, Brier score loss, and log loss when predicting co-complex edges. More importantly, it also outperformed SWC in terms of precision-recall AUC when used together with the Markov Cluster algorithm (MCL) to detect protein complexes. Lastly, it also outperformed various unsupervised weighting methods in all the said performance evaluations. These evaluations were performed on two yeast PPINs.

Keywords: protein complex prediction, graph clustering, PPIN

1 Background

Proteins are large molecules that are involved in many important functions and cellular processes in our body. However, proteins very rarely act alone and in-

J. Caro et al. (eds.), *Proceedings of the Workshop on Computation: Theory and Practice (WCTP 2023)*, Atlantis Highlights in Computer Sciences 20,
https://doi.org/10.2991/978-94-6463-388-7_21

stead, they interact with other proteins to perform their functions. The physical interaction between two proteins is called a *protein-protein interaction* (PPI). The collection of PPIs within an organism forms a large network called *protein-protein interaction network* (PPIN).

Within this PPIN are subgraphs or clusters of PPIs called *protein complexes*. Protein complexes are large multimolecular units, comprising proteins interacting at the same time and place [38], that perform a specific cellular process. Thus, identifying these protein complexes would help us understand the biological processes within our cells.

Several developments in high-throughput experimental methods (e.g. TAP-MS [33] and Y2H [48]) have resulted in the rise of genome-wide PPINs (interactomes). The availability of these interactomes has encouraged researchers to data mine them in order to computationally detect (predict) protein complexes. Most of the protein complex detection methods developed so far transform the problem of protein complex detection into a graph clustering problem. More specifically, a PPIN is first represented as a large graph or network of proteins, where the proteins are the vertices and their interactions are the edges. Then, these complex detection methods cluster the said network into subgraphs, which correspond to the predicted protein complexes.

However, due to the high-throughput nature of these experimental methods, PPINs contain a non-negligible amount of noise [13, 18], which ultimately affects the performance of complex detection methods. This signifies the need for integrating additional biological insights into these methods to increase their accuracy.

2 Preliminaries

2.1 Graph Clustering

A PPIN can be represented as a graph $G = (V, E, w)$, where V is the set of vertices corresponding to the proteins, $E \subseteq V \times V$ is the set of edges corresponding to the PPIs, and $w : E \to \mathbb{R}_{\geq 0}$ is the edge weight of the PPI. Note that $w(v_i, v_j) = 0$ if $(v_i, v_j) \notin E$. Moreover, G can be represented as an adjacency matrix A such that $A_{ij} = w(v_i, v_j)$.

These graph representations of PPINs allow us to perform graph clustering algorithms to the network, which output clusters or subgraphs corresponding to the detected protein complexes.

Markov Cluster Algorithm The Markov Cluster (MCL) algorithm [14, 44] is one of the many graph clustering algorithms frequently applied in the context of bioinformatics [15]. In fact, several studies have extended and modified the algorithm, particularly for protein complex detection [4, 36, 39].

Given an undirected graph G and its corresponding adjacency matrix A, the original MCL algorithm *roughly* works based on the following steps.

1. First, matrix A is converted to a column-stochastic matrix M where each of its column sums to 1. This is done by normalizing the columns of A. Note that the entry M_{ij} is the probability of walking from node j to node i (transition probability).
2. The *expansion* operation is performed by matrix squaring M (i.e. $M_{expand} = M \cdot M$). This assigns new transition probabilities to all the existing edges by expanding the reach of each node to other nodes in the graph.
3. The *inflation* operation is performed by taking the r^{th} power of each entry of the matrix M_{expand}, then normalizing the columns of the resulting matrix so that the resulting matrix is column-stochastic. This has the effect of strengthening walks within a cluster and weakening walks between two different clusters.
4. Repeat the process of *expansion* and *inflation* until a steady state is achieved.
5. The converged state of the column-stochastic matrix can be interpreted as the clusters of the graph G.

In summary, the whole process is essentially a simulation of random walks on the graph G. Moreover, the MCL algorithm has an inflation parameter that can be set by the user to control the granularity of the resulting clusters.

2.2 Weighting Methods

Note that while the MCL algorithm can work on unweighted graphs, it has been reported that the algorithm predicts more accurate clusters when used on weighted protein networks. This is due to the high amount of noise in PPINs [13, 18], which significantly affects the performance of protein complex detection methods that rely only on the topology of the protein network. Thus, several weighting methods have been developed in the past to mitigate this issue.

2.3 Local Topology of Protein Pairs

Local-topology-based weighting schemes assume that protein pairs that share a high number of common neighbors are more likely to interact with each other. This assumption was supported by the findings of Chua et al. [10], where they found that the majority ($\sim$ 70%) of protein pairs that are level-1 or level-2 neighbors of each other share the same functions, and are thus more likely to interact with each other.

Several weighting schemes based on common neighbors have been used and developed. Among these are CD-Distance [7] and FS-Weight [10]. Despite their simplicity, topology-based weighting schemes are effective in terms of improving the performance of protein complex detection algorithms. For instance, a study by Beltran et al. [4] has shown that using FS-Weight as a pre-processing step significantly improves the performance of plain (unweighted) MLR-MCL with balance [36] (a variant of MCL).

Liu et al. [24] further improved the performance of these topology-based weighting schemes by proposing an iterative topological weighting method. The

formula of this is shown below.

$$w^{(k)}(u,v) = \frac{\sum_{x \in N_u \cap N_v} w^{(k-1)}(x,u) + \sum_{x \in N_u \cap N_v} w^{(k-1)}(x,v)}{\sum_{x \in N_u} w^{(k-1)}(x,u) + \sum_{x \in N_v} w^{(k-1)}(x,v) + \lambda_u^{(k)} + \lambda_v^{(k)}} \quad (1)$$

where $w^0(x,u) = 1$ if $(x,u) \in E$; otherwise, $w^0(x,u) = 0$; and the neighbor set $N_u = \{v|(u,v) \in E\}$. The $\lambda_u^{(k)}$ and $\lambda_v^{(k)}$ are the penalty terms used to penalize proteins with very few immediate neighbors. In particular, they are defined as follows.

$$\lambda_u^{(k)} = \max\left\{0, \frac{\sum_{x \in V} \sum_{v \in N_x} w^{(k-1)}(v,x)}{|V|} - \sum_{v \in N_u} w^{(k-1)}(v,u)\right\} \quad (2)$$

The authors remarked that this iterative scoring method achieves the best performance on $k = 2$ and that increasing the number of iterations further does not improve its performance significantly.

In their other study, they used the said iterative weighting approach (called iterative AdjustCD), together with their protein complex prediction algorithm called CMC [25]. The results of this study have shown that iterative AdjustCD significantly improves the performance of CMC in terms of predicting protein complexes compared to the non-iterative version.

Experimental Reliability Weighting schemes based on experimental reproducibility assume that PPIs reported in multiple independent experiments are more reliable than those that are reported only once. One such method under this category is MV scoring [22], which takes into account both experimental reproducibility (number of experiments) and experimental plurality (throughput). It has the formula:

$$MV(u,v) = N_e^a \sum_{i=1}^{N_e} \frac{1}{plurality(i)} \quad (3)$$

where N_e is the number of independent experiments that report the PPI (u,v) and $plurality(i)$ is the number of reported PPIs of experiment i. Note that a is a parameter that dictates how much weight should be given to the experimental reproducibility factor (i.e. N_e) versus the experimental plurality factor ($plurality(i)$). In the said study [22], the optimal value of a was experimentally determined to be $a = 2$ (hence, this value was the one used in this study).

