Better Network Modeling For Link Prediction In Protein-Protein Interaction Networks

Ho Yin Yuen (andy.aa.yuen@connect.polyu.hk)
Hong Kong Polytechnic University

Jesper Jansson
Hong Kong Polytechnic University

Research Article

Keywords: Protein-Protein Interaction, Link Prediction, Network Modeling, Complex Network, Graph Theory

Posted Date: October 27th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-939985/v1

RESEARCH

Better Network Modeling for Link Prediction in Protein-Protein Interaction Networks

Ho Yin Yuen[1*] and Jesper Jansson[1,2*]

A preliminary version of this article has been published in the proceedings of the 20th IEEE International Conference on Bioinformatics and Bioengineering (IEEE BIBE 2020), pp. 53-60, 2020.

*Correspondence:
andy.aa.yuen@connect.polyu.hk;
jesper.jansson@polyu.edu.hk
[1]Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China
[2]Graduate School of Informatics, Kyoto University, 606-8501, Kyoto, Japan
Full list of author information is available at the end of the article

Abstract

Background: Protein-protein interaction (PPI) data is an important type of data used in functional genomics. However, inaccuracies in high-throughput experiments often result in incomplete PPI data. Computational techniques are thus used to infer missing data and to evaluate confidence scores, with link prediction being one such approach that uses the structure of the network of PPIs known so far to find good candidates for missing PPIs. Recently, a new idea called the *L3 principle* introduced biological motivation into PPI link predictions, yielding predictors that are superior to general-purpose link predictors for complex networks. However, the previously developed L3 principle-based link predictors are only an approximate implementation of the L3 principle. As such, not only is the full potential of the L3 principle not realized, they may even lead to candidate PPIs that otherwise fit the L3 principle being penalized.

Result: In this article, we propose a formulation of link predictors without approximation that we call *ExactL3* (*L3E*) by addressing missing elements within L3 predictors in the perspective of network modeling. Through statistical and biological metrics, we show that in general, L3E predictors perform better than the previously proposed methods on seven datasets across two organisms (human and yeast) using a reasonable amount of computation time. In addition to L3E being able to rank the PPIs more accurately, we also found that L3-based predictors, including L3E, predicted a different pool of real PPIs than the general-purpose link predictors. This suggests that different types of PPIs can be predicted based on different topological assumptions and that even better PPI link predictors may be obtained in the future by improved network modeling.

Keywords: Protein-Protein Interaction; Link Prediction; Network Modeling; Complex Network; Graph Theory

1 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 signal transduction of protein physical docking [3], large-scale studies can provide insights into the molecular machinery of living systems [4]. On a basic level, researchers can abstract

biological components such as signaling pathways as a chain of PPIs [5], or protein complexes as graph clusters [6] for network analysis. In larger-scale studies, PPIs network can even be used as a building block that associates with other biological networks for better prioritization of candidate disease proteins or improved drug repurposing [7, 8].

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 accuracy of the proposed 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 [9, 10], since protein sequences compare proteins' functions genetically. Some of the other approaches include the use of protein structures, RNA co-expression, and protein annotations [11] [12]. Undoubtedly, the success of these methods stems from 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 computer networks, recommender systems, and social networks [13]. These link predictors can also be applied to PPI data, but they are usually not specific enough to characterize PPIs well, and there are no guarantees on their correctness and reliability. Due to this concern, Kovács *et al.* [14] 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. The study 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 [15] that favors paths of length two. Since then, studies have already successfully improved existing network biology techniques by incorporating the L3 principle, including drugs-disease network analysis [16] and protein fold recognition [17].

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 [18]. 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 one has a different way of addressing the issue. However, the penalization in the L3 link predictor is applied 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 neighborhoods of PPIs 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 article, we define the link predictor in a way that more accurately corresponds to the biological motivation of the L3 principle. Our approach is coined *ExactL3*

($L3E$). We show experimentally that L3E is better at inferring unknown PPIs than the previous methods, which gives further evidence that the network structure of PPIs can be accurately reconstructed from partial data.

We would like to remark here that the preliminary conference version of this article [19] contains an error in the presented formula for $P_{xy}^{(L3E)}$ (Formula (4) in [19]) and that the experimental results were obtained using a slightly different (and correct) version of the formula that was implemented as intended in the program code that was provided. In this article, we have corrected the error and also generalized the formula to further improve the performance of our link predictor; see Formulas (6) and (7) in Section 4.2 below.

The article is organized as follows. Section 2 reviews some known general and PPI-specific link prediction techniques. Then, we provide the problem definition and the formulation of L3E in Section 3 and Section 4, respectively. Using the materials described in Section 5, we evaluate how well these link predictors perform in synthetic datasets of certain structures in Section 6. In Section 7, we then evaluate the predictive power and biological significance of L3E using statistical and biological metrics. Finally, in Section 8, we discuss the underlying differences between L3E and other link predictors and give some ideas for potential future improvements.

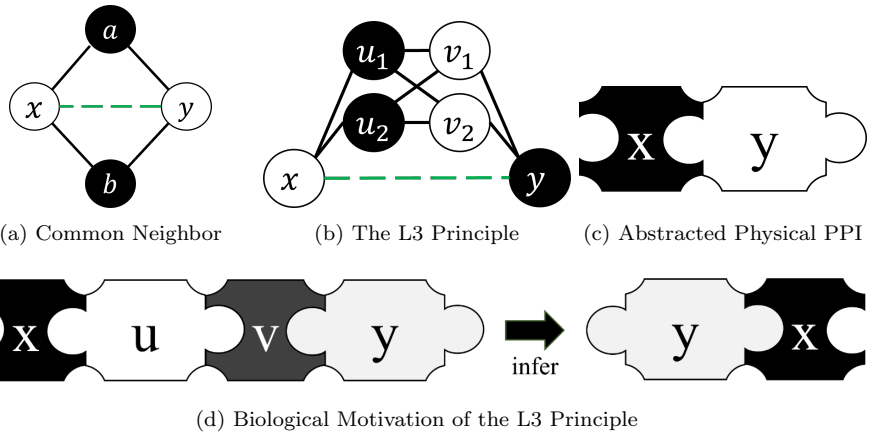
2 Previous Work

Link prediction infers new edges based on the properties of the nodes as well as the overall topology of the existing edges [13]. Many classes of link prediction approaches exist, and this article will focus on *similarity-based* link predictions, where candidate edges are selected based on the similarity of nodes' immediate or extended neighborhoods. Some link predictors of this type are reviewed next. From here on, for any node a, let $N(a)$ denote the set of neighbor nodes of a, and for any set A of nodes, let $N(A) = \bigcup_{a \in A} N(a)$.

2.1 General Link Prediction

The Common Neighbors (CN) concept originates from social networks [15]. It models a social phenomenon: the more friends two individuals share, the more likely they are to also be friends of each other. Then, the CN score of any two nodes a and b is $|N(a) \cap N(b)|$. The assumption here is that 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 that they have similar functions [20]. That is, if two proteins interact with a similar set of proteins then their functions should be similar.

However, a high-degree node will contribute to the CN scores of many more node pairs than a low-degree node will. Consequently, to reduce the influence that a single node may have, it is a good idea to penalize high-degree nodes in the CN index. To do so, the Resource Allocation (RA) algorithm [21] 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 [22], a logarithmic 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 a survey of the normalization schemes used in many other general link predictors, see [13].



(a) Common Neighbor (b) The L3 Principle (c) Abstracted Physical PPI

(d) Biological Motivation of the L3 Principle

Figure 1: Principles behind link prediction in PPI networks. (a) and (b) illustrate conditions that would lead CN and the L3 principle, respectively, to predict that an edge between the two non-adjacent nodes x and y is in fact missing. (c) A graphical representation of the occurrence of a physical PPI between protein x and protein y. (d) Using the abstraction in (c), if the PPIs are arranged as shown on the left, we can infer the existence of a PPI between protein y and protein x as shown on the right.

2.2 PPI-Specific Link Prediction

Link predictors can also consider parts of the network beyond the immediate neighborhoods of nodes. For example, in the context of PPI networks, [23] applies random walks to identify and connect pairs of nodes with similar distances to the other nodes in the network. This can be classified as a global approach in similarity-based link predictions.

In another study, Nakajima *et al.* [24] used 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, in the study of our focus [14], Kovács *et al.* presented the so-called *L3 algorithm*, which 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_3 will denote an undirected length-2 path consisting of three nodes and two edges, and P_4 will denote an undirected length-3 path consisting of four nodes and three edges. Using this notation, the observation above can be stated as: the more P_4-subgraphs and the fewer P_3-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. From here on, we shall refer to this principle as the *L3 principle*.

After the L3 principle was proposed [14], other researchers have also taken inspiration from it to formulate better link predictors for PPI networks. This includes *CH2_L3* (abbreviated as *CH2* below) [18], a link predictor that extends the general link predictor *CRA* [25], as well as the *Sim* [26] link predictor. Both of these are similarity-based link predictions that use information from immediate neighborhoods, just like our method L3E. For this reason, they are also included in the experimental comparison below. The mechanisms of CH2 and Sim are described in more detail in Section 4.1 and Section 5.2.

3 Preliminaries

3.1 Problem Definitions

Given an undirected graph $G = (V, E)$, the task is to determine, for each pair of non-adjacent nodes in V, 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 2.2, 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). This kind of double summation will be abbreviated as in 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 2.1, in this formula, 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 link predictor [14] does this by using a square root modifier according to Formula (3) below.

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

3.2 Our Contributions

We observe that the normalization modifier in Formula (3) does not completely implement the L3 principle. More precisely, Formula (3) only uses the set U, the

set V, and the node degrees to evaluate an xy-node pair. It does not take P_3-subgraphs into account and may penalize xy-links that are highly likely despite having high-degree nodes u and v. *ExactL3* addresses these problems by employing an alternative approach to normalization. Before presenting the formulation, we first give more intuition behind the L3 principle introduced in Section 2.2.

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, where each P_4 can be represented as (x, u, v, y). This can be condensed into one central idea, that the size of $N(y)$ reflects the compatibility of the binding interfaces of x and y (the same idea applies analogously to $N(x)$). In the original L3 predictor, counting P_4's enables the evaluation because in a P_4, node u represents the interface of y, and $u \in N(x)$; thus, node u provides evidence that x is compatible with y (and analogously for v). In contrast, L3E performs link predictions more accurately by directly evaluating the ratio of compatible and incompatible nodes in the neighborhoods. For example, one should penalize the score P_{xy} if either x or y has neighbors that cannot form a P_4, and reward it otherwise. In the next section, we formulate the *ExactL3* link predictor.

4 Detailed Formulation of ExactL3

To describe the properties that characterize the L3 principle, we define an *ideal L3 graph* G_{L3} as a graph that can be obtained by taking a complete bipartite graph with two parts U and V, and attaching a new node x as a neighbor of all nodes in U and attaching a new node y as neighbor of all nodes in V. This results in a graph with the four basic *L3-elements*: node x, node y, set U, and set V, which are the fundamental components of an ideal L3 graph. Fig. 2(a) illustrates an example ideal L3 graph. Its nodes have been colored white and black in such a way that no pair of nodes with the same color are adjacent and every pair of nodes with different colors (except x and y) are adjacent.

To model real PPI networks, we need to consider *non-ideal L3 graphs* that can deviate from ideal L3 graphs in the following ways:

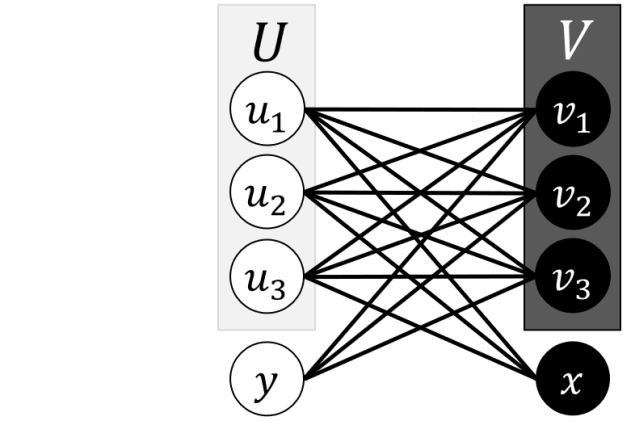
- An edge between x and U is missing, or an edge between y and V is missing.
- An edge between U and V is missing.
- An edge between two nodes in U, or between two nodes in V, exists.
- An edge between x and V exists, or an edge between y and U exists.

Recall that we defined $U = N(x) \cap N(N(y))$ and $V = N(y) \cap N(N(x))$ in Section 3. These definitions induce, for any specified pair of nodes x and y, the L3-elements of an L3 graph whose fitness to the L3 principle can be evaluated by measuring how well the following conditions are met:

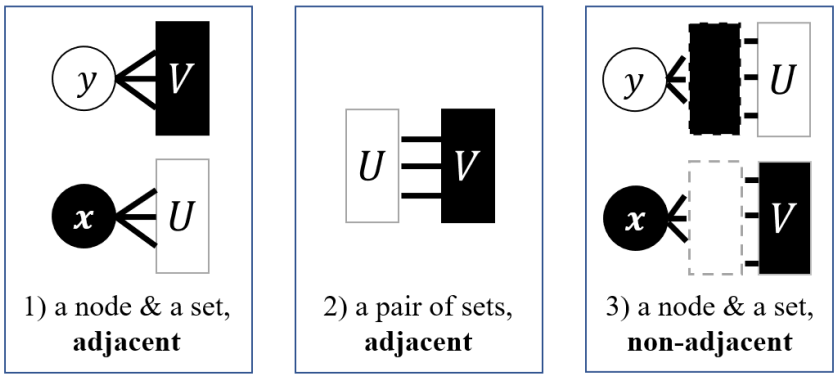
I $N(x) = U$ and $N(y) = V$ (see Fig. 2(b)-1)

II $N(v) \setminus \{y\} = U$ for every $v \in V$ and
$N(u) \setminus \{x\} = V$ for every $u \in U$ (see Fig. 2(b)-2)

III $N(x) = N(v) \setminus \{y\}$ for every $v \in V$ and
$N(y) = N(u) \setminus \{x\}$ for every $u \in U$ (see Fig. 2(b)-3)



(a)



(b)

Figure 2: Illustrating the conditions that would lead the L3 principle to assign the maximum possible score to P_{xy}. (a) An example of an ideal L3 graph with the four L3-elements x, U, V, and y. (b) Three measures of how well a (possible non-ideal) L3 graph fits the L3 principle based on its L3-elements.

As an example, consider a non-ideal L3 graph obtained by inserting a single edge of the form $\{u_i, u_j\}$ into an ideal L3 graph. Then, $N(u_i) \setminus \{x\} \neq V$ and $N(u_j) \setminus \{x\} \neq V$ hold, thus violating condition II. Also, $N(y) \neq N(u_i) \setminus \{x\}$ and $N(y) \neq N(u_j) \setminus \{x\}$, which violates condition III. However, this graph is still quite close to being an ideal L3 graph. To quantify how well conditions I, II, III are met, we introduce two similarity metrics in the next subsection.

4.1 Similarity Metrics

Similarity metrics are formulas that score the similarity of two sets with appropriate penalization so that the size of the two sets has a minimum effect on the score. In the case of PPI networks, the sets would be node subsets such as the neighborhood of a node. Such metrics allow us to formalize the relationships in Figure 2(b) as mentioned above. In the following sections, we review two well-studied similarity metrics that will be included in our improved link predictor. (See the summary in Table 2 in Section 5.2 for their precise formulas.)

4.1.1 Simple Ratio

Given two sets A and B, one of the simplest possible metrics is the *Simple Ratio* in Formula (4), which measures the size of the intersection relative to the size of one of the sets.

$$f_1(A, B) = \frac{|A \cap B|}{|A|} \tag{4}$$

To give an example, the *CRA* link predictor [25] utilizes this to extend the CN principle for general link prediction (including PPI networks). CRA computes the link prediction score of node x and y by first extracting the common neighbors, $A = N(x) \cap N(y)$. Then, each node $a \in A$ is evaluated according to $f_1(N(a), A)$. The sum of these scores, which is $\sum_{a \in A} f_1(N(a), A)$, will then be the link prediction score for nodes x and y. It is defined in this way because CRA is only interested in if $N(a)$ is a subset of A, regardless of the size of set A.

4.1.2 Jaccard coefficient

Formula (5) is the *Jaccard coefficient* [27] for set A and set B. Note that it uses a different denominator than the one in Section 4.1.1.

$$f_2(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{5}$$

This evaluation assumes that both sets are equally important and the maximum possible score can only be obtained when $A = B$. (In comparison, in Formula (4) in Section 4.1.1, the best score can be obtained even if $A \subsetneq B$ or $B \subsetneq A$.) This idea is utilized in the *Sim* link predictor [26]. To be precise, Sim independently scores the similarity of node x and nodes v using $f_2(N(x), N(v))$, and node y and nodes u using $f_2(N(y), N(u))$. The summation of these scores then become the link prediction score for the corresponding node x and y.

4.2 ExactL3 Formulations

Using any similarity metric f, we can quantify how close a non-ideal L3 graph is to being ideal by accounting for conditions I, II, and III described at the beginning of this section as follows:

- Condition $N(x) = U$: use $f(N(x), U)$
- Condition $N(y) = V$: use $f(N(y), V)$
- Condition $N_{\neg x}(u) = V$ for every $u \in U$: use $\sum_U f(N_{\neg x}(u), V)$
- Condition $N_{\neg y}(v) = U$ for every $v \in V$: use $\sum_V f(N_{\neg y}(v), U)$
- Condition $N(x) = N_{\neg y}(v)$ for every $v \in V$: use $\sum_V f(N(x), N_{\neg y}(v))$
- Condition $N(y) = N_{\neg x}(u)$ for every $u \in U$: use $\sum_U f(N(y), N_{\neg x}(u))$

where the notation $N_{\neg b}(a)$ is a shorthand for $N(a) \setminus \{b\}$. Then, we complete the formulation by combining them as in Formula (6). We formulate the link prediction score as a sum taken over all pairs of nodes (u, v) for $u \in U$ and $v \in V$ since each P_4 that increases the likelihood of the edge between x and y corresponds to one

such (u, v). Note that $f(N(x), U)$ and $f(N(y), V)$ can be evaluated outside of the inner sum since they do not depend on both u and v at the same time.

$$P_{xy}^{(L3E(f))} = f(N(x), U) \cdot f(N(y), V) \cdot \sum_{U,V} f(N_{\neg x}(u), V) \cdot f(N_{\neg y}(v), U) \cdot f(N(x), N_{\neg y}(v)) \cdot f(N(y), N_{\neg x}(u)) \tag{6}$$

Fig. 3 gives a graphical explanation of Formula (6) using the similarity metric f_1 from Section 4.1.1. From now on, the link predictor obtained by letting $f = f_1$ in Formula (6) will be denoted by L3E(f_1); similarly, plugging in f_2 from Section 4.1.2 into Formula (6) gives a link predictor that we will refer to as L3E(f_2). To illustrate the L3E formulation with an example, consider the non-ideal L3 graph mentioned previously in this section that was obtained by inserting a single edge of the form $\{u_i, u_j\}$ into an ideal L3 graph. For this graph, $u_j \in N(u_i)$ although $u_j \notin V$, which means that $N_{\neg x}(u_i)$ and V are not completely identical and the third term in Formula (6) will be slightly smaller than its maximum possible value. Moreover, the fact that $N_{\neg x}(u_j) \neq V$ will also contribute to the third term not being maximized, and $N(y) \neq N_{\neg x}(u_i)$ and $N(y) \neq N_{\neg x}(u_j)$ will prevent the sixth term from being maximized.

Formula (6) uses neighborhoods with the node x or y excluded (e.g., $N_{\neg x}(u)$). For normalization purposes, it may in fact be advantageous to include x or y in the neighborhoods. Similar modifications appear explicitly in CH2 predictors [18], where an offset of one is appended as compensation, and implicitly in L3 [14] and CRA [25]. To see why the normalization might be useful, suppose that we are evaluating (x, y) and that $N(u_i) = V_i \cup \{x\}$ and $N(u_j) = V_j \cup \{x\}$ for two nodes $u_i, u_j \in U$ and $V_i, V_j \subseteq V$ with $|V_i| < |V_j|$. Then $f(N_{\neg x}(u_i), V_i) = f(N_{\neg x}(u_j), V_j)$ is possible although it would be better to have $f(N_{\neg x}(u_i), V_i) < f(N_{\neg x}(u_j), V_j)$ because the larger size of V_j provides stronger evidence that u_j and V_j are compatible. Here, if we use neighborhoods that include x then we would get $f(N(u_i), V_i) < f(N(u_j), V_j)$, which might be preferable. Formula (7) below introduces an alternative L3E formulation based on this observation, which we shall refer to as L3E' in the experiments in later sections.

$$P_{xy}^{(L3E'(f))} = f(N(x), U) \cdot f(N(y), V) \cdot \sum_{U,V} f(N(u), V) \cdot f(N(v), U) \cdot f(N(x), N(v)) \cdot f(N(y), N(u)) \tag{7}$$

In particular, L3E'(f_1) and L3E'(f_2) will refer to the link predictors obtained by selecting $f = f_1$ and $f = f_2$, respectively, in Formula (7).

4.3 Time Complexity

Here, the computational complexity of L3 (evaluating Formula (3)) and L3E' (evaluating Formula (7)) will be analyzed. The analysis of the latter also applies to L3E. Let n denote the number of nodes in G. In comparison, the CN link predictor is known to run in $O(n^3)$ time [28].

The main operations in both L3 and L3E' are the set operations on node neighborhoods. We first discuss the two set operations included in our formulation, the

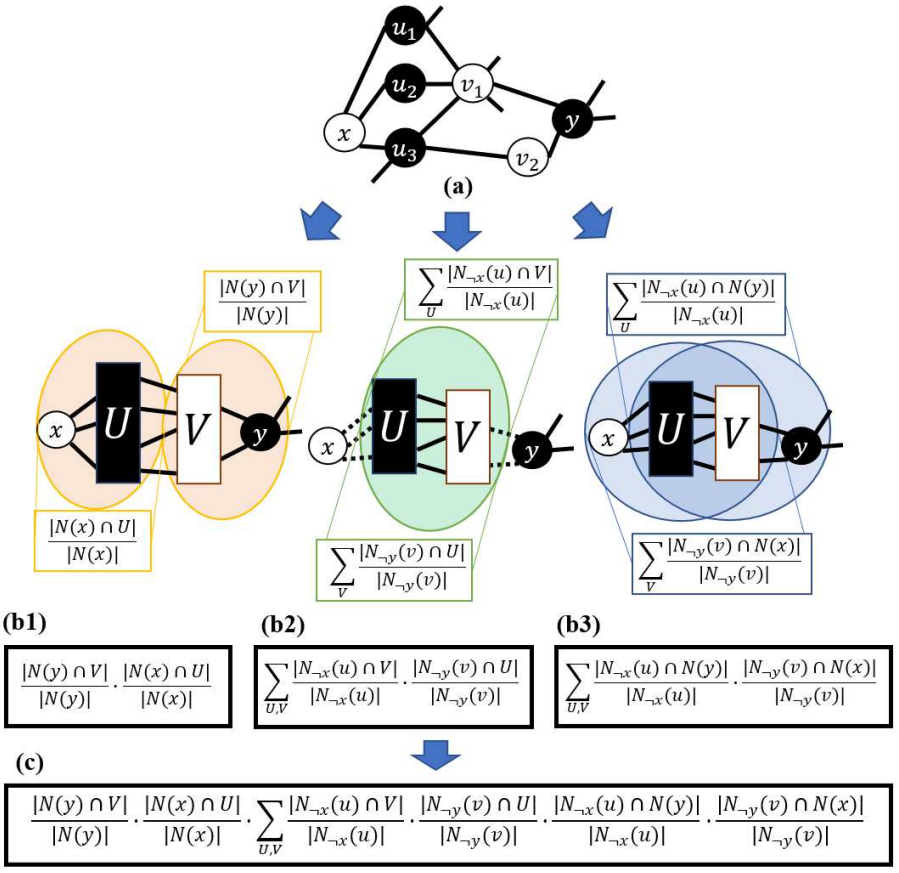


Figure 3: The idea behind Formula (6). (a) An L3 graph with L3-elements x, U, V, and y. (b1)–(b3) Each of the six parts of Formula (6) corresponds to one of the six conditions used to measure how close the graph is to being an ideal L3 graph. Here, the Simple Ratio f_1 from Section 4.1.1 has been selected as the similarity metric. (c) Combining all six parts yields Formula (6) for the score $P_{xy}^{L3E(f_1)}$.

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, computing $A \cap B$ takes $O(\min(|A|, |B|))$ time. For the set union operation $A \cup B$, one has to access all elements of both sets if the 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 L3E' 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. In the case of f_2, the normalization uses set intersection and set union operations and takes

$O(\min(|N(v)|, |N(x)|) + (|N(v)| + |N(x)|) + \min(|N(u)|, |N(y)|) + (|N(u)| + |N(y)|) + \min(|N(v)|, |U|) + (|N(v)| + |U|) + \min(|N(u)|, |V|) + (|N(u)| + |V|)) = O(n)$ time by the above. In the case of f_1, the time complexity is upper bounded by the time complexity for the case of f_2. In summary, the time complexity of L3E' is $O(n^5)$.

5 Materials

In this section, we will give a brief overview of the PPI datasets that were used in our experiments and the other link predictors that were compared to L3E predictors.

Saccharomyces cerevisiae (Yeast)				
Dataset \Number of	Raw Size (MB)	Nodes	PPIs	Candidate PPIs
BioGRID	316	7,085	113,116	20,045,849.4
STRING	85.5*	4,673	94,529	9,212,026.6
MINT	38.3	4,049	16,927	5,980,266.7
Homo sapiens (Human)				
Dataset \Number of	Raw Size (MB)	Nodes	PPIs	Candidate PPIs
BioGRID	166	24,760	452,684	220,833,040.0
STRING	717*	15,668	308,614	88,982,499.1
MINT	55.0	7,534	22,324	15,493,875.9
HuRI	161	8,109	51,127	21,899,033.2

Table 1: Overview of the PPI datasets used in the experiments. An asterisk denotes the combined file size of multiple essential metadata files.

5.1 Datasets

Our experiments used seven real PPI datasets from two organisms, the well-known model yeast (*Saccharomyces cerevisiae*, strain S288C) and human (*Homo sapiens*). We included multiple datasets for the same organism because the methodology used to obtain them and their confidence thresholds often differ [29]. Six of the datasets were integrated PPI datasets from three different literature sources (BioGRID [30], STRING [31], and MINT [32]) where both the yeast and the human variations were considered for each one. The seventh dataset that we used was HuRI [33], a more recent human PPI dataset obtained from a single experimental source.

We used the datasets' annotations to extract physical PPIs (binary PPIs) only as follows: 'physical' for BioGRID; 'binding' for STRING; 'direct interaction', 'physical association', and 'association' for MINT. (All PPIs in HuRI are physical PPIs.) Next, every directional PPI was converted into a non-directional PPI, and all duplicate PPIs (due to multiple evidence in the literature) as well as all self-interactions were excluded. The number of nodes, PPIs, and candidate PPIs for each of the seven processed datasets is listed in Table 1.

5.2 Link Predictors

L3E predictors, using each of the two similarity metrics f_1 and f_2 from Section 4.1, was compared to five other link predictors in the literature, with an extra negative-control predictor that selected PPIs uniformly at random. Table 2 summarizes the link predictors used in the experiments. The mechanism of each link predictor is as elaborated in the previous sections: among the *CN-based* link predictors, CN infers edges according to the principle shown in Fig. 1(a) and CRA infers edges using the f_1 similarity metric as explained in Section 4.1.1, while among the *L3-based*

link predictors, L3 infers edges based on the principle shown in Fig. 1(b), Sim infers edges using the f_2 similarity metric defined in Section 4.1.2, and CH2 rewards edges for which the nodes in U and V are connected to many other nodes in $U \cup V$ but not connected to many nodes outside of $U \cup V$.

Type	Link predictor	Score function $P_{xy} =$
CN-based	Common Neighbors (CN) [15]	$\lvert N(x) \cap N(y)\rvert$
	CRA [25]	$\sum_{a \in A} \frac{\lvert N(a) \cap A\rvert}{\lvert N(a)\rvert}$
L3-based	L3 [14]	$\sum_{U,V} \frac{1}{\sqrt{\lvert N(u_i)\rvert \cdot \lvert N(v_j)\rvert}}$
	CH2_L3 (CH2) [18]	$\sum_{U,V} \frac{\sqrt{(1+\lvert N(u) \cap c\rvert) \cdot (1+\lvert N(v) \cap c\rvert)}}{\sqrt{(1+\lvert N(u) \setminus c) \setminus \{x,y\}\rvert) \cdot (1+\lvert N(v) \setminus c) \setminus \{x,y\}\rvert)}}$
	Sim [26]	$\sum_V \frac{\lvert N(v) \cap N(x)\rvert}{\lvert N(v) \cup N(x)\rvert} + \sum_U \frac{\lvert N(u) \cap N(y)\rvert}{\lvert N(u) \cup N(y)\rvert}$
	ExactL3 (L3E) predictors	Plug in either f_1 or f_2 into Formula (6) or (7)
control	rand	Rank the edges uniformly at random

Table 2: Overview of the link predictors used in the experiments. In the table, $A = N(x) \cap N(y)$ and $c = U \cup V$. (For the other definitions, refer to Sections 2 and 3.)

6 Link Prediction in Synthetic Datasets

In this section, we present the results of our first set of experiments, designed to test how well the L3E link predictors realized the L3 principle compared to the other predictors in Table 2 on some synthetic datasets. The rand link predictor is not considered here since it cannot generate a link prediction score, so we use an alternative control predictor that simply counts the number of P_4's between the two given nodes x and y instead. To generate the synthetic datasets, we start with an ideal L3 graph G (recall the definitions from Section 4) having 50 nodes in U and 50 nodes in V. Then, in the experiments, we add or remove edges from G that induce changes in the scores computed by the link predictors. By modifying an ideal L3 graph in this way, we can see how sensitive each link predictor is when dealing with changes that make G diverge from its ideal form. Since different link predictors use different scales, we normalize all their scores to values between zero and one. From here on, an edge of the form $\{u_i, v_j\}$, where $u_i \in U$, $v_j \in V$, and $i \neq j$, will be referred to as a *compatible edge*. Similarly, an edge of the form $\{u_i, u_j\}$ where $u_i, u_j \in U$ and $u_i \neq u_j$, or of the form $\{v_i, v_j\}$ where $v_i, v_j \in V$ and $v_i \neq v_j$, or of the form $\{x, v_i\}$ where $v_i \in V$, or of the form $\{y, u_i\}$ where $u_i \in U$, is called an *incompatible edge*.

6.1 Removing Compatible Edges

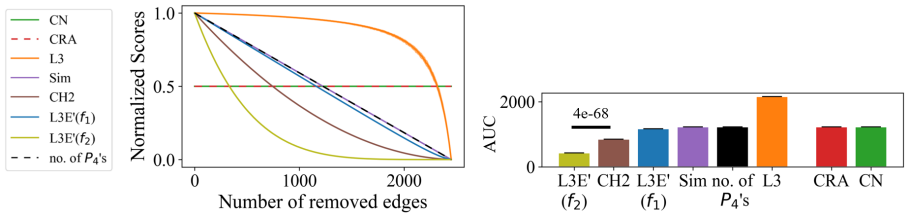
Our first experiment started with the ideal L3 graph G and removed one of the compatible edges, chosen uniformly at random, from G in each iteration until all the $(50 \cdot 50) - 50 = 2450$ compatible edges had been removed. Since the 50 edges of the form $\{u_i, v_i\}$ were never removed, the four L3-elements x, y, U, and V remained the same throughout the experiment. In every iteration, P_{xy} for each link predictor were computed. We repeated the above ten times and calculated the median, minimum, and maximum scores to capture the variance. The results are

plotted in Fig. 4(a). (The results for L3E are plotted separately in Fig. S1(a) since they overlap with L3E'.) As can be seen by looking at the curve for the control predictor, the number of P_4's decreases as the number of remaining compatible edges decreases. This implies that the scores for a link predictor that realizes the L3 principle should decrease as well, and that to be more sensitive than the control predictor in score penalization, a link predictor's area under curve (AUC) should be smaller than that of the control predictor. In this regard, the predictor L3E'(f_2) outperformed all the other predictors, and CH2 and L3E'(f_1) also did quite well. The same applies for L3E(f_1) and L3E(f_2) in Fig. S1(a). Note that CN and CRA have a constant score throughout the iteration: since x and y have no common neighbors in G, the scores computed by CN and CRA never change when edges are removed. For L3, its underwhelming performance can be attributed to the following: in early iterations, many pairs of nodes from U and V contribute to the score, and since these nodes have a high node degree, each pair has a low L3 score. Their individual contributions are consequently very small, which means that when one edge is deleted, the score computed by L3 remains close to its initial score. In contrast, in later iterations, few pairs of nodes from U and V contribute (and these nodes have a lower degree), so deleting an edge affects the score more.

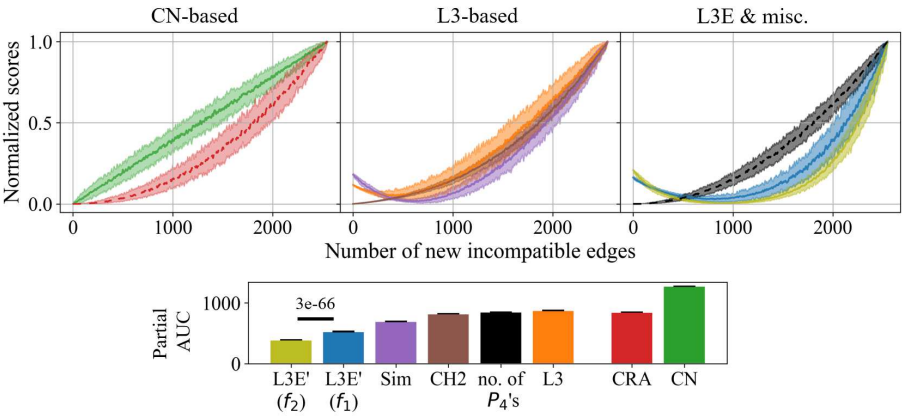
6.2 Adding Incompatible Edges

The second experiment was complementary to the one in Section 6.1. Starting from the ideal L3 graph G, one incompatible edge was inserted into G in every iteration until all the $\binom{50}{2} + \binom{50}{2} + 50 + 50 = 2550$ incompatible edges had been inserted. Each edge to be inserted was chosen uniformly at random among the incompatible edges that had not been inserted yet. The experiment was repeated ten times, as in Section 6.1.

The results are plotted in Fig. 4(b). (As above, the results for L3E are plotted separately in Fig. S1(b) since they overlap with L3E'.) In this experiment, one might expect to see strictly decreasing scores as additional edges are inserted into G, disrupting its ideal L3 structure. However, as shown by the control predictor, the addition of incompatible edges increases the number of P_4's non-linearly because the more edges that already exist in G, the more P_4's between x and y will be created for each additional edge. Therefore, any L3-based predictor will eventually show an increasing score. Yet, L3-based predictors with proper penalization should still be less sensitive than the control predictor. By this, we mean that for a link predictor, the partial AUC starting from the point on the x-axis where its minimum normalized score occurs should be smaller than that of the control predictor. Here, L3E'(f_1) and L3E'(f_2) outperformed all the other predictors: all L3 predictors show an initially decreasing score as explained, but L3E is the least sensitive during the increase in scores as demonstrated by it having the smallest partial AUC. The same applies for L3E(f_1) and L3E(f_2) in Fig. S1(b). For CN-based predictors, the scores are not directly related to the number of P_4's, but rather the number of common neighbors created by incompatible edges of the form $\{y, u_i\}$ and $\{x, v_i\}$. (Thus, we do not compare them to the control predictor.) Here, CRA is better than CN since it compensates for interconnectedness within common neighborhoods by also taking edges of the form $\{u_i, u_j\}$ and $\{v_i, v_j\}$ into account.



(a) Removing compatible edges (Section 6.1)



(b) Adding incompatible edges (Section 6.2)

Figure 4: Changes in scores for different link predictors when an ideal L3 graph is modified by: (a) removing compatible edges; and (b) adding incompatible edges. The shaded regions denote the variance (the minimum and maximum values) among replication experiments, and the solid lines denote the medians. The AUC bar charts correspond to the respective plots. The statistical bars denote the statistical significance (p-value) between the two corresponding samples using student's t-test. In (b), a Savitzky–Golay filter using a polynomial of degree 3 and a window size of 21 was applied to make the curves smoother.

6.3 Summary

We conclude that the L3E predictors are able to penalize the absence of compatible edges as well as the presence of incompatible edges better than the other link predictors that were considered in the experiments, which suggests that L3E provides a more accurate implementation of the L3 principle for this particular kind of synthetic data. In the next section, we will evaluate the performance of L3E on some real PPI datasets.

7 Link Prediction in PPI Datasets

In this section, we experimentally evaluate the predictive power and biological significance of L3E using real PPI datasets. The datasets and link predictors used are described in Section 5. To evaluate the predictive power of L3E, we prepared our datasets by removing 50%, 40%, 30%, 20%, and 10% of the edges chosen uniformly at random (different sample sizes), and repeated this ten times for each sample size, yielding a total of 50 samples for each dataset. Next, for each of the 50 samples and each link predictor, we computed the scores of all non-neighboring pairs of nodes x and y (called *candidate edges*) and ranked them according to their scores. In each

such experiment, we then selected the k top-ranked candidate edges to be the set of predicted edges, where k denotes the number of edges that were removed from that dataset in the sampling preprocessing step. (In other words, the accuracy would be 100% if and only if the predicted edges were exactly those that had been removed earlier.) Finally, the performance of the various link predictors was evaluated by analyzing and comparing the sets of edges that they predicted in the experiments.

7.1 ExactL3 Improves PPI Link Predictions

A standard tool for evaluating statistical predictors is the precision-recall (PR) curve [34]. Using the true-positive PPIs (tp), the false-positive PPIs (fp), and the false-negative PPIs (fn) in the outcomes of the experiments, *precision* is then defined as $\frac{tp}{tp+fp}$ and *recall* is defined as $\frac{tp}{tp+fn}$. As implied by the name, a PR curve consists of a link predictor's precision- and recall-values computed for various datasets, and thus illustrates the trade-off between precision and recall. In general, the larger the area under the precision-recall curve (also referred to as the PR AUC), the better [35].

In our experiments, we first considered the datasets in which 50% of the PPIs had been removed. Fig. 5 shows the PR curves and PR AUC-values of the link predictors. Due to L3E' and L3 having a similar performance, with L3E' being slightly better than L3E (see Fig. S2 for a detailed comparison), only the former is included in Fig. 5. For the same reason, we shall focus on L3E' in the experiments from now on. According to the figure, L3E'(f_1) is generally the best predictor both among L3E predictors and among other predictors in the sense of having a PR curve that upper-bounds those of the other link predictors most of the time and having the largest PR AUC; the only exception is the STRING Yeast dataset in Fig. 5(a2).

To ensure the proper design of our methodology, we employed the random link predictor (rand) as a negative control. However, the probability of randomly choosing a real PPI from any of the pools of candidate PPIs in the datasets summarized in Table 1 is roughly at most 1%, and so the PR AUC of rand is almost 0 (see Fig. S3). Because of its insignificance, the rand predictor was therefore excluded from Fig. 5. We also computed the p-value of the PR AUC for all the predictors against rand in Tables S1 and S2, which confirmed that all the predictors are statistically significant and thus far better than selecting PPIs at random. (The largest p-value was 1e-13, i.e., far from statistically insignificant.)

Next, we conducted the same experiments for the other datasets, in which 40%, 30%, 20%, and 10% of the edges had been removed. The computed PR curves and PR AUC-values are plotted in Figs. S4–S7. The outcomes are similar to that of the experiment described above, where 50% of the edges had been removed. To give a summary of Figs. S4–S7, we extracted the PR AUC for each of the predictors in the experiments, and plotted them in Fig. 6 to show the changes in PR AUC as the number of edges removed in the dataset decreases. As in Fig. 5, L3E'(f_1) outperforms all the other link predictors in most datasets with high statistical significance. Another observation is that the PR AUC along the x-axis decreases, which may be because of the rapid drop in precision-recall or the drop in maximum recall as the percentage of removed edges in the datasets decreases

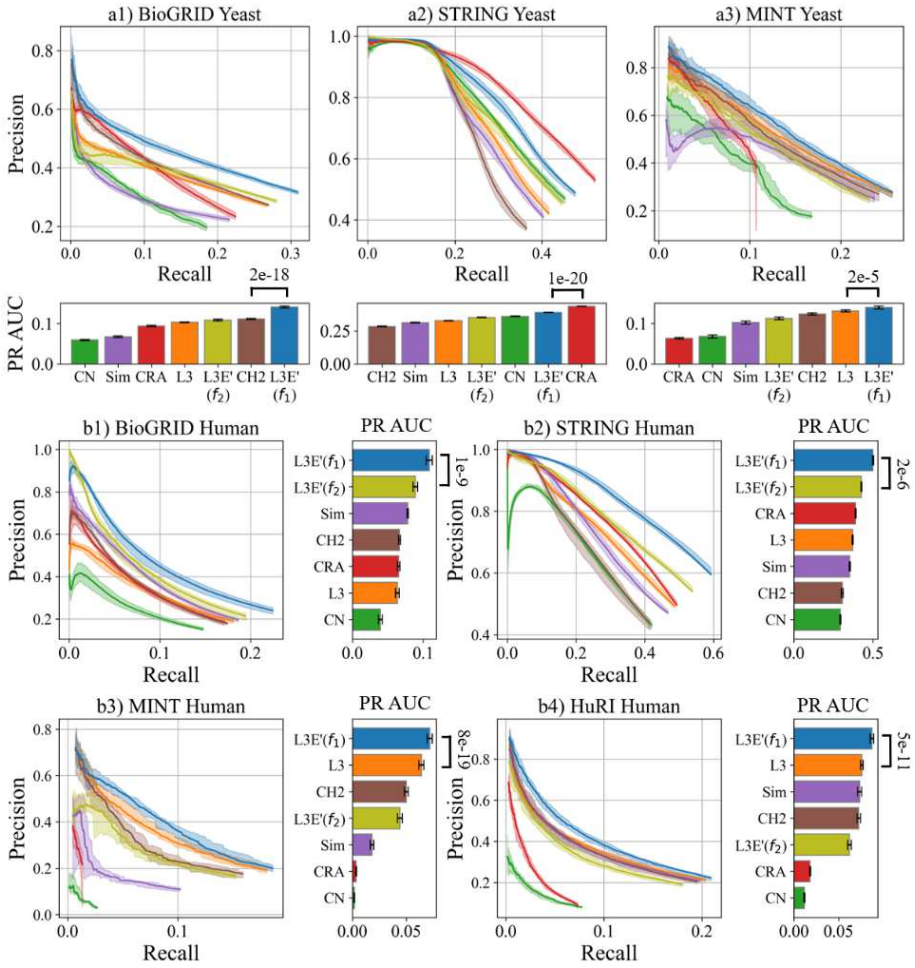


Figure 5: (Section 7.1) Precision-Recall (PR) curves of the link predictors computed with 50% of the PPIs removed in every dataset. The solid lines show the median values and the shaded regions indicate the variance (the minimum and maximum values). The accompanying bar charts show the predictors' PR AUC-values (the larger, the better), and the statistical bars denote the statistical significance between two samples in terms of the p-value using student's t-test. Observe that the PR curve of L3E'(f_1) dominates the other PR curves most of the time.

(see Figs. S3–S7). To investigate the reason for this, we evaluated the PR AUC of the random predictor as a negative control (Table S3). There is a gradual decrease in the PR AUC as the number of removed PPIs decreases, suggesting that if fewer PPIs are removed then it is more difficult for a predictor to pick a real PPI at random.

In addition to the predictive power, another important aspect to consider in the evaluation of link predictors is the computation time. Table 3 summarizes the computation times taken by the experiments in Fig. 5. The experiments were conducted using a setup of 14 cores and 32GB RAM. A larger setup consisting of 24 cores and 128GB RAM was used for the BioGRID Human and STRING Human datasets due to their massive size. For L3E', the computation time increases more rapidly than for the simpler predictors CN, L3, and CRA as the datasets scale up (e.g., BioGRID

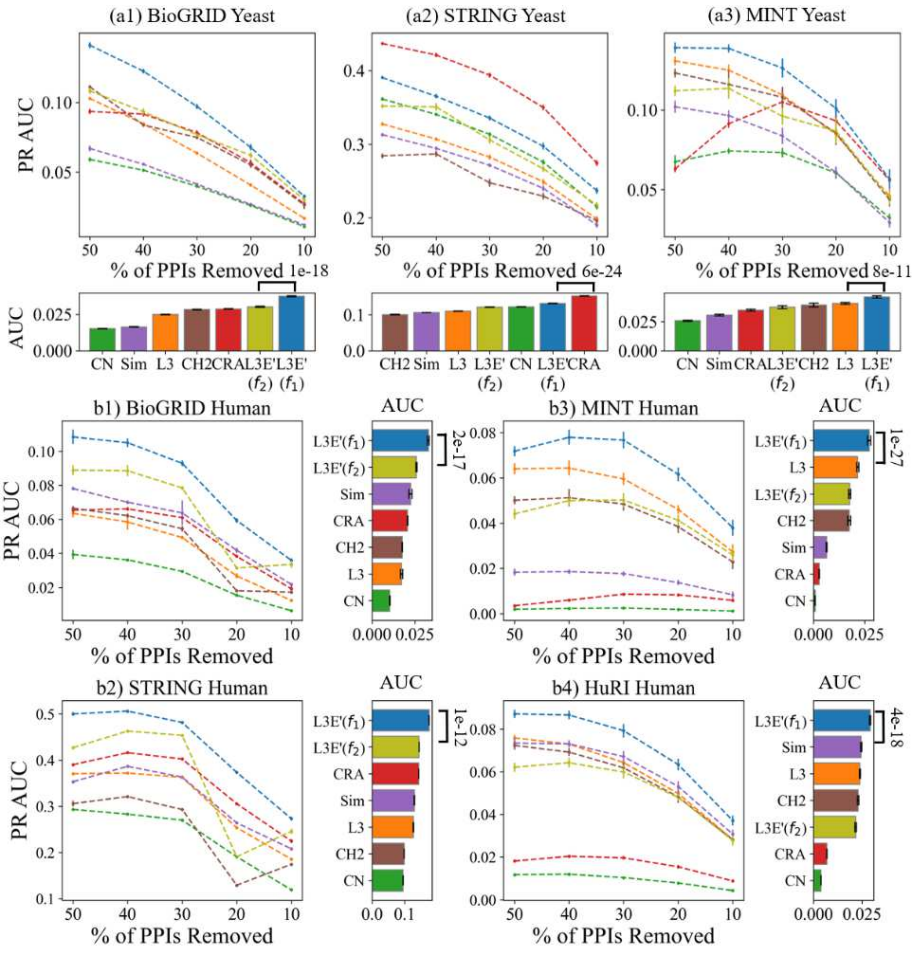


Figure 6: (Section 7.1) Illustrating how the PR AUC of the link predictors changes as the percentage of PPIs removed from the datasets decreases. The dotted curves are interpolations of the data points (50%, 40%, 30%, 20%, and 10%) and the vertical bars on the data points indicate the variance (the minimum and maximum values). The bar charts show the AUCs of the PR AUCs, i.e., the total area under each dotted line, and the statistical bars give the statistical significance between two samples in terms of the p-value using student's t-test. According to these experiments, L3E'(f_1) is better than the other link predictors since its PR AUC curve dominates the others almost everywhere.

Yeast and Human), in accordance with the time complexity analysis in Section 4.3. Yet, if L3E' is compared to other non-trivial L3-based predictors (CH2, Sim), it can be seen that L3E'(f_1) is able to obtain larger AUC-values (sometimes twice as large, according to Fig. 5) using much less time than CH2 and roughly the same amount of time as Sim. As for L3E'(f_2), it's slower in practice than L3E'(f_1) although both methods have the same theoretical time complexity (see Section 4.3).

Overall, the above findings lead us to conclude that L3E'(f_1) has the best predictive power (in terms of precision-recall across datasets of different sample sizes) using a reasonable amount of computation time.

CN	CRA	L3	CH2_L3	Sim	**L3E'(f_1)**	**L3E'(f_2)**
BioGRID Yeast						
2.65 ± 0.06	2.51 ± 0.03	5.76 ± 0.08	46.9 ± 1.54	12.4 ± 0.24	11.9 ± 0.55	52.8 ± 1.18
STRING Yeast						
1.22 ± 0.11	1.2 ± 0.01	2.96 ± 0.88	10.3 ± 0.71	3.81 ± 0.07	6.35 ± 0.46	13.0 ± 0.6
MINT Yeast						
0.82 ± 0.03	0.81 ± 0.01	0.88 ± 0.01	0.91 ± 0.03	0.91 ± 0.02	0.9 ± 0.02	0.98 ± 0.06
BioGRID Human						
1.25 ± 0.01	1.12 ± 0.04	16.1 ± 0.37	117 ± 4.78	35.5 ± 3.6	27.9 ± 0.89	130 ± 5.00
STRING Human						
0.42 ± 0.01	0.45 ± 0.01	6.72 ± 0.09	25.3 ± 0.5	9.61 ± 0.17	12.0 ± 0.19	31.5 ± 0.71
MINT Human						
2.03 ± 0.14	1.93 ± 0.02	2.12 ± 0.11	2.17 ± 0.02	2.12 ± 0.02	2.13 ± 0.02	2.27 ± 0.04
HuRI Human						
2.87 ± 0.1	2.73 ± 0.03	3.28 ± 0.09	3.6 ± 0.06	3.41 ± 0.04	3.41 ± 0.04	3.74 ± 0.04

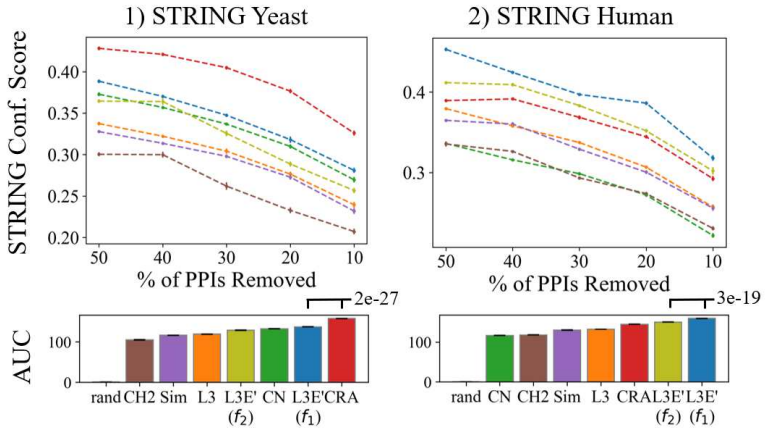
Table 3: Average computation times (in minutes) with standard deviation (denoted by $\pm$) for the experiments in Fig. 5.

7.2 ExactL3 Predictions are Biologically Relevant

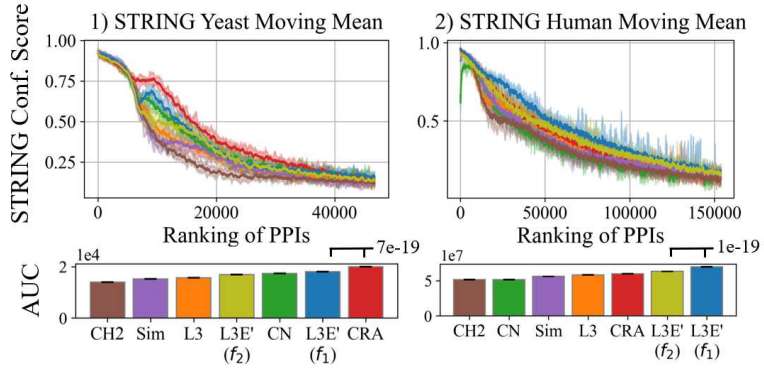
In addition to the statistical significance of L3E' in PPI link prediction, we are also interested in measuring the strength of the biological evidence that supports the predicted PPIs. The first measure that we consider is the STRING confidence score [31]. The STRING confidence score estimates the confidence of a PPI by evaluating evidence for the two proteins such as whether their genes co-express, whether the proteins co-occur phylogenetically, whether the proteins appear together frequently in the literature, and more. We extracted the STRING confidence scores from the STRING datasets, interpreting every null score as a zero. Fig. 7(a) shows the mean STRING confidence scores of the predicted PPIs across different sample sizes of all datasets for every predictor. The random predictor (rand) has been omitted from the figure due to its insignificance. According to the plots, L3E' is the best predictor for the human dataset and the second-best predictor for the yeast dataset (after CRA). This indicates that PPIs predicted by L3E' are biologically relevant.

To investigate whether there is a correlation between the ranking of a predicted PPI and its STRING confidence score, we plotted the moving mean (window size of 100, 10 steps forward in each iteration) of the STRING confidence score along with the ranking of the predicted PPIs in Fig. 7(b) for datasets with 50% of the PPIs removed. The figures for the rest of the sample sizes are included in Figs. S8 and S9. The moving mean shows that for every predictor, the predicted PPIs that are ranked higher indeed have a higher STRING confidence score than those that are ranked lower. The difference between L3E' and the other predictors is that, like in the situation in Fig. 7(a), L3E' is the best predictor for the human dataset where the predicted PPIs in general have higher confidence scores, and the second-best predictor for the yeast dataset.

Next, we computed the Gene Ontology (GO) Semantic Similarity (GOSemSim) scores of the predicted PPIs. The GOSemSim score estimates the similarity of two proteins based on the similarity of their so-called GO annotations that describe proteins in terms of their role as a cellular component, their role in molecular functions, and their role within biological processes. The implementation that we used was a GOSemSim package written in the R programming language [36] based on



(a) Mean STRING confidence score of the top predicted PPIs for link predictors of dataset sample size from 50% to 90%, with 10% of increase each interval.



(b) Moving mean of STRING confidence score along the top predicted PPIs for link predictors of dataset with 50% of the PPIs removed.

Figure 7: (Section 7.2) (a) The mean STRING confidence score across all sample sizes, and (b) the moving means of the STRING confidence scores for sample size 50%. The shaded regions in Fig. 7(b) illustrate the variance (the minimum and maximum values) in STRING confidence scores.

Wang's method [37] with the BMA strategy; null GOSemSim scores are ignored in the computations. Fig. S10 shows the GOSemSim scores of the predicted PPIs of all predictors across the datasets of different sample sizes. The link predictors are separated for comparison according to the principle they are based on (Table 2: CN-based, L3-based, or control). While the differences between predictors are less striking than in the experiments above, we can observe that the CN-based predictors have better GOSemSim scores than the L3-based predictors in general. This is natural because CN characterizes protein pairs with similar functions (Section 2.1). Among the L3-based predictors, we can see that L3E' beats the others with statistical significance (in terms of AUC-values using student's t-test) for four of the seven datasets. Hence, PPIs that are ranked highly by L3E' may possess some functional bias that is encouraged by GOSemSim, e.g., physical PPIs with high L3 scores may reside in neighboring cellular components.

Overlap ratio of predicted PPIs between types of link predictors				
dataset \overlap between	CN-based	L3-based	CN & L3-based	CRA & L3E'(f_1)
BioGRID Yeast	69%	79 ± 10 %	30 ± 6 %	35%
STRING Yeast	89%	92 ± 2 %	72 ± 6 %	74%
MINT Yeast	43%	72 ± 8 %	32 ± 2 %	34%
BioGRID Human	64%	69 ± 4 %	38 ± 4 %	37%
STRING Human	54%	58 ± 4 %	44 ± 3 %	44%
MINT Human	37%	71 ± 10 %	4 ± 2 %	5%
HuRI	64%	79 ± 7 %	24 ± 3 %	23%

Table 4: Overlap ratios of predicted PPIs between different types of link predictors for datasets with 50% of the PPIs removed (Table S4 and S5 show the complete data). 'CN-based' and 'CRA & L3E'(f_1)' denotes the overlap ratio of the predicted PPIs between CN and CRA, and between CRA and L3E'(f_1) respectively. For 'L3-based', since there are multiple L3-based predictors (L3, CH2, Sim, L3E'(f_1), and L3E'(f_2)), we calculated the overlap ratio for each pair of predictors. We then took the mean of these ratios as the final value, and also computed the standard deviation. The same applies to 'CN & L3-based' where a CN predictor is compared to a L3-based predictor. Blue color denotes a relatively higher overlap ratio and red a relatively smaller overlap. Ratios are rounded to nearest integers.

8 Discussion

We have proposed a way to implement the L3 principle in link predictors that we call ExactL3 (L3E). Using the L3E predictors, we are able to deal with hypothetical PPI subgraphs much better than other link predictors (Section 6). L3E can also predict PPIs with strong statistical significance (Section 7.1) and sufficient biological relevance (Section 7.2). In summary, we have demonstrated that the L3E predictors are effective predictors of missing protein-protein interactions that are better than previous methods.

The modeling strength of L3E comes from two main ideas, the realization that the L3 principle can be decomposed into a series of computations that compare graph neighborhoods, and that these comparisons can be computed using similarity metrics. These address what the other L3 predictors are lacking: the original L3 predictor [14] simplifies the L3 principle into counting the number of P_4's, which does not address compatibility of protein interfaces; the CH2 predictors [18] merely adopt the modeling approach of the CRA predictor in L3 subgraphs, which again does not address protein compatibility; and the Sim predictor [26] models protein compatibility using the Jaccard coefficient but only partially since it lets the sets U and V contribute to the final score independently, ignoring the biological motivation of the L3 principle (see also Fig. 1(d)).

The CRA predictor, a CN-based predictor, is one of the best link predictors within our experiments but with a huge variance in its performance. For example, it appears to outperform L3E in Fig. 6(a2) and Fig. 7(a1), but it also underperforms in some cases such as in Fig. 6(a3) and Fig. 6(b3). We hypothesized that this is due to the different paradigms adopted by L3E and CRA in their respective network modeling, so we further investigated the similarity between the pools of PPIs predicted by CRA and L3E'(f_1), i.e., the ratio of the overlap. Surprisingly, as shown in Table 4, these two predictors show a lower overlap ratio compared to the mean overlap ratios of L3-based predictor pairs or CN-based predictor pairs. A lower overlap ratio can also be seen even if we compute overlap ratios of pairs where one predictor is CN-based and another is L3-based predictor. This implies that the PPIs predicted by

L3E are similar to those predicted by other L3 predictors, although L3E is better at ranking them (see Section 7). Furthermore, this suggests that since L3E and CRA predict differing sets of PPIs with competing performance based on different assumptions, the two methods could perhaps be used together in a complementary way to obtain even better link predictions.

Apart from the improved link prediction performance of L3E, these predictors can also be used as a heuristic to narrow down candidate proteins for biological problems. A study from Liu et al. [17] improves protein folding recognition by constructing a protein similarity network based on the L3 principle to identify proteins that could fold in similar ways as the query protein. Since adding network data yields better performance than using protein sequence and profile data only, we believe that L3E could also be used in other similar scenarios.

We anticipate that the use of biological network data will become even more prevalent in various biological problems. Therefore, methods such as L3E may turn out to be useful for many other applications beyond link prediction in protein-protein interaction networks in the future.

Abbreviations

PPIs - protein-protein interactions; CN - common neighbor; L3E - ExