Better Link Prediction for Protein-Protein Interaction Networks

Ho Yin Yuen
Department of Computing
The Hong Kong Polytechnic University
Hong Kong, China
andy.aa.yuen@connect.polyu.hk

Jesper Jansson
Department of Computing
The Hong Kong Polytechnic University
Hong Kong, China
jesper.jansson@polyu.edu.hk

***Abstract*—In functional genomics, experimentally obtained protein-protein interaction (PPI) data is often incomplete. To deal with this issue, computational approaches are used to infer missing data and to evaluate confidence scores. Link prediction is one such approach that uses the structure of the network of PPIs known so far to find good candidates for missing PPIs. In a recent study by Kovács *et al.*, a novel PPI-specific link predictor was proposed. Their link predictor is biologically motivated by the so-called *L3 principle* and it was shown to be superior to other general link predictors when applied to PPI data. However, the *L3* link predictor is only an approximate implementation of the L3 principle. As such, not only is the full potential of the L3 principle not realized, it may even penalize candidate PPIs that otherwise fit the L3 principle. In this paper, we formulate an *L3*-based link predictor without approximation, coined *ExactL3*. We show computationally that *ExactL3* is better than the previously proposed methods on four major PPI datasets (STRING, BioGRID, IntAct/HuRI, and MINT). The predicted PPIs are also shown to be much more functionally relevant. This confirms that *ExactL3* is a better link predictor for PPI networks, and demonstrates its ability to characterize PPIs by only the topological features of binary PPI networks.**

***Index Terms*—Protein-Protein Interaction, Link Prediction, Complex Network, Graph Theory**

I. Introduction

In the post-genomic era, high-throughput techniques have been developed to retrieve and analyze high-level and dynamic cellular activities. An important example is the development of techniques that enable large-scale characterization of protein interactions [1]. This has led to a new type of interactome for system biology, the Protein-Protein Interaction (PPI) network [2]. A PPI network is a form of complex network where a node represents a protein, and an edge indicates that two proteins can interact with each other. Since PPIs describe signals transduction of protein physical docking [3], large-scale studies can provide insights into the molecular machinery of living systems [4]. On a basic level, a PPI network can represent signaling pathways as a chain of PPIs [5], and a protein complex as a graph cluster [6]. In more advanced use cases, analysis in targeting disruptive PPIs can even introduce new cancer therapeutic strategies [7].

The basis of meaningful and comprehensive discoveries is a complete and reliable PPI network. However, measurement errors or incomplete experimental data may lead to some parts of the constructed PPI network having the wrong structure. For this reason, computational tools have been developed to evaluate the edges in an existing PPI network or to find good candidates for new edges that should be added in order to make the resulting network more biologically sound. The most direct approaches use protein sequences data [8] [9], since protein sequences compare proteins' functions genetically. Some of the other approaches include the use of protein structures, RNA co-expression, and protein annotations [10] [11]. Undoubtedly, these methods are successful by utilizing features to describe proteins, subsequently characterizing PPIs.

On the other hand, general-purpose link prediction techniques have been developed for complex networks, such as PPI networks and social networks [12]. These link predictors raise interests due to their abilities to characterizes PPIs with only binary PPIs data. However, they are usually not specific enough to characterize PPIs, and there are no guarantees on their correctness and reliability. Due to this concern, Kovács *et al.* [13] recently introduced a novel link predictor based on a biological motivation that they called the *L3 principle*. This principle hypothesizes that two proteins linked by many different paths of length three have a higher likelihood of also interacting directly with each other. Using the L3 principle, the *L3* link predictor infers new PPIs by scoring the structure of candidate PPIs, and keeping the candidates with the highest scores. [13] also argued that for PPI networks, being linked by many paths of length two has the opposite effect, and showed experimentally that the *L3* link predictor outperforms a vast number of general link predictors, including the famous Common Neighbor [14] that favors paths of length two.

Despite the strength of the *L3* principle, some researchers claim that our understanding of the *L3* link predictor is limited and that it was derived empirically rather than from any theoretical knowledge [15]. In fact, one can regard the *L3* link predictor as an *approximation* in the sense that it penalizes the score of a neighborhood if some of its properties imply that it is a coincidence. This generally happens to any link predictor and each has different measures to address this. However, the penalization in the *L3* link predictor applies even to PPIs that should be rewarded for such properties. So, a better approach would be to evaluate its fitness to the L3 principle by characterizing a neighborhood of PPI

2471-7819/20/$31.00 ©2020 IEEE
DOI 10.1109/BIBE50027.2020.00017

more precisely, namely to reward desirable graph structures such as paths of length three, and penalize undesirable graph structures such as paths of length two. In this paper, we show how to define the link predictor in a way that more accurately corresponds to the biological motivation of the L3 principle. Our approach is coined *ExactL3*. The advantages of the *ExactL3* link predictor is that it is better at inferring undiscovered PPIs, and it demonstrates how to characterize PPIs biologically with only the PPI topology.

The paper is organized as follows. Section II reviews some known general and PPI-specific link prediction techniques. Then, we provide the problem definition and the formulation of *ExactL3* in Sections III and IV, respectively. We evaluate *ExactL3* in Sections V and VI by comparing it to five other link predictors in a series of computational experiments, using data from four different databases for two organisms. Finally, the paper wraps up with a discussion of the results and some suggested follow-ups in Section VII.

II. Previous Work

Link prediction infers new edges based on the properties of the nodes as well as the overall topology of the existing edges [12]. Many subclasses of link prediction approaches exist, and this paper will be focusing on similarity-based link predictions, where nodes are connected based on their topological similarity. Two such predictors are reviewed next.

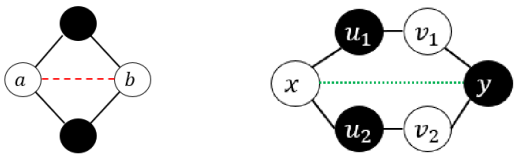
A. General Link Prediction

The Common Neighbors (CN) concept originates from social networks [14]. This concept characterizes a social phenomenon: the more friends two individuals shares, the more likely they are also be friends of each other. From here on, for any node a, let $N(a)$ denote the set of neighbor nodes of a. Then, the CN score of any two nodes a and b is $|N(a) \cap N(b)|$. The higher the CN score, the more confident we can be that the two nodes should be adjacent. In the context of PPIs, a high CN score of two proteins implies they are of similar functions [16]. That is, if two proteins interact with similar set of proteins, their functions are then similar.

However, a high-degree node will contribute to the CN scores of many more node pairs than a low-degree node will. Consequently, it is a good idea to penalize high-degree nodes in the CN index. To do so, the Resource Allocation (RA) algorithm [17] makes high-degree nodes contribute less by using the following formula instead for every pair of nodes a and b: $\sum_{z \in N(a) \cap N(b)} \frac{1}{|N(z)|}$. In addition to RA, there exist many other normalization schemes. In the Adam-Adar (AA) Index [18], a logarithm modifier (motivated in the context of social networks mining) is used to do the normalization: $\sum_{z \in N(a) \cap N(b)} \frac{1}{\log(|N(z)|)}$. For many other normalization schemes based on different motivations, see the survey of general link predictions [12].

B. PPI-specific Link Prediction

Given the context of PPI networks, link predictions can extend beyond neighborhoods of nodes. A study [19] applies random walks to identify and connect pairs of nodes that have similar distances to the other nodes in the network, and showed that this method reconstructs PPI networks with greater biological relevance. This can be classified as the global approach in similarity-based link predictions.



(a) Common Neighbors in PPI (b) L3 Hypothesis in PPI

Fig. 1: Let the color of the node be the type of its protein interface, and assume that node pairs with different colors are compatible. In (a), node a and node b have common neighbors, which suggests that the interface of a and the interface of b are similar; therefore, one can assume that a and b will not be compatible with each other, as indicated by the dashed red line. In (b), nodes x and y are connected by P_4-subgraphs. It is possible for node x, v_1 and v_2 to have the same type of interface (white) and nodes y, u_1 and u_2 to have a complementary interface (black), as shown in the figure. In this case, x and y will be compatible, as shown by the dotted green line. The other possibility, i.e., that x and y have the same interface, is unlikely because u_1 is not known to interact with y and v_1 is not known to interact with x; additional P_4-subgraphs between x and y (such as the one involving u_2 and v_2) make this scenario even less likely.

Another study [20] uses protein complex datasets on top of PPI datasets to investigate how many PPIs might be missing from those PPI datasets. Assuming that each protein complex must induce a connected subgraph in the corresponding PPI network, the minimum number of edges that have to be added to ensure that this condition holds in the network thus gave lower bounds on the number of missing PPIs in various databases. This also shows how PPI datasets can be augmented with external feature data, utilizing the biological context.

Finally, the study of our focus [13], the L3 algorithm is biologically motivated by the following observation: Since a physical PPI is the physical docking of two proteins, it can only occur if the interfaces of the two proteins are compatible. Now, if nodes x and y in a PPI network share many neighbors, it can be expected that the interface of x is similar to the interface of y. Two proteins with identical or nearly identical interfaces are usually not compatible (they cannot dock with each other), which means that the PPI network will not have an edge between x and y in this case. See Fig. 1(a) for an illustration. On the other hand, if there are many paths of length 3 between x and y in the network then x and y are likely to be compatible, as shown in Fig. 1(b). Following standard graph theory notation, P_4 will denote an undirected length-3 path consisting of four nodes and three edges. The observation above can be stated as: the more P_4-subgraphs that connect a pair of nodes x and y, the more certain it is that x and y should be connected by an edge. In the rest of this paper, we shall refer to this principle as the *L3 principle*.

III. Problem Definition

Given an undirected graph $G = (V, E)$, our goal is to iterate through all non-adjacent node pairs in V, and determine for each such pair whether or not an edge between them should be added to E. Every non-adjacent node pair $\{x, y\}$ will be assigned a score P_{xy} that measures, in a relative sense, the confidence with which one can say that x and y should be connected by an edge. As explained in Section II-B, one can compute P_{xy} based on the L3 principle simply by counting the number of P_4-subgraphs between x and y. For this purpose, define $U = N(x) \cap N(N(y))$ and $V = N(y) \cap N(N(x))$, i.e., let U be the set of neighbors of x at distance 2 from y and analogously for V. Then, every P_4-subgraph between x and y is an undirected simple path of the form (x, u, v, y), where $u \in U$ and $v \in V$. Note that a node may belong to $N(x)$ as well as $N(y)$ and also to both U and V, in which case it will be able to take the role of either u or v in a P_4-subgraph. With these definitions, one can count the number of P_4-subgraphs between x and y using Formula 1, and this kind of double summation will be abbreviated as Formula 2 to simplify the notation from now on.

$$P_{xy}^{(1)} = \sum_{i=1}^{|U|} \sum_{j=1}^{|V|} \begin{cases} 1 & \text{if } u_i \in N(v_j) \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

$$P_{xy}^{(1)} = \sum_{U,V} 1 \tag{2}$$

However, similar to what was mentioned in Section II-A, high-degree nodes in the sets U and V will contribute to many more P_4-subgraphs than low-degree nodes, giving them a disproportionate influence on the value of P_{xy}. Hence, Formula 2 should be adjusted to penalize high-degree nodes. The *L3* algorithm [13] does this by using a square root modifier as in Formula 3 below.

$$P_{xy}^{(L3)} = \sum_{U,V} \frac{1}{\sqrt{|N(u_i)| \cdot |N(v_j)|}} \tag{3}$$

The normalization modifier makes Formula 3 an approximation, similar to the AA index in Section II-A. Namely, only set U, set V, and the degree of its nodes are used to evaluate the L3 structure of a xy node pair. This does not sufficiently characterize a L3 structure, and may incorrectly penalize xy-links that are otherwise highly likely (despite having high degree nodes u and nodes v). *ExactL3* addresses these problems by proposing an alternative approach to normalization. Before presenting the formulation, we first give more intuition behind the *L3* principle introduced in Section II-B.

Recall that in the *L3* principle, the interface compatibility of node x and node y can be evaluated using the number of P_4-subgraphs. The use of P_4 is justified by considering $N(x)$ and $N(y)$. First, we evaluate if x is incompatible with nodes in $N(y)$. Since proteins with similar interfaces are incompatible, we are then evaluating if protein x has similar interfaces to all proteins $N(y)$. Similarly, we evaluate if nodes in $N(x)$ are incompatible with y for the same reason. Combining both evaluations, since $U \subseteq N(x)$ and $V \subseteq N(y)$, a P_4 consisting of (x, u, v, y) is a minimum expression of the *L3* principle. $|U|$ evaluates the number of compatible nodes of x, and $|V|$ evaluates the number of incompatible nodes of x. This applies symmetrically to y as well (i.e., $|V|$ evaluates the number of nodes compatible with y, and $|U|$ evaluates the number of nodes incompatible with y). So, the number of P_4-subgraphs determines the PPI confidence between x and y.

Taking the above considerations into account, *ExactL3* addresses the evaluation more accurately by directly evaluating the ratio of compatible nodes and incompatible nodes. For example, one has to penalize the confidence score of an edge between x and y if either of them has adjacent nodes that cannot form a P_4, and reward if otherwise. In the next section, we formulate the *ExactL3* link predictor.

IV. Methods

In this section, we formalize the ideas of *ExactL3* (Formula 4). It is based on the concept of the Jaccard similarity coefficient and a simple penalization index. The *Jaccard similarity coefficient* measures the overall similarity and dissimilarity of two sets A and B of data and is defined as the value $\frac{|A \cap B|}{|A \cup B|}$.

$$P_{xy}^{(L3E)} = \frac{|U|}{|N(x)|} \cdot \frac{|V|}{|N(y)|} \cdot \sum_{U,V} \frac{|N(v) \cap N(x)|}{|N(v) \cup N(x)|} \cdot \frac{|N(u) \cap N(y)|}{|N(u) \cup N(y)|} \tag{4}$$

As explained in Section 3, the *L3* principle involves two parts: to evaluate if x is incompatible with nodes in $N(y)$, and to evaluate if nodes in $N(x)$ are incompatible with y. *ExactL3* is defined to directly address these two parts.

The first part can be realized as depicted in Fig. 2. The figure demonstrates the use of $\frac{|N(v) \cap N(x)|}{|N(v) \cup N(x)|}$(b) and $\frac{|V|}{|N(y)|}$(c) applied to the all *L3* paths of xy (a). For each *L3* path between xy, (b) is used to evaluate whether node v_1 in $N(y)$ are similar to x in terms of their neighborhood (set U). Since each *L3* path will yield a (b) score of that path, these scores will be summed together (Formula 4). Then, (c) is used to evaluate how compatible y is with set V, since there could be non-V nodes in $N(y)$. The combination of (b) (c) as in Formula 4 can then realize the first part of the *L3* principle, since if x is similar to all v (Qb), and y is compatible with V (Qc), then x is likely to be compatible with y (Qa) (i.e., incompatible with nodes in $N(y)$). The second part of the *L3* principle is to evaluate if nodes in $N(x)$ are incompatible with y. This is accomplished analogously by the remaining terms in Formula 4 ($\frac{|N(u) \cap N(y)|}{|N(u) \cup N(y)|}$ and $\frac{|U|}{|N(x)|}$).

A. Time Complexity

We now analyze the time complexity of *L3* (Formula 3) and *ExactL3* (Formula 4). Let n denote the number of nodes in G. The CN link predictor is already known to run in $O(n^3)$ time [21].

The main operations in both *L3* and *ExactL3* are the set operations on node neighborhoods. We first elaborate on how

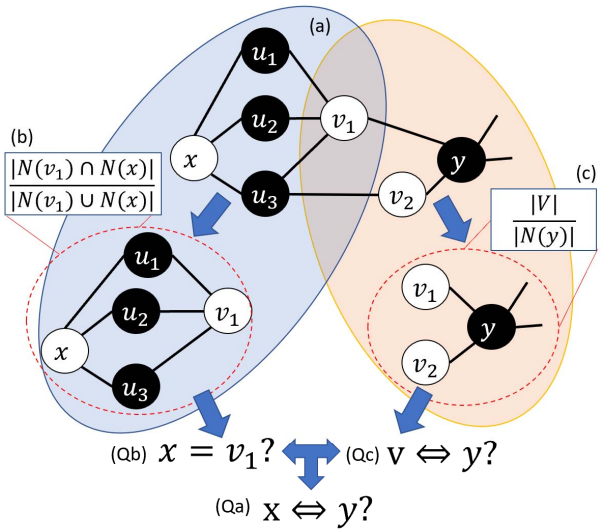


Fig. 2: The central idea of the *ExactL3* formulation. (Qa) To estimate the compatibility of x and y based on the *L3* principle, (a) we identify all *L3* paths between x and y. For any *L3* path $\{x, u_i, v_1, y\}$, we ask two questions: (Qb) Are the interfaces of x and v_1 similar in terms of their neighborhood? (Qc) Are v nodes compatible with y in terms of $N(y)$? These two questions are encoded in the scoring functions (b) and (c) so that the higher the scores, the more likely the answers will be yes. To model all *L3* paths, Formula 4 combines the contributions of (b) and (c) to evaluate if x is incompatible with $N(y)$ and also the analogous scoring functions for evaluating if y is incompatible with $N(x)$.

to perform two set operations, the set intersection and the set union. Every graph neighborhood will be precomputed and stored in a hash table so that it takes $O(1)$ time to check if a node belongs to a set $N(a)$. To do the set intersection operation $A \cap B$, simply look up each of the elements of the smaller set in the hash table for the larger set. Thus, $A \cap B$ takes $O(\min(|A|, |B|))$ time. For the set union operation $A \cup B$, one has to access all elements of both sets if intersection is empty, so it takes $O(|A|+|B|)$ time. Notice that the time complexity for the set union operation dominates that of the set intersection operation.

Both *L3* and *ExactL3* need to evaluate $O(n^2)$ pairs of nodes to perform link prediction. For each such pair $\{x, y\}$, the sets $U = N(x) \cap N(N(y))$ and $V = N(y) \cap N(N(x))$ are constructed in $O(n^2)$ time. (To construct U, check each of the $O(n)$ nodes in $N(x)$ to see if any of its $O(n)$ neighbors is in the hash table for y's neighborhood, and if so, include it in U; construct V in the same way.) After that, *L3* iterates over the $O(n^2)$ pairs in U and V and applies the normalization from Formula 3 to each one in $O(1)$ time. Therefore, *L3* runs in $O(n^4)$ time. Next, *ExactL3* is the same as *L3* except that the normalization is done according to Formula 4 instead. Here, the normalization uses set intersection and set union operations and takes $O(\min(|N(v)|, |N(x)|) + (|N(v)|+|N(x)|) + \min(|N(u)|, |N(y)|)+(|N(u)|+|N(y)|)) = O(n)$ time by the above. In summary, the time complexity of *ExactL3* is $O(n^5)$.

V. Materials

The *ExactL3* formulation (Formula 4) (*L3E*), the original *L3* link predictor [13], two other recent link predictors based on the *L3* principle (CH2_L3 [15] (*CH2*), Sim [22]), and two other general link predictors (CN [14], CRA [23]) will be evaluated and compared against each other on four PPI datasets (BioGRID [24], STRING [25], IntAct [26], MINT [27]) of two organisms (Saccharomyces cerevisiae S288c (Yeast), Homo sapiens (Human)). For the human dataset, the HuRI dataset [28] is used as it is one of the variations within the IntAct dataset (identifier IM-25472).

Two computational setups are used for the experiments, one with 24 cores and 128GB RAM (for BioGRID and STRING human datasets), another with 8 cores and 16GB RAM (the rest of the datasets). The program was written in Python 3.6 with multiprocessing capabilities.

For all datasets, only physical PPIs (i.e. binary interactions) are considered, which is the focus of the *L3* principle. We extract PPIs using annotations in datasets as follows: 'physical' for BioGRID; 'binding' for STRING; 'direct interaction', 'physical association', and 'association' for both IntAct and MINT; HuRI by default includes only binary interactions. For co-complex interactions in IntAct, the Spoke Model is used to convert them into binary interactions [29] (i.e., assume prey proteins interact only with bait proteins in the co-complex).

We do 10 experiments for each dataset. In each experiment, we first remove 50% of the edges chosen uniformly at random; here and in the next section, let γ denote the number of edges that were removed in a particular experiment. For each link predictor, we then rank all non-neighboring nodes x and y (called *candidate edges*) according to their scores. The top γ ranked candidate edges are selected as the predicted edges.

We evaluate the quality of the predicted edges by a statistical method (Precision-Recall) as well as three biological metrics: Gene Ontology Semantic Similarity (GOSemSim) scores, confidence scores, and gene essentiality. To compute the GOSemSim scores between proteins, we use an R language package [30] based on Wang's method [31] with the BWA strategy. The confidence score of each predicted PPI is extracted from the STRING dataset directly. Both scores are defined to be zero if the computation returns null or the score does not exist. For essential genes datasets, all 1080 essential genes of yeast, and all 3230 essential genes of human were downloaded from [32] and [33] (using the backup in [34]) respectively.

VI. Results

In this section, we first evaluate *ExactL3* (L3E) and other link predictors by Precision-Recall. Then, we evaluate the biological relevance of the top predicted PPIs in three metrics. *ExactL3* has the best performance in all evaluations.

A. Experiment Statistics

Table I summarizes the graph properties of all the PPI datasets. In each experiment, any node that becomes isolated during the edge removal process will be removed from the graph so that no candidate edges involving that node will be

	# of Nodes		# of PPIs		Average # of Candidate PPIs		γ	
	Yeast	Human	Yeast	Human	Yeast	Human	Yeast	Human
BioGRID	6,815	24,381	172,448	604,747	20,115,400	224,868,220.7	86,224	302,373
IntAct / HuRI	5,418	8,135	143,750	167,331	1,166,076	30,904,949.3	71,875	61,413
STRING	4,574	13,712	46,298	122,827	8,870,602.6	74,147,866.4	23,149	83,665
MINT	4,056	7,430	39,429	33,146	6,479,681	17,396,746.3	19,714	16,573

TABLE I: Dataset graph properties

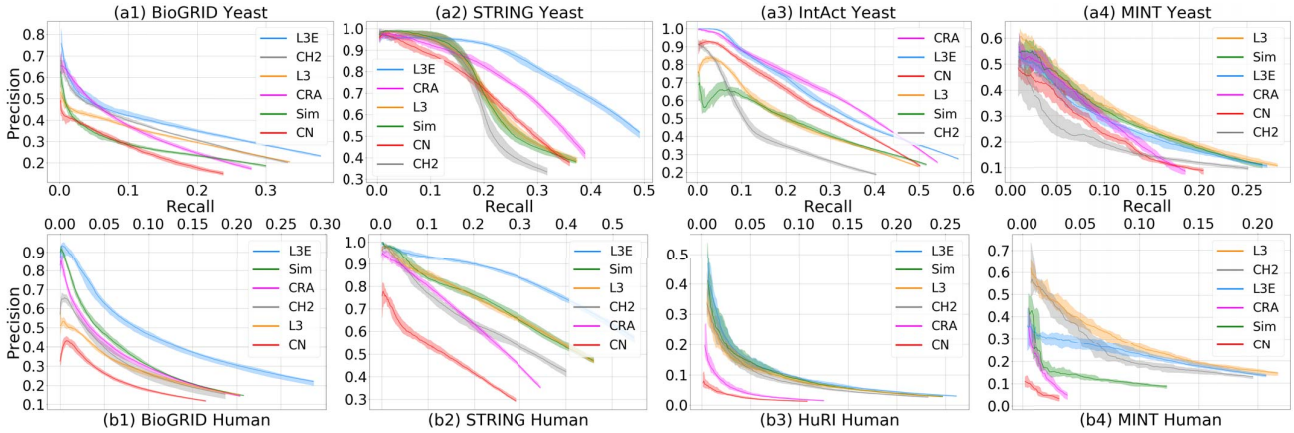


Fig. 3: (Section VI-B) Precision-Recall (PR) of the link predictors. The shaded regions show the PR-curves' ranges in the ten experiments, and the solid lines are their average values. Each figure legend ranks the link predictors according to their PR-curves.

considered. Hence, the average number of resulting candidate edges for each dataset is presented in the table. In all cases, the human datasets are much larger than the yeast datasets (e.g., HuRI has 26.5 times more candidate edges). We can rank the size of the datasets from largest to smallest as follows: BioGRID, IntAct/HuRI, STRING, MINT. Table II shows the average running times for all the link predictors in ten experiments for each dataset. Here, we can see that STRING and MINT take less time than BioGRID and IntAct/HuRI, as expected from their scales. Also, the running time of *ExactL3* increases drastically compared to *L3*, or *CN* as the dataset size increases, which agrees with the time complexity analysis in Section IV-A. To summarize, the running time of the link predictors can be ranked as follows (low to high): CN, CRA, *L3*, Sim, *ExactL3*, CH2_L3. For *ExactL3*, all obtained scores were between 0.000222 and 0.216915, with a precision up to 19 decimal places, and there were no significant truncation errors.

Saccharomyces cerevisiae (Yeast)						
	CN	*L3*	*ExactL3*	CRA	CH2_L3	Sim
BioGRID	2.81	8.00	60.95	2.68	81.74	20.72
IntAct	1.63	4.11	33.49	1.73	42.93	10.07
STRING	1.09	1.32	1.63	1.11	1.70	1.46
MINT	0.94	0.99	1.03	0.89	0.93	0.97
Homo sapiens (Human)						
	CN	*L3*	*ExactL3*	CRA	CH2_L3	Sim
BioGRID	0.91	17.83	104.95	0.98	154.75	46.16
HuRI	4.15	6.07	10.50	4.30	11.01	7.89
STRING	0.23	0.94	1.51	0.24	1.31	1.09
MINT	2.52	2.66	2.93	2.48	2.83	2.74

TABLE II: Average running times (in minutes) taken over ten experiments. For Human BioGRID and Human STRING, more computational resources are used, as stated in Section V.

B. ExactL3 Improves PPI Link Predictions Statistically

In this section, we evaluate the changes in Precision-Recall of the predicted edges, starting from the top 100 edges to the top γ edges (see Table I) for each link predictor in all datasets. Precision is defined as the ratio of true-positive edges in all predicted edges, and Recall is defined as the ratio of true-positive edges in all removed edges. The PR curve shows the decrease in a link predictor's performance (of identifying true-positive edges) as the ratio of recovered edges increases.

In Fig. 3, we can see a general trend across datasets for *ExactL3*. *ExactL3* performs well on BioGRID and STRING (1,2), similar to the other methods on IntAct/HuRI (3), and worse than the others on MINT (4). Also note that regardless of the scale (organism), *ExactL3* is able to perform consistently, unlike predictors such as CN in IntAct/HuRI. *ExactL3* has the best performance for the majority of the datasets (five cases) compared to other predictors (*L3* is the second best, with two cases). Within the top three of all datasets, they are mostly dominated by *L3*-based predictors (6 out of 8 times). This shows that the *L3* principle can be used to predict missing PPIs well in terms of Precision-Recall.

C. ExactL3 Predictions are More Biologically Relevant

Apart from evaluating the link predictors statistically, we are also interested in the biological relevance of the predicted PPIs. We evaluate the link predictors with the following three metrics: Gene Ontology (GO) Semantic Similarity (GOSemSim), STRING Confidence Scores, and Gene Essentiality. The motivation for each metric will be elaborated on below.

1) GOSemSim Scores: GO annotations are used to describe features of proteins, and the annotations are divided into three root categories: cellular component, molecular function, and

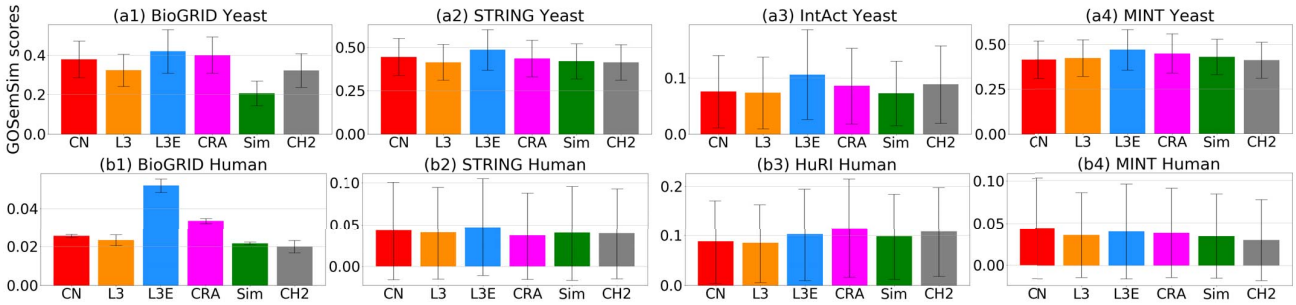


Fig. 4: (Section VI-C) Average GOSemSim for the top 10% of the γ predicted edges. The higher the GOSemSim scores, the better.

biological process. While GO annotation is a single keyword, this keyword is actually a part of its "GO tree" where many other annotations are parent class of this annotation. As such, GO annotations are highly specific and rich descriptors in comparing the functional similarity of two proteins, and those with high GOSemSim scores are likely to have PPIs [35].

Fig. 4 shows the GOSemSim score for the top 10% of the top γ predicted PPIs of each link predictor across all datasets. *ExactL3* has the best performance in both the yeast and the human datasets (except (b3) being slightly worse). The second best predictor is CRA, which outperforms *L3* consistently as also shown by the data in [13]. Here, we can see that in contrast to the previous section, CRA and CN is always the top two and three respectively. This is natural as they are CN-based link predictors, and the CN principle prioritizes proteins with similar functions (i.e., high GOSemSim score). Since *ExactL3* is based on *L3*, the high GOSemSim scores further show the biological relevance of its link prediction.

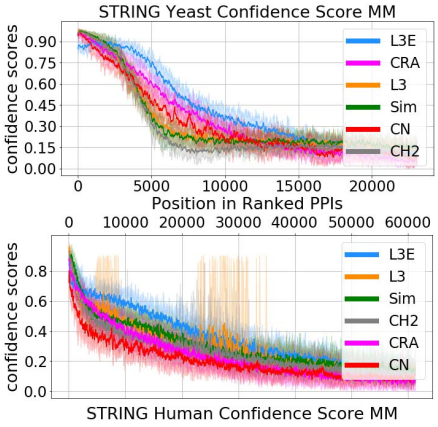


Fig. 5: Moving averages of the STRING confidence scores for the top γ edges predicted by each link predictor. The shaded regions and the labels are arranged as in Fig. 3.

2) STRING Confidence Scores: The STRING confidence score estimates the confidence of a PPI from various factors such as gene co-expression, literature mining, etc. The moving averages (MM) of the STRING confidence scores for the top γ edges are shown in Fig. 5, with an MM-window size of 100 edges, and where each iteration slides pass 10 edges. MM reflects the scores of the corresponding local portions of PPIs. Unlike Section VI-C1, all of the top γ edges are evaluated because STRING confidence scores have more comprehensive biological evidence and can thus satisfactorily evaluate low-rank PPIs. According to Fig. 5, *ExactL3* has the best and the most consistent performance among the link predictors across two organisms.

3) Gene Essentiality: Essential genes are proteins that its deletion will be lethal to an organism. Since they are important proteins, they are more likely to be expressed as network hubs in PPI networks [36]. Naturally, it is easier for link predictors to pick PPIs related to essential proteins due to its topological significance (i.e., high degree). So, if a link predictor can detect essential genes with less-apparent topological traits (i.e, low degree), the link predictor is then better normalized to discover functional PPIs.

Next, we evaluate how sensitive the link predictors are in identifying PPIs with essential genes. Then, we evaluate the average degree of these essential nodes. This way, we can evaluate the topological bias of the essential genes selected by the link predictors. Fig. 6 shows the cumulative hit of essential genes, and Fig. 7 shows the node degree of these essential genes within the top 10% of the γ predicted PPIs. As shown in Fig. 6, the predicted PPIs by *ExactL3* are most sensitive to essential genes in datasets of both organisms, except in Fig. 6(a1) but with only a small difference to the top ones. Then, in Fig. 7 we can also see that the top identified essential genes by *ExactL3* have relatively lower node degree than other link predictors. This shows that *ExactL3* can identify essential genes more sensitively while being more independent to the topological properties (i.e., not finding hub nodes). Note that in some cases, Sim picks essential genes with much lower node degree. However, it is not as sensitive to essential genes as *ExactL3*.

VII. Discussion

In this paper, *ExactL3* is formulated based on the biological motivation of the *L3* principle. *ExactL3* shows improved performance statistically and functionally compared to two CN-based link predictors (CN, CRA) and three *L3*-based link predictors (*L3*, CH2_L3, Sim).

A. ExactL3 Characterizations

Overall, *ExactL3* shows improved performance statistically and functionally in numerous yeast (small-scale) and human

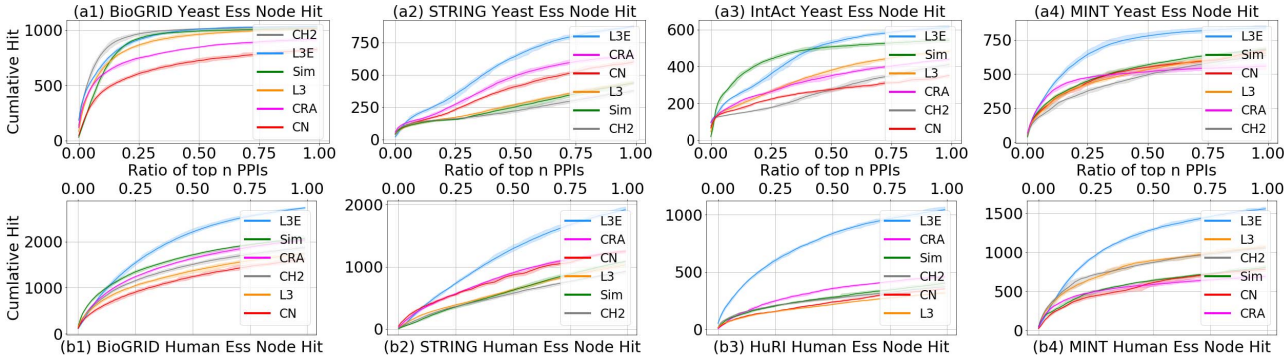


Fig. 6: (Section VI-C3) Comparing the cumulative hit of essential genes in the order of the γ^{th} predicted PPIs for each link predictor. Here, the higher the curve is, the better. Each figure legend ranks the link predictors according to their curves.

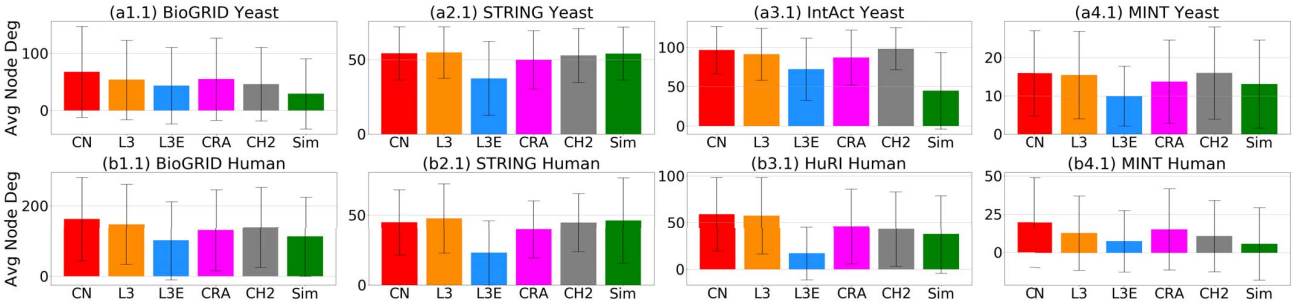


Fig. 7: The average node degree of the essential genes within the top 10% of the γ predicted PPIs. Each figure maps to the corresponding figure in Fig. 6 (e.g., Fig 7(a1.1) maps to Fig 6(a1)). Here, the lower the average node degree, the better.

(large-scale) PPI datasets. Its performance is most apparent in Section VI-C3 (Gene Essentiality), matching our assumption that if *ExactL3* characterizes PPIs better, then it will be sensitive to essential genes even with low degree. As for Precision-Recall, Fig. 3 confirms that *ExactL3* is the best of the six evaluated methods for most of the datasets, and always among the top-performing ones. Furthermore, *ExactL3*'s Precision-Recall pattern seems more consistent across different organisms in the same dataset than the other methods'. It is worth investigating if there is a topological bias within the datasets that results in this pattern for *ExactL3* but not the others.

Surprisingly, *ExactL3* performs well in terms of GOSemSim scores, contrary to other *L3*-based predictors (and also as shown in the data of [13]). As explained in Section VI-C1, it is natural for CRA and CN to be better than the *L3*-based predictors due to their inherent advantage (CN indicates two nodes having similar functions). So for *ExactL3*, we believe the logical explanation would be that it characterizes specific subtypes of physical PPIs better, or physical PPIs that are obtained by specific types of experiments. Methods such as the ones in Section VII-C may help us to understand the apparent discrepancy in performance among different datasets.

B. Significance

Characterizations of PPIs can not only be used to predict PPIs, but also to assess the confidence of existing PPIs for PPI analysis [7]. As such, it is important to characterize PPIs according to the context. Naturally, conventional features that directly imply PPIs such as domain interactions [37], or gene co-expression are used for characterizations instead. Thus, graph features in PPI networks have not been fully utilized, as shown by the lack of related research in the review in [7]. In this paper, we show that given a proper hypothesis (the *L3* principle), PPIs can be characterized more specifically by modifying the mathematical terms (*ExactL3*). This yields a way to characterize a biological phenomenon topologically, and also provides a cheap method to evaluate PPIs when only binary interactions data is available (i.e., no external biological data) to characterize PPIs.

In addition to the above, *ExactL3* may be a valuable tool in PPI analysis. In some studies such as [38], proteins with significant topological properties such as betweenness and closeness are shown to be feasible drug targets. Similarly, an improved *L3* link predictor like *ExactL3* could be useful to find significant PPIs. For example, since incompatible neighbor nodes reduce the *ExactL3* score, one of the topological interpretations of a node pair having a high *ExactL3* score is that there may be an abundance of compatible proteins and similar node pairs nearby. This shows that the interaction of such node pairs are significant, and may even be essential.

C. Future Work

It would be useful to know exactly what types of subgraphs (network motifs) are prioritized by *ExactL3*. Insights into what these motifs imply biologically will then help explain why

ExactL3 is better than the other link predictors, reveal more precisely to what extent *ExactL3* can characterize PPIs, and potentially lead to even better link predictors.

Also, datasets have to be dissected to investigate the bias of each link predictor more comprehensively. In this paper, we use the dataset annotations to extract relevant PPIs for evaluations. However, it is known that some experimental methods, such as the Y2H assay, can obtain binary PPIs more accurately compared to other methods [28]. So, it would be better to divide PPIs within a dataset into various types of PPIs to better highlight specific traits of the link predictors. Another way would be to perform GO enrichment analyses, and to see what types of physical binding activities the abundant GO terms can be attributed to.

Finally, since the current *ExactL3* formulation has a high running time (see Table II), it would be helpful to derive more efficient yet equivalent mathematical formulations.

D. Availability

The algorithms and the scripts written to generate and extract the data for experiments, and a command-line program to use *ExactL3* are all included in the following GitHub repository: https://github.com/andy897221/ExactL3_PPI

REFERENCES

[1] D. Eisenberg, E. M. Marcotte, I. Xenarios, and T. O. Yeates, "Protein function in the post-genomic era," *Nature*, vol. 405, no. 6788, pp. 823–826, 2000.

[2] M. E. Cusick, N. Klitgord, M. Vidal, and D. E. Hill, "Interactome: gateway into systems biology," *Human Molecular Genetics*, vol. 14, pp. R171–R181, Oct 2005.

[3] J. De Las Rivas and C. Fontanillo, "Protein–protein interactions essentials: Key concepts to building and analyzing interactome networks," *PLOS Computational Biology*, vol. 6, pp. 1–8, 06 2010.

[4] J. De Las Rivas and C. Fontanillo, "Protein–protein interaction networks: unraveling the wiring of molecular machines within the cell," *Briefings in Functional Genomics*, vol. 11, pp. 489–496, 08 2012.

[5] M. Steffen, A. Petti, J. Aach, P. D'haeseleer, and G. Church, "Automated modelling of signal transduction networks," *BMC Bioinformatics*, vol. 3, no. 1, p. 34, 2002.

[6] A. J. Enright, S. Van Dongen, and C. A. Ouzounis, "An efficient algorithm for large-scale detection of protein families," *Nucleic Acids Research*, vol. 30, pp. 1575–1584, 04 2002.

[7] X. Peng, J. Wang, W. Peng, F.-X. Wu, and Y. Pan, "Protein–protein interactions: detection, reliability assessment and applications," *Briefings in Bioinformatics*, vol. 18, pp. 798–819, 07 2016.

[8] M. Michaut *et al.*, "InteroPORC: automated inference of highly conserved protein interaction networks," *Bioinformatics*, vol. 24, pp. 1625–1631, 05 2008.

[9] S. Pitre *et al.*, "PIPE: a protein-protein interaction prediction engine based on the re-occurring short polypeptide sequences between known interacting protein pairs," *BMC Bioinformatics*, vol. 7, no. 1, p. 365, 2006.

[10] R. Jansen, H. Yu, D. Greenbaum, Y. Kluger, N. J. Krogan, S. Chung, A. Emili, M. Snyder, J. F. Greenblatt, and M. Gerstein, "A bayesian networks approach for predicting protein-protein interactions from genomic data," *Science*, vol. 302, no. 5644, pp. 449–453, 2003.

[11] Q. C. Zhang *et al.*, "Structure-based prediction of protein-protein interactions on a genome-wide scale," *Nature*, vol. 490, no. 7421, pp. 556–560, 2012.

[12] L. Lü and T. Zhou, "Link prediction in complex networks: A survey," *Physica A: Statistical Mechanics and its Applications*, vol. 390, no. 6, pp. 1150 – 1170, 2011.

[13] I. A. Kovács *et al.*, "Network-based prediction of protein interactions," *Nature Communications*, vol. 10, no. 1, p. 1240, 2019.

[14] M. S. Granovetter, "The strength of weak ties," *American Journal of Sociology*, vol. 78, no. 6, pp. 1360–1380, 1973.

[15] A. Muscoloni, I. Abdelhamid, and C. V. Cannistraci, "Local-community network automata modelling based on length-three-paths for prediction of complex network structures in protein interactomes, food webs and more," *bioRxiv*, doi:10.1101/346916, 2018.

[16] R. Sharan, I. Ulitsky, and R. Shamir, "Network-based prediction of protein function," *Molecular Systems Biology*, vol. 3, no. 1, p. 88, 2007.

[17] T. Zhou, L. Lü, and Y.-C. Zhang, "Predicting missing links via local information," *The European Physical Journal B*, vol. 71, no. 4, pp. 623–630, 2009.

[18] L. A. Adamic and E. Adar, "Friends and neighbors on the web," *Social Networks*, vol. 25, no. 3, pp. 211 – 230, 2003.

[19] C. Lei and J. Ruan, "A novel link prediction algorithm for reconstructing protein–protein interaction networks by topological similarity," *Bioinformatics*, vol. 29, pp. 355–364, 12 2012.

[20] N. Nakajima, M. Hayashida, J. Jansson, O. Maruyama, and T. Akutsu, "Determining the minimum number of protein-protein interactions required to support known protein complexes," *PLOS ONE*, vol. 13, no.4, article e0195545, pp. 1–17, 04 2018.

[21] L. Lü, C.-H. Jin, and T. Zhou, "Similarity index based on local paths for link prediction of complex networks," *Phys. Rev. E*, vol. 80, p. 046122, Oct 2009.

[22] Y. Chen, W. Wang, J. Liu, J. Feng, and X. Gong, "Protein interface complementarity and gene duplication improve link prediction of protein-protein interaction network," *Frontiers in Genetics*, vol. 11, p. 291, 2020.

[23] C. V. Cannistraci, G. Alanis-Lobato, and T. Ravasi, "From link-prediction in brain connectomes and protein interactomes to the local-community-paradigm in complex networks," *Scientific Reports*, vol. 3, no. 1, p. 1613, 2013.

[24] R. Oughtred, C. Stark, B.-J. Breitkreutz, J. Rust, L. Boucher, and C. e. l. Chang, "The BioGRID interaction database: 2019 update," *Nucleic Acids Research*, vol. 47, pp. D529–D541, 11 2018.

[25] D. Szklarczyk *et al.*, "STRING v10: protein–protein interaction networks, integrated over the tree of life," *Nucleic Acids Research*, vol. 43, pp. D447–D452, 10 2014.

[26] S. Kerrien *et al.*, "The IntAct molecular interaction database in 2012," *Nucleic Acids Research*, vol. 40, pp. D841–D846, 11 2011.

[27] L. Licata, L. Briganti, D. Peluso, L. Perfetto, M. Iannuccelli, and E. e. l. Galeota, "MINT, the molecular interaction database: 2012 update," *Nucleic Acids Research*, vol. 40, pp. D857–D861, 11 2011.

[28] K. Luck *et al.*, "A reference map of the human binary protein interactome," *Nature*, vol. 580, pp. 402–408, Apr 2020.

[29] L. Hakes, D. L. Robertson, S. G. Oliver, and S. C. Lovell, "Protein Interactions from Complexes: A Structural Perspective," *Comparative and Functional Genomics*, vol. 2007, pp. 1–5, 2007.

[30] G. Yu, F. Li, Y. Qin, X. Bo, Y. Wu, and S. Wang, "GOSemSim: an R package for measuring semantic similarity among GO terms and gene products," *Bioinformatics*, vol. 26, pp. 976–978, 02 2010.

[31] J. Z. Wang, Z. Du, R. Payattakool, P. S. Yu, and C.-F. Chen, "A new method to measure the semantic similarity of GO terms," *Bioinformatics*, vol. 23, pp. 1274–1281, 03 2007.

[32] G. Giaever *et al.*, "Functional profiling of the Saccharomyces cerevisiae genome," *Nature*, vol. 418, no. 6896, pp. 387–391, 2002.

[33] M. Lek *et al.*, "Analysis of protein-coding genetic variation in 60,706 humans," *Nature*, vol. 536, pp. 285–291, Aug 2016.

[34] H. Luo, Y. Lin, F. Gao, C.-T. Zhang, and R. Zhang, "DEG 10, an update of the database of essential genes that includes both protein-coding genes and noncoding genomic elements," *Nucleic Acids Research*, vol. 42, pp. D574–D580, 11 2013.

[35] S. Jain and G. D. Bader, "An improved method for scoring protein-protein interactions using semantic similarity within the gene ontology," *BMC Bioinformatics*, vol. 11, p. 562, Nov 2010.

[36] X. He and J. Zhang, "Why do hubs tend to be essential in protein networks?," *PLOS Genetics*, vol. 2, pp. 1–9, 06 2006.

[37] J. Wojcik and V. Schächter, "Protein-protein interaction map inference using interacting domain profile pairs ," *Bioinformatics*, vol. 17, pp. S296–S305, 06 2001.

[38] A. Podder, M. Pandit, and L. Narayanan, "Drug target prioritization for alzheimer's disease using protein interaction network analysis," *OMICS: A Journal of Integrative Biology*, vol. 22, no. 10, pp. 665–677, 2018. PMID: 30346884.