The idea behind MV scoring is that PPIs reported in multiple experiments (i.e. those with $N_e > 1$), as well as those that are reported in low-throughput experiments (i.e. those with smaller values of $plurality(i)$), are given higher scores. It is assumed here that PPIs reported in low-throughput (low plurality) experiments are more reliable than those that are reported in high-throughput ones.

In the said study, MV scoring was applied to four protein complex detection algorithms and was shown to generally improve their performance.

Gene Co-expression The idea behind gene co-expression weighting methods is that a protein pair whose genes are highly co-expressed is more likely to have an interaction than random protein pairs [5, 16] and are more likely to be functionally associated [46].

One simple, yet popular method to measure the co-expression of two proteins is by using the Pearson correlation coefficient (PCC) on the gene expressions of proteins.

For instance, a study by Yu and Kong [49] used PCC to weight protein networks based on gene expression correlation. This weighting method was used as a preprocessing step in their protein complex detection method. Given proteins $u, v \in V$ with gene expressions $u_{gene\_exp} = \{x_i\}$ and $v_{gene\_exp} = \{y_i\}$ for $i = 1, 2, 3, \ldots, n$ at n time points, the PCC of the gene expression profiles of proteins u and v is

$$PCC(u, v) = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2 \sum_{i=1}^{n}(y_i - \overline{y})^2}} \tag{4}$$

where $\overline{x}$ and $\overline{y}$ are the mean gene expressions of u and v, respectively. In the said study, negatively correlated protein pairs are removed from the network. Moreover, the gene expression data used in the study is GSE3431 [43], which contains 36 time points of gene expression profiles of yeast over three successive metabolic cycles (12 time points per cycle).

Gene Ontology Semantic Similarity The Gene Ontology Consortium (GOC) is a collaborative work that aims to provide comprehensive resources describing the properties and functions of gene products (e.g. proteins) [2, 11]. It offers two primary knowledgebases:

1. **Gene Ontology (GO)**. The GO is a vocabulary of *GO terms* describing a specific property of a gene product in terms of three domains: *cellular component* (CC), *biological process* (BP), and *molecular function* (MF). It is organized into a directed acyclic graph (DAG), where each node is a GO term and each edge corresponds to a relationship (e.g. *is a, part of, has part*) between two GO terms.
2. **GO Annotations**. A GO annotation is an association of a particular gene product with a particular GO term. Several genomes of various organisms have already been annotated with GO terms and are widely available (e.g. the GO annotations for the *Saccharomyces cerevisiae* provided by Saccharomyces Genome Database (SGD) [9]).

A number of GO-based weighting schemes have been proposed over the years, most of which are based on the concept of semantic similarity measures (SSMs). In the context of GO, an SSM scores the similarity between two proteins by getting the semantic similarity of the GO annotations of the two proteins.

One example of an SSM is the Topological Clustering Semantic Similarity (TCSS) algorithm [19]. Because the GO is incomplete [2], the GO DAG is said

to be unbalanced in the sense that some paths of the graph have little detail or depth compared to others. TCSS takes into account this unbalanced nature of GO DAG by clustering it into disjoint subgraphs. A protein pair belonging to the same subgraph would have a higher score compared to a protein pair with different subgraphs.

It is important to note that TCSS produces three scores for each protein pair, one for each of the three domains of GO (i.e. cellular component (CC), biological process (BP), and molecular function (MF)). In other words, for a protein pair (u, v), its semantic similarity is

$$TCSS_{CC}(u, v) = x \tag{5}$$

$$TCSS_{BP}(u, v) = y \tag{6}$$

$$TCSS_{MF}(u, v) = z \tag{7}$$

where x, y, z are the SSMs of (u, v) in terms of CC, BP, and MF.

The authors of TCSS then compared the algorithm with other SSMs where they evaluated the SSMs in terms of their ability to distinguish positive and negative PPIs, as well as their correlation with gene expression and protein families. Results generally and significantly favored TCSS over the other SSMs.

Supervised Weighting Methods Aside from the discussed unsupervised weighting methods, many supervised weighting methods that integrate multiple data sources or features together were also developed in the past.

However, some of these methods do not primarily focus on protein complex detection. In particular, some of them focuses only on predicting interactions and/or co-complex relationship between two proteins without applying in the context of protein complex detection [1, 20, 31].

One supervised weighting method that primarily focuses on protein complex detection is SWC [47], which is the main inspiration for this study. In the said study, they integrated other heterogeneous data sources into the PPIN to create a composite protein network. These data sources or features are:

- Topological weights of a combined PPIN which is weighted via iterative AdjustCD [25]. The combined PPIN is the combination of all the PPIs from BioGRID [40], IntAct [21], and MINT [23] (these databases were downloaded in November 2011). Note that both direct (level 1) and indirect (level 2) interactions were topologically weighted.
- Functional association scores from the STRING [42] database (downloaded in January 2012), where only the scores greater than 0.5 were kept.
- Co-occurrence of protein pairs in the PubMed literature (downloaded in January 2012), which is based on the formula:

$$\text{CO\_OCCUR}(u, v) = \frac{|A_u \cap A_v|}{|A_u \cup A_v|} \tag{8}$$

where $u, v \in V$ are proteins and A_u, A_v are the sets of PubMed publications that contain u, v respectively.

Note that SWC had a total of four features since direct and indirect topological weights were treated as separate features. For brevity, we will refer to the four features as TOPO, TOPO_L2, STRING, and CO_OCCUR throughout this paper (they were called PPI, L2-PPI, STRING, and PUBMED, respectively in the paper of SWC).

These four features are combined together to create a composite protein network. Then, each edge in the network is weighted using a naive Bayes model based on its posterior probability of belonging to the same complex. Using this weighted network, six graph clustering algorithms were used to predict protein complexes which include MCL [14, 44].

The authors compared SWC with another supervised weighting method that uses LogitBoost [45], as well as other unsupervised weighting methods (iterative AdjustCD and the STRING database). The results generally favored SWC on both yeast and human composite protein networks.

The authors of SWC emphasized the issues that come along when only the PPI data is used to predict protein complexes. Aside from the fact that PPINs have a huge amount of noise, they remarked that not all proteins in a protein complex interact with each other and not all proteins that interact with each other are from the same complex. Thus, relying only on the topology of the PPIN is not ideal for detecting protein complexes. The authors further emphasized that SWC solves the aforementioned issues by enriching the original PPIN so that new edges are added for protein pairs belonging to the same complex but do not necessarily physically interact with each other.

3 Objectives of the Study

The objective of this study is to develop a weighting method for protein complex detection, based on the idea of co-complex probability weighting as was done in SWC. More specifically, the objectives are

1. to determine what additional biological data sources can be added to extend SWC features to increase its performance;
2. to use a more advanced machine learning model for more accurate integration of data sources; and
3. to analyze the individual importances of the biological data sources used.

4 Datasets

The following datasets were used in this study.

- The yeast composite protein network used in SWC [47] was used in this study. This contains TOPO, TOPO_L2, STRING, and CO_OCCUR features as discussed in Section 2.3.
- As a secondary protein network, the Database of Interacting Proteins [35] PPIN was also used (February 2017 version). Note that, unlike the previous protein network, the DIP PPIN is much smaller and more recent.

- For the gene expression data, GSE3431 [43] was used, which was downloaded from the Saccharomyces Genome Database [9] (SGD) website. This dataset contains 36 time points of gene expression profiles of yeast.
- For the Gene Ontology (GO) data and annotations, the data were downloaded from the Gene Ontology website. In particular, the October 31, 2011 version of GO was used, while the October 29, 2011 version of yeast GO annotations was used. These versions were particularly selected in order to ensure fairness since the external data sources provided by SWC (i.e. STRING and CO_OCCUR) are late-2011 to early-2012 data.
- The database iRefIndex [32] (version 19.0, release date: August 22, 2022) was also used in this study. This database is a combination of various protein interaction databases, which are BIND [3], BioGRID [41], CORUM [34], DIP [35], HPRD [29], IntAct [26], MINT [23], MPact [17], MPPI [27], and OPHID [6]. While the version of this data is very recent, only the publication entries before 2012 were selected from this database, for the same reason stated in the previous item (i.e. to ensure fairness).
- The CYC2008 dataset [30] was used as the gold standard protein complex dataset. This dataset contains 408 verified *Saccharomyces cerevisiae* protein complexes backed by highly reliable small-scale studies.

5 Building the Composite Network

In this study, two yeast PPINs were used. First is the "Combined PPIN" used in SWC, which was obtained by combining BioGRID [40], IntAct [21], and MINT [23]. Second is a more recent and much smaller PPIN, which is the DIP [35] PPIN. The UniProt [12] Retrieve/ID mapping service was used to map each UniProtKB AC/ID in the DIP PPIN to its corresponding systematic name.

Both of these PPINs were filtered such that protein pairs with no common neighbors were removed from the network. This process resulted in the Combined PPIN having a total of 106,328 direct (level 1) PPIs and the DIP PPIN having a total of direct 12,509 PPIs.

Using the features that will be discussed in the next sections, two composite protein networks were derived from these two PPINs. For brevity, let's call the first one derived from the Combined PPIN as the "Original" composite network, and the second one derived from the DIP PPIN as the "DIP" composite network.

5.1 Using the Features of SWC

Given an unweighted PPIN $G_{ppi} = (V_{ppi}, E_{ppi})$, a *base* composite network $G_{base} = (V, E, F_{base})$ is built using the four features of SWC (i.e. TOPO, TOPO_L2, STRING, and CO_OCCUR). Note that $V_{ppi} \subseteq V$ and $E_{ppi} \subseteq E$ since new edges that are not present in the PPIN are added to the composite network based on the features. Moreover, $F_{base} = \{\text{TOPO, TOPO\_L2, STRING, CO\_OCCUR}\}$ are the four features of SWC. Each feature $F \in F_{base}$ maps each edge $(u, v) \in E$

to a score $s \geq 0$ depending on the association of the edge on that feature. If there is no association between u and v on a certain feature, then their score is set to 0 on that feature. In other words,

$$F(u,v) = \begin{cases} 0, & \text{if } (u,v) \in E \text{ is not related at feature } F \\ s, & \text{otherwise} \end{cases} \tag{9}$$

For instance, we can express a direct PPI (u, v) whose topological weight (based on iterative AdjustCD [25]) is 0.5 as $\text{TOPO}(u, v) = 0.5$. The same notation goes with all the other features.

The feature scores for TOPO, TOPO_L2, STRING, and CO_OCCUR were already provided by the authors of SWC for the "Original" composite network. The "DIP" composite network, however, needed to be topologically re-weighted to get its TOPO and TOPO_L2 scores. The process of getting TOPO and TOPO_L2 scores for the DIP composite network is the same as the one done in SWC.

5.2 Extending the Features of SWC

In this study, additional features were added, namely: REL, CO_EXP, GO_CC, GO_BP, and GO_MF. For brevity, let

$$F_{new} = \{ \text{ REL, CO\_EXP, GO\_CC, GO\_BP, GO\_MF } \} \tag{10}$$

REL. This feature is based on experimental reliability, in particular, MV Scoring [22]. This feature uses the iRefIndex [32] database to retrieve the experimental reproducibility of each PPI and the experimental plurality of each experiment.

Since MV Scoring is unbounded while the rest of the features are normalized, additional modifications were done on the MV Scoring to normalize it to the scale between 0 and 1.

Let $X = \ln MV(u, v)$ and $\overline{X}$ and SD be the mean and standard deviation of X. Let the standardized, log-transformed MV scoring be

$$MV_{new}(u,v) = \frac{X - \overline{X}}{SD} \tag{11}$$

Then, MV_{new} is bounded such that any value that exceeds the third standard deviation is set to the third standard deviation. Let this operation be $Bound(MV_{new})$. After bounding, $Bound(MV_{new})$ is then normalized. Let this operation be $Norm(Bound(MV_{new}))$. The result of the post-processes described above is the REL feature,

$$REL(u,v) = Norm(Bound(MV_{new})) \tag{12}$$

CO_EXP. Using GSE3431 [43] gene expression data, protein pairs are weighted via PCC,

$$CO\_EXP(u,v) = PCC(u,v) \tag{13}$$

GO_CC, GO_BP, and GO_MF. These features are derived from the output SSMs of TCSS [19] using the GO and GO annotation data described in Section 4.

$$GO\_CC(u,v) = TCSS_{CC}(u,v) \tag{14}$$
$$GO\_BP(u,v) = TCSS_{BP}(u,v) \tag{15}$$
$$GO\_MF(u,v) = TCSS_{MF}(u,v) \tag{16}$$

Thus, the full list of features used in this study is

$$F_{full} = F_{base} \cup F_{new} \tag{17}$$

Using all these features, the full composite network G is constructed from the base composite network G_{base}. This is done by adding the new features to G_{base}, that is, each edge in G_{base} was scored according to each feature in F_{new}. In other words, $G = (V, E, F_{full})$. Note that no new edges were added to the base composite network G_{base} (i.e. G and G_{base} have the same vertices and edges) since only additional feature scores of existing edges were added.

The composite network can be visualized in tabular form, as is shown in Table 1.

PROTEIN U	PROTEIN V	TOPO	TOPO_L2	STRING	...
YDR098C	YDR130C	0.67	0.98	0.72	...
YDR098C	YHR069C	0.40	0.00	0.12	...
YDR130C	YHR069C	0.51	0.32	0.99	...
⋮	⋮	⋮	⋮	⋮	⋱

Table 1. An example of a composite protein network.

Table 2 shows the summary of the two composite networks. The second column is the PPIN from which the composite network was built. The third column is the number of direct (level 1) PPIs after filtering out protein pairs with no common neighbors from the PPIN. The fourth column is the number of edges of the composite network after integrating all the features.

Composite Network	Source PPIN	Num. of PPIs	Num. of edges
Original	Combined PPIN	106,328	531,800
DIP	DIP PPIN	12,509	349,795

Table 2. Summarizing statistics of the two composite networks.

6 Weighting the Composite Network

The composite network G can now be weighted using all the features. In particular, each edge in G is weighted according to its co-complex probability, which is the same idea used in SWC [47].

However, this study used a different model from the model used in SWC, which is naive Bayes. The problem with naive Bayes is its reliance on the assumption that the features are independent, which makes the probability estimates of this model frequently inaccurate.

Hence, this study proposes the use of another machine learning model called *gradient-boosted decision trees* (GBDT), more specifically, using XGBoost [8]. The main reason for choosing this model is based on the fact that XGBoost was shown to dominate and even outperform various state-of-the-art deep learning models in tabular datasets [37], making this model particularly fitting to this study since composite networks are tabular in form.

For brevity, let's call the weighting method proposed in this study as XGW.

6.1 XGBoost

XGBoost [8] is a software library implementation of gradient boosting. Gradient boosting works by creating a series of "weak models" (usually classification and regression trees (CART)), with each successive model built on top of the errors of the previous model. This series of weak models are then added together to make a final prediction. Mathematically, this can be written as [8]

$$\hat{y}_i = \sum_{k=1}^{K} f_k(\mathbf{x_i}), f_k \in \mathcal{F} \tag{18}$$

where K is the number of trees, f_k is a tree (i.e. one of the "weak models"), and $\mathbf{x_i}$ and $\hat{y}_i$ are the features and predicted value of sample i, respectively.

One important difference between XGBoost and traditional GBDTs is the regularized objective function, which allows us to control overfitting. In particular, the objective function to be minimized [8] is

$$\mathcal{L}(\phi) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k) \tag{19}$$

$$\text{where } \Omega(f) = \gamma T + \frac{1}{2}\lambda w^2 \tag{20}$$

where l is any differentiable loss function and Ω (which depends on γ and λ) is the regularization term to control overfitting.

In this study, XGBoost was used for binary classification, that is, it attempts to classify each edge in the network whether it is a co-complex edge or a non-co-complex edge. In particular, XGBoost was used to predict the co-complex probability of an edge (u, v) by setting the loss function l to be the loss function used in binary logistic regression (i.e. log loss, see Equation 22). This allows XGBoost to output class probabilities, not labels. The predicted co-complex probability of an edge is then set as the final weight of that edge.

6.2 Training

Let $G = (V, E, F_{full})$ be the composite protein network. Let $T = \{T_1, T_2, \ldots, T_n\}$ be the set of n training protein complexes selected from CYC2008 [30] and let T_{prot} be the set of all proteins in T. A subset D of E is used as the training samples for the XGBoost classifier, where $D = \{(u, v) \mid (u, v) \in E \wedge u, v \in T_{prot}\}$. Each edge $(u, v) \in D$ is labeled as 1 if it is a co-complex and 0 if it is non-co-complex, according to the set of training complexes. In other words, each edge $(u, v) \in D$ is labeled as 1 if $\exists T_i \in T : u, v \in T_i$ and 0 otherwise.

Using these labeled samples and their corresponding feature scores F_{full}, XGBoost learns its parameters which are the trees f_i. After learning the parameters, each edge $(u, v) \in E$ is weighted according to its probability of belonging to class label 1 (co-complex probability).

Note that while XGW and SWC have exactly the same set of training complexes (and hence, the same *positive* training samples), they differ in how they generate their set of *negative* training samples (negative training samples need to be generated since the set of training complexes only provide positive training samples, i.e. the co-complex edges).

During the training phase, SWC labels all the edges of the entire network (E), while XGW labels only the edges in $D \subseteq E$. What this means is that SWC treats all the other edges aside from the co-complex training pairs as non-co-complex edges, which results in mislabeling some edges since some of them are in reality, co-complex edges (but which are not part of the training complexes). The labeling approach used in XGW mitigates this issue by labeling only a subset of the entire network, in particular, labeling only the edges whose both proteins can be found in T_{prot}. In other words, XGW assumes that a protein pair $(u, v) \in T_{prot} \times T_{prot}$ that is not a co-complex pair in T, is a true non-co-complex pair. Experiments have shown that using this labeling approach is a lot more effective than the labeling approach used in the SWC study.

7 Protein Complex Detection

After weighting, the MCL algorithm is run on two versions of the weighted composite protein network. The first version is the protein network with all the edges present. The second version is a filtered version where only the top 20,000 edges were selected (as was done in the SWC study).

This whole process then produces two sets of predicted clusters per weighted composite network (one using all the edges, another using only 20,000 edges). Each predicted cluster C was then scored according to the following cluster density formula adopted from SWC:

$$dens(C) = \frac{\sum_{u \in C, v \in C} w(u, v)}{|C|(|C| - 1)} \tag{21}$$

These cluster density scores were then used in performance evaluation, which will be discussed in the next section.

Moreover, four different inflation parameters were used in running the MCL algorithm. These are $I = 2, 3, 4, 5$.

8 Performance Evaluation Setup

The performance of the two supervised weighting methods (XGW and SWC) was compared together with all the unsupervised weighting methods used (the nine features themselves).

Moreover, another set of unsupervised weighting methods (called *super features*, for brevity) were introduced. These super features are essentially hand-picked combinations of two or more features, which are averaged together.

8.1 Super features

The following is the list of super features used in this study.

1. *ALL.* This is the mean of all the nine features.
2. *GO_SS.* This is the mean of all the GO SSMs (i.e. GO_CC, GO_BP, and GO_MF).
3. *TOPOS.* This is the mean of TOPO and TOPO_L2.
4. *ASSOC.* This is the mean of STRING, CO_OCCUR, REL, and CO_EXP.
5. *TOPO_GO.* This is the mean of TOPO, GO_CC, GO_BP, and GO_MF.
6. *TOPO_CO_EXP.* This is the mean of TOPO and CO_EXP.
7. *TOPO_GO_CO_EXP.* This is the mean of TOPO, GO_CC, GO_BP, GO_MF, and CO_EXP.

Essentially, the protein network is weighted according to these super features as well, aside from the aforementioned supervised and unsupervised weighting methods.

Thus, in total, there are 19 weighting methods that were evaluated (two supervised methods, nine features, seven super features, and one unweighted method). Note that the unweighted method is essentially just the (filtered) PPIN (that is, PPIs with no common neighbors are removed) where all its scores are set to 1.

8.2 Cross-validation

Performance evaluations were done on 10 rounds of cross-validation. In each round, 134 (that is, 90% out of 149) protein complexes of size greater than three are selected from CYC2008 as the set of testing complexes. The rest of the unselected complexes ($n = 274$ complexes) were used for training. Note that only 15 of the 274 training complexes are large (size greater than three), while the rest are small (size is either two or three proteins). For XGW, training is done using all the features F_{full}, while for SWC, only on F_{base}.

In each round, the following performance evaluations were conducted

- Classification of co-complex edges.
- Prediction of protein complexes when combined with the MCL algorithm.

8.3 XGBoost Hyperparameters

A hyperparameter search in each cross-validation round using the training samples was also performed with the following values

- `max_depth`: {3, 4}
- `gamma`: {0, 0.5}
- `lambda`: {50, 100}
- `subsample`: {0.6, 0.8}
- `colsample_bytree`: {0.6, 0.8}

Tuning was done via five-fold cross-validation grid searching on the training set using scikit-learn [28]. Moreover, the number of boosting rounds and the learning rate were set to 1000 and 0.01, respectively, while the objective function was set to `binary:logistic` so that the model outputs class probabilities.

These parameters were selected to make the model as conservative as possible (that is, to avoid overfitting). For instance, the lambda (λ) regularization parameter was set to a high value. This is because most of the training complexes are small while all of the testing complexes are large, which means that the training and testing datasets are vastly different. Strongly regularizing the model will help prevent it from overfitting to small complexes in order to better predict large complexes.

9 Predicting Co-complex Edges

For co-complex edge classification, three performance evaluation metrics were computed for each of the 19 weighting methods, which were averaged over the 10 rounds of cross-validation. To eliminate the bias of supervised methods in classifying training co-complex edges well, training co-complex edges were not included in the calculation of the following metrics.

- Log loss is a measure quantifying the difference between the actual class of a sample and its predicted probability. It is defined as

$$L_{\log}(y, p) = -(y \log(p) + (1 - y) \log(1 - p)) \tag{22}$$

  where $y \in \{0, 1\}$ is the true label and $p = \Pr(y = 1)$ is the probability estimate.
  Here, the lower the log loss, the better the performance of the method.
- Brier score loss is similar to log loss in that it measures the mean squared difference between the actual class of a sample and its predicted probability as well. It is defined as

$$BS = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} (y_i - p_i)^2 \tag{23}$$

  Here, the lower the Brier score loss, the better the performance of the method.

– The precision-recall area under the curve (PR AUC) is a summarizing statistic of the precision-recall curve, which measures the tradeoff between precision and recall at different thresholds.
Here, the higher the PR AUC, the better the performance of the method.

9.1 Log loss and Brier score loss

Figure 1 shows the top 10 weighting methods in terms of log loss and Brier score loss for both the composite protein networks. As expected, the two supervised weighting methods topped the performance evaluation. More importantly, XGW significantly outperformed all the other weighting methods in both composite networks, including SWC. Another thing to note is that all of the top unsupervised weighting methods were super features, indicating that a simple average of multiple features is effective for co-complex edge classification.

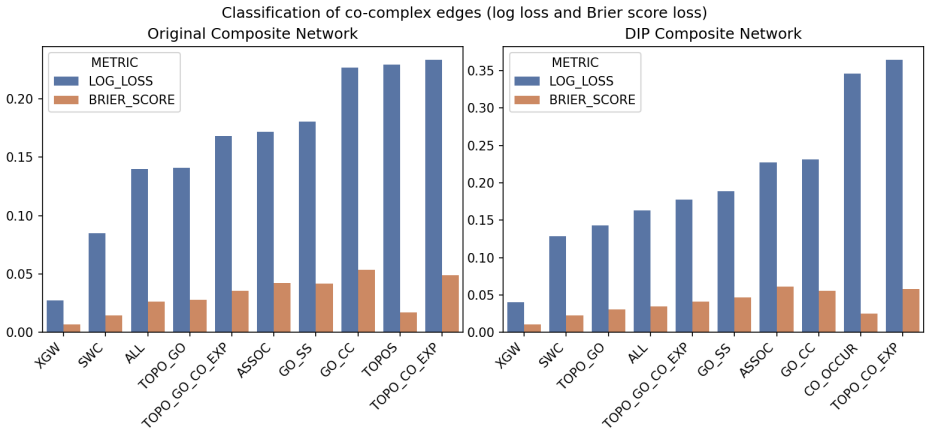

Fig. 1. Log loss and Brier score loss of the top 10 weighting methods for the Original and DIP composite network in terms of co-complex edge classification.

9.2 Precision-Recall Area Under the Curve

Figure 2 shows the top 10 weighting methods in terms of precision-recall area under the curve for both composite protein networks. Here, XGW ranked at least top two in terms of this metric (in the case of the DIP composite network, its score is very close to the score of the top method, which is GO_CC). It is important to note, however, that while GO_CC performed well in this metric, it performed relatively lower when it comes to log loss and Brier score loss; hence, it is inconsistent.

Another surprising result is the low rank of SWC on this metric, which may be attributed to the fact that naive Bayes models are bad estimators of probability.

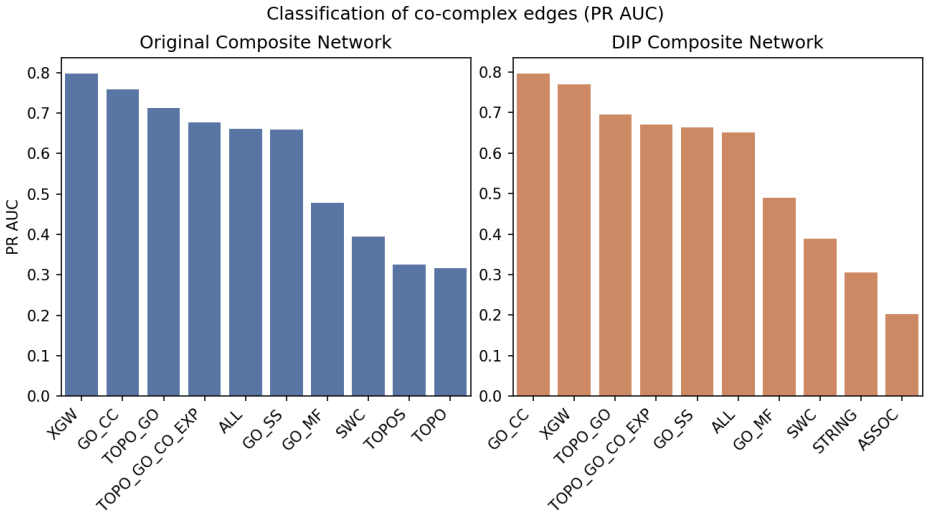

Fig. 2. The precision-recall area under the curve of the top 10 weighting methods for the Original and DIP composite networks in terms of co-complex edge classification.

The consistently high performance of XGW across all three performance metrics shows that XGW is an effective method for classifying co-complex edges.

10 Predicting Complexes

Next, the performance of the 19 weighting methods when used with the MCL algorithm to predict protein complexes was evaluated. For this evaluation, the evaluation setup performed in the SWC study [47] was adopted.

A predicted cluster P_i is said to match a protein complex C_j if

$$JaccardIndex(P_i, C_j) = \frac{|P_i \cap C_j|}{|P_i \cup C_j|} \geq match\_thresh \tag{24}$$

Following the precision and recall formula in [47], we have

$$Recall_d = \frac{|\{C_i \mid C\_i \in C \wedge \exists P_j \in P, dens(P_j) \geq d, P_j \bigotimes C_i\}|}{|C|} \tag{25}$$

$$Prec_d = \frac{|\{P_j \mid P_j \in P, dens(P_j) \geq d \wedge \exists C_i \in C, C_i \bigotimes P_j\}|}{|\{P_k \mid P_k \in P, dens(P_k) \geq d \wedge (\nexists T_i \in T, T_i \bigotimes P_k \vee \exists C_i \in C, C_i \bigotimes P_k)\}|} \tag{26}$$

where C is the set of test protein complexes, P is the set of predicted protein complexes, T is the set of training protein complexes, and $P \bigotimes C$ means cluster P matches complex C.

The precision and recall were computed under different cluster score thresholds d. More specifically, they were computed for these values of d:

$$d = \{0, 0.01, 0.02, 0.03, \ldots, 0.99, 1.0\} \tag{27}$$

.

As a summarizing statistic of the precision-recall curves, the AUC of the PR curve was computed across the 10 cross-validation rounds and was averaged across all rounds and across all the inflation parameter settings of MCL (see Section 7 for the MCL inflation parameter settings).

The precision-recall AUC was computed on four different scenarios:

- Using only the top 20,000 edges and with a $match\_thresh = 0.5$ (which is labeled as `20k_050` on the following graphs)
- Using all the edges and with a $match\_thresh = 0.5$ (which is labeled as `all_050`)
- Using only the top 20,000 edges and with a $match\_thresh = 0.75$ (which is labeled as `20k_075`)
- Using all the edges and with a $match\_thresh = 0.75$ (which is labeled as `all_075`

Figure 3 shows the performance of the top eight weighting methods when used with the MCL algorithm to predict protein complexes for the two composite protein networks.

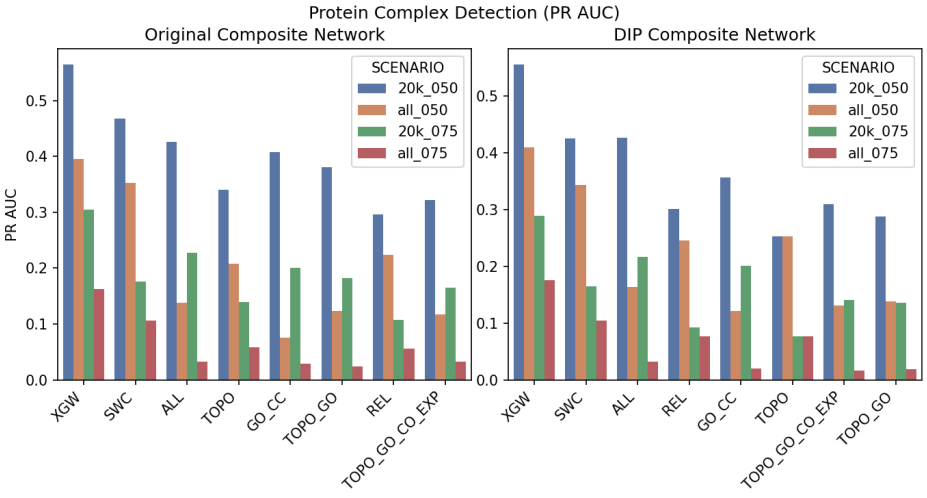

Fig. 3. The Precision-Recall Area under the curve of the top eight weighting methods for the Original and DIP composite networks in terms of protein complex detection.

XGW ranked first in both the composite protein networks while SWC ranked second. These supervised weighting methods ranked high because they weight

each edge based on co-complex probability. Another unsupervised weighting method that performed well is ALL. This signifies that a simple averaging of multiple data sources can achieve decent protein complex detection performance, although not quite as accurate as the supervised weighting methods.

Surprisingly, despite not being a top method in terms of co-complex edge classification, the feature TOPO performed relatively well for protein complex detection. This is consistent with the findings of other studies that show significant improvement in protein complex detection when using topological weighting such as FS-Weight [10], AdjustCD [25], and CD-Distance [7].

11 Feature Importances

Aside from the discussed performance evaluations, the importances of the features used were also estimated. XGBoost automatically computes feature importances based on the amount of accuracy gained when a certain feature is used to split a branch of a tree. This provides a rough estimate of the importance of each feature in predicting the target value (i.e. whether an edge is co-complex or not).

Figure 4 shows the importance of each feature in both the composite networks. As you can see, GO_CC is the most significant, indicating that the colocalization of two proteins is a good predictor if they belong to the same complex. Indeed, this is intuitively true since two proteins can only be a member of a protein complex if they are located closely to each other. Moreover, the STRING feature was also a good predictor. This is expected since it is a highly established database of protein pair functional association that uses multiple evidence types.

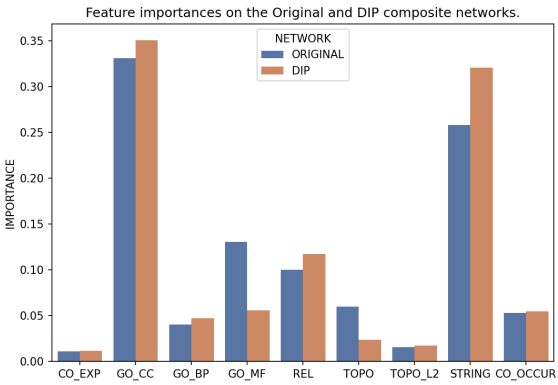

Fig. 4. Feature importances on the Original and DIP composite network.

One surprising result is the fact that the topological features, TOPO and TOPO_L2, were relatively less important compared to others. This is rather in-

teresting as it directly contradicts the high performance of TOPO in the protein complex detection evaluation in the previous section.

However, note that XGBoost calculates feature importances based on accuracy gain when predicting co-complex *edges* (not protein complexes). Thus, this finding is still consistent with the findings in co-complex edge classification in Section 9, which is the fact that TOPO is not one of the top predictors for co-complex edge classification.

The low performance of TOPO on co-complex edge classification, as well as its low feature importance, may be due to the fact that PPINs have a high amount of noise and/or the fact that not all proteins belonging to the same complex interact with each other.

12 Conclusion

In this paper, a supervised weighting method using XGBoost (called XGW) was proposed to weight two composite protein networks based on co-complex probability. XGW outperformed SWC, another supervised weighting method, in all the performance evaluation metrics. This is because XGW extends the features of SWC by using other published unsupervised weighting methods in the literature. In addition, XGW also uses a more advanced machine learning model. With these extensions, more accurate co-complex probability estimates were achieved. More specifically, XGW outperformed SWC in terms of co-complex edge classification and protein complex detection.

Moreover, similar to SWC, XGW also offers a tool to visualize the importances of the features used. The feature importances were calculated based on how much accuracy is gained when a feature is used to split a branch. This allows us to judge what features are noisy and what features are relevant in predicting co-complex edges.

13 Future Directions

While the study offers promising results, further improvements are still needed. First, a more extensive hyperparameter tuning is needed for XGBoost in order to gain better performance and generalization for other protein networks. Second, XGW needs to be applied to other clustering algorithms as well, not just MCL, in order to determine if it performs well with other clustering algorithms. Third, a core-attachment scheme may be used as an additional feature/post-processing step such as the one used in [4, 39]. Moreover, the insights gained from the feature importances results may be used for feature selection to select only the most important and relevant features. Lastly, it would also be interesting to apply this weighting method to protein networks of other species as well to see if it generalizes well.

14 Acknowledgments

The authors would like to thank the authors of SWC for permitting us to use their software and data, and the DOST Engineering Research and Development for Technology for funding the presentation of this paper to the conference.

Appendix

The source code and datasets, as well as supplementary materials, can be found at https://github.com/avancayetano/xgw.

References

1. Armean, I.M., Lilley, K.S., Trotter, M.W.B., Pilkington, N.C.V., Holden, S.B.: Co-complex protein membership evaluation using Maximum Entropy on GO ontology and InterPro annotation. Bioinformatics 34(11), 1884–1892 (Jun 2018)
2. Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., Butler, H., Cherry, J.M., Davis, A.P., Dolinski, K., Dwight, S.S., Eppig, J.T., Harris, M.A., Hill, D.P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J.C., Richardson, J.E., Ringwald, M., Rubin, G.M., Sherlock, G.: Gene ontology: tool for the unification of biology. the gene ontology consortium. Nat. Genet. 25(1), 25–29 (May 2000)
3. Bader, G.D., Donaldson, I., Wolting, C., Ouellette, B.F., Pawson, T., Hogue, C.W.: BIND–the biomolecular interaction network database. Nucleic Acids Res. 29(1), 242–245 (Jan 2001)
4. Beltran, J., Montes, C., Villar, J.J., Valdez, A.R.: A hybrid method for protein complex prediction in weighted protein-protein interaction networks. Philippine Science Letters 10 (02 2017)
5. Bhardwaj, N., Lu, H.: Correlation between gene expression profiles and protein-protein interactions within and across genomes. Bioinformatics 21(11), 2730–2738 (03 2005), https://doi.org/10.1093/bioinformatics/bti398
6. Brown, K.R., Jurisica, I.: Online predicted human interaction database. Bioinformatics 21(9), 2076–2082 (May 2005)
7. Brun, C., Chevenet, F., Martin, D., Wojcik, J., Guénoche, A., Jacq, B.: Functional classification of proteins for the prediction of cellular function from a protein-protein interaction network. Genome Biol. 5(1), R6 (Dec 2003)
8. Chen, T., Guestrin, C.: Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. pp. 785–794 (2016)
9. Cherry, J.M., Hong, E.L., Amundsen, C., Balakrishnan, R., Binkley, G., Chan, E.T., Christie, K.R., Costanzo, M.C., Dwight, S.S., Engel, S.R., Fisk, D.G., Hirschman, J.E., Hitz, B.C., Karra, K., Krieger, C.J., Miyasato, S.R., Nash, R.S., Park, J., Skrzypek, M.S., Simison, M., Weng, S., Wong, E.D.: Saccharomyces genome database: the genomics resource of budding yeast. Nucleic Acids Res. 40(Database issue), D700–5 (Jan 2012)
10. Chua, H.N., Sung, W.K., Wong, L.: Exploiting indirect neighbours and topological weight to predict protein function from protein-protein interactions. Bioinformatics 22(13), 1623–1630 (Jul 2006)

11. Consortium, T.G.O., Aleksander, S.A., Balhoff, J., Carbon, S., Cherry, J.M., Drabkin, H.J., Ebert, D., Feuermann, M., Gaudet, P., Harris, N.L., Hill, D.P., Lee, R., Mi, H., Moxon, S., Mungall, C.J., Muruganugan, A., Mushayahama, T., Sternberg, P.W., Thomas, P.D., Van Auken, K., Ramsey, J., Siegele, D.A., Chisholm, R.L., Fey, P., Aspromonte, M.C., Nugnes, M.V., Quaglia, F., Tosatto, S., Giglio, M., Nadendla, S., Antonazzo, G., Attrill, H., dos Santos, G., Marygold, S., Strelets, V., Tabone, C.J., Thurmond, J., Zhou, P., Ahmed, S.H., Asanitthong, P., Luna Buitrago, D., Erdol, M.N., Gage, M.C., Ali Kadhum, M., Li, K.Y.C., Long, M., Michalak, A., Pesala, A., Pritazahra, A., Saverimuttu, S.C.C., Su, R., Thurlow, K.E., Lovering, R.C., Logie, C., Oliferenko, S., Blake, J., Christie, K., Corbani, L., Dolan, M.E., Drabkin, H.J., Hill, D.P., Ni, L., Sitnikov, D., Smith, C., Cuzick, A., Seager, J., Cooper, L., Elser, J., Jaiswal, P., Gupta, P., Jaiswal, P., Naithani, S., Lera-Ramirez, M., Rutherford, K., Wood, V., De Pons, J.L., Dwinell, M.R., Hayman, G.T., Kaldunski, M.L., Kwitek, A.E., Laulederkind, S.J.F., Tutaj, M.A., Vedi, M., Wang, S.J., DEustachio, P., Aimo, L., Axelsen, K., Bridge, A., Hyka-Nouspikel, N., Morgat, A., Aleksander, S.A., Cherry, J.M., Engel, S.R., Karra, K., Miyasato, S.R., Nash, R.S., Skrzypek, M.S., Weng, S., Wong, E.D., Bakker, E., Berardini, T.Z., Reiser, L., Auchincloss, A., Axelsen, K., Argoud-Puy, G., Blatter, M.C., Boutet, E., Breuza, L., Bridge, A., Casals-Casas, C., Coudert, E., Estreicher, A., Livia Famiglietti, M., Feuermann, M., Gos, A., Gruaz-Gumowski, N., Hulo, C., Hyka-Nouspikel, N., Jungo, F., Le Mercier, P., Lieberherr, D., Masson, P., Morgat, A., Pedruzzi, I., Pourcel, L., Poux, S., Rivoire, C., Sundaram, S., Bateman, A., Bowler-Barnett, E., Bye-A-Jee, H., Denny, P., Ignatchenko, A., Ishtiaq, R., Lock, A., Lussi, Y., Magrane, M., Martin, M.J., Orchard, S., Raposo, P., Speretta, E., Tyagi, N., Warner, K., Zaru, R., Diehl, A.D., Lee, R., Chan, J., Diamantakis, S., Raciti, D., Zarowiecki, M., Fisher, M., James-Zorn, C., Ponferrada, V., Zorn, A., Ramachandran, S., Ruzicka, L., Westerfield, M.: The Gene Ontology knowledgebase in 2023. Genetics 224(1), iyad031 (03 2023), https://doi.org/10.1093/genetics/iyad031
12. Consortium, T.U.: UniProt: the Universal Protein Knowledgebase in 2023. Nucleic Acids Research 51(D1), D523–D531 (11 2022), https://doi.org/10.1093/nar/gkac1052
13. D'haeseleer, P., Church, G.M.: Estimating and improving protein interaction error rates. Proc. IEEE Comput. Syst. Bioinform. Conf. pp. 216–223 (2004)
14. Dongen, S.: Graph clustering by flow simulation. PhD thesis, Center for Math and Computer Science (CWI) (05 2000)
15. Enright, A.J., Van Dongen, S., Ouzounis, C.A.: An efficient algorithm for large-scale detection of protein families. Nucleic Acids Res 30(7), 1575–1584 (Apr 2002)
16. Grigoriev, A.: A relationship between gene expression and protein interactions on the proteome scale: analysis of the bacteriophage T7 and the yeast Saccharomyces cerevisiae. Nucleic Acids Res 29(17), 3513–3519 (Sep 2001)
17. Güldener, U., Münsterkötter, M., Oesterheld, M., Pagel, P., Ruepp, A., Mewes, H.W., Stümpflen, V.: MPact: the MIPS protein interaction resource on yeast. Nucleic Acids Res. 34(Database issue), D436–41 (Jan 2006)
18. Hart, G.T., Ramani, A.K., Marcotte, E.M.: How complete are current yeast and human protein-interaction networks? Genome Biol 7(11), 120 (2006)
19. Jain, S., Bader, G.D.: An improved method for scoring protein-protein interactions using semantic similarity within the gene ontology. BMC Bioinformatics 11(1), 562 (Nov 2010), https://doi.org/10.1186/1471-2105-11-562

20. Jansen, R., Yu, H., Greenbaum, D., Kluger, Y., Krogan, N.J., Chung, S., Emili, A., Snyder, M., Greenblatt, J.F., Gerstein, M.: A Bayesian networks approach for predicting protein-protein interactions from genomic data. Science 302(5644), 449–453 (Oct 2003)
21. Kerrien, S., Aranda, B., Breuza, L., Bridge, A., Broackes-Carter, F., Chen, C., Duesbury, M., Dumousseau, M., Feuermann, M., Hinz, U., Jandrasits, C., Jimenez, R.C., Khadake, J., Mahadevan, U., Masson, P., Pedruzzi, I., Pfeiffenberger, E., Porras, P., Raghunath, A., Roechert, B., Orchard, S., Hermjakob, H.: The IntAct molecular interaction database in 2012. Nucleic Acids Res 40(Database issue), D841–846 (Jan 2012)
22. Kritikos, G.D., Moschopoulos, C., Vazirgiannis, M., Kossida, S.: Noise reduction in protein-protein interaction graphs by the implementation of a novel weighting scheme. BMC Bioinformatics 12(1), 239 (Jun 2011)
23. Licata, L., Briganti, L., Peluso, D., Perfetto, L., Iannuccelli, M., Galeota, E., Sacco, F., Palma, A., Nardozza, A.P., Santonico, E., Castagnoli, L., Cesareni, G.: MINT, the molecular interaction database: 2012 update. Nucleic Acids Res 40(Database issue), D857–861 (Jan 2012)
24. Liu, G., Li, J., Wong, L.: Assessing and predicting protein interactions using both local and global network topological metrics. Genome Inform. 21, 138–149 (2008)
25. Liu, G., Wong, L., Chua, H.N.: Complex discovery from weighted PPI networks. Bioinformatics 25(15), 1891–1897 (Aug 2009)
26. Orchard, S., Ammari, M., Aranda, B., Breuza, L., Briganti, L., Broackes-Carter, F., Campbell, N.H., Chavali, G., Chen, C., del Toro, N., Duesbury, M., Dumousseau, M., Galeota, E., Hinz, U., Iannuccelli, M., Jagannathan, S., Jimenez, R., Khadake, J., Lagreid, A., Licata, L., Lovering, R.C., Meldal, B., Melidoni, A.N., Milagros, M., Peluso, D., Perfetto, L., Porras, P., Raghunath, A., Ricard-Blum, S., Roechert, B., Stutz, A., Tognolli, M., van Roey, K., Cesareni, G., Hermjakob, H.: The MIntAct projectIntAct as a common curation platform for 11 molecular interaction databases. Nucleic Acids Research 42(D1), D358–D363 (11 2013), https://doi.org/10.1093/nar/gkt1115
27. Pagel, P., Kovac, S., Oesterheld, M., Brauner, B., Dunger-Kaltenbach, I., Frishman, G., Montrone, C., Mark, P., Stümpflen, V., Mewes, H.W., Ruepp, A., Frishman, D.: The MIPS mammalian protein-protein interaction database. Bioinformatics 21(6), 832–834 (Mar 2005)
28. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12, 2825–2830 (2011)
29. Peri, S., Navarro, J.D., Amanchy, R., Kristiansen, T.Z., Jonnalagadda, C.K., Surendranath, V., Niranjan, V., Muthusamy, B., Gandhi, T.K.B., Gronborg, M., Ibarrola, N., Deshpande, N., Shanker, K., Shivashankar, H.N., Rashmi, B.P., Ramya, M.A., Zhao, Z., Chandrika, K.N., Padma, N., Harsha, H.C., Yatish, A.J., Kavitha, M.P., Menezes, M., Choudhury, D.R., Suresh, S., Ghosh, N., Saravana, R., Chandran, S., Krishna, S., Joy, M., Anand, S.K., Madavan, V., Joseph, A., Wong, G.W., Schiemann, W.P., Constantinescu, S.N., Huang, L., Khosravi-Far, R., Steen, H., Tewari, M., Ghaffari, S., Blobe, G.C., Dang, C.V., Garcia, J.G.N., Pevsner, J., Jensen, O.N., Roepstorff, P., Deshpande, K.S., Chinnaiyan, A.M., Hamosh, A., Chakravarti, A., Pandey, A.: Development of human protein reference database as an initial platform for approaching systems biology in humans. Genome Res. 13(10), 2363–2371 (Oct 2003)

30. Pu, S., Wong, J., Turner, B., Cho, E., Wodak, S.J.: Up-to-date catalogues of yeast protein complexes. Nucleic Acids Res. 37(3), 825–831 (Feb 2009)
31. Qiu, J., Noble, W.S.: Predicting co-complexed protein pairs from heterogeneous data. PLoS Comput Biol 4(4), e1000054 (Apr 2008)
32. Razick, S., Magklaras, G., Donaldson, I.M.: iRefIndex: a consolidated protein interaction database with provenance. BMC Bioinformatics 9, 405 (Sep 2008)
33. Rigaut, G., Shevchenko, A., Rutz, B., Wilm, M., Mann, M., Sraphin, B.: A generic protein purification method for protein complex characterization and proteome exploration. Nature biotechnology 17, 1030–2 (11 1999)
34. Ruepp, A., Brauner, B., Dunger-Kaltenbach, I., Frishman, G., Montrone, C., Stransky, M., Waegele, B., Schmidt, T., Doudieu, O.N., Stümpflen, V., Mewes, H.W.: CORUM: the comprehensive resource of mammalian protein complexes. Nucleic Acids Res. 36(Database issue), D646–50 (Jan 2008)
35. Salwinski, L., Miller, C.S., Smith, A.J., Pettit, F.K., Bowie, J.U., Eisenberg, D.: The database of interacting proteins: 2004 update. Nucleic Acids Res. 32(Database issue), D449–51 (Jan 2004)
36. Satuluri, V., Parthasarathy, S., Ucar, D.: Markov clustering of protein interaction networks with improved balance and scalability. In: Proceedings of the First ACM International Conference on Bioinformatics and Computational Biology. p. 247256. BCB '10, Association for Computing Machinery, New York, NY, USA (2010), https://doi.org/10.1145/1854776.1854812
37. Shwartz-Ziv, R., Armon, A.: Tabular data: Deep learning is not all you need. CoRR abs/2106.03253 (2021), https://arxiv.org/abs/2106.03253
38. Spirin, V., Mirny, L.A.: Protein complexes and functional modules in molecular networks. Proc Natl Acad Sci U S A 100(21), 12123–12128 (Sep 2003)
39. Sriganesh, S., Kang, N., Hon Wai, L.: MCL-CAw: a refinement of MCL for detecting yeast complexes from weighted PPI networks by incorporating core-attachment structure. BMC Bioinformatics 11(1), 504 (Oct 2010)
40. Stark, C., Breitkreutz, B.J., Chatr-Aryamontri, A., Boucher, L., Oughtred, R., Livstone, M.S., Nixon, J., Van Auken, K., Wang, X., Shi, X., Reguly, T., Rust, J.M., Winter, A., Dolinski, K., Tyers, M.: The BioGRID Interaction Database: 2011 update. Nucleic Acids Res 39(Database issue), 698–704 (Jan 2011)
41. Stark, C., Breitkreutz, B.J., Reguly, T., Boucher, L., Breitkreutz, A., Tyers, M.: BioGRID: a general repository for interaction datasets. Nucleic Acids Res 34(Database issue), D535–9 (Jan 2006)
42. Szklarczyk, D., Franceschini, A., Kuhn, M., Simonovic, M., Roth, A., Minguez, P., Doerks, T., Stark, M., Muller, J., Bork, P., Jensen, L.J., von Mering, C.: The STRING database in 2011: functional interaction networks of proteins, globally integrated and scored. Nucleic Acids Res 39(Database issue), D561–568 (Jan 2011)
43. Tu, B.P., Kudlicki, A., Rowicka, M., McKnight, S.L.: Logic of the yeast metabolic cycle: temporal compartmentalization of cellular processes. Science 310(5751), 1152–1158 (Nov 2005)
44. Van Dongen, S.: Graph clustering via a discrete uncoupling process. SIAM Journal on Matrix Analysis and Applications 30(1), 121–141 (2008), https://doi.org/10.1137/040608635
45. Wang, H., Kakaradov, B., Collins, S.R., Karotki, L., Fiedler, D., Shales, M., Shokat, K.M., Walther, T.C., Krogan, N.J., Koller, D.: A complex-based reconstruction of the Saccharomyces cerevisiae interactome. Mol Cell Proteomics 8(6), 1361–1381 (Jun 2009)

46. Wolfe, C.J., Kohane, I.S., Butte, A.J.: Systematic survey reveals general applicability of "guilt-by-association" within gene coexpression networks. BMC Bioinformatics 6, 227 (Sep 2005)
47. Yong, C.H., Liu, G., Chua, H.N., Wong, L.: Supervised maximum-likelihood weighting of composite protein networks for complex prediction. BMC Syst Biol 6 Suppl 2(Suppl 2), S13 (2012)
48. Young, K.H.: Yeast two-hybrid: so many interactions, (in) so little time. Biol Reprod 58(2), 302–311 (Feb 1998)
49. Yu, Y., Kong, D.: Protein complexes detection based on node local properties and gene expression in PPI weighted networks. BMC Bioinformatics 23(1), 24 (Jan 2022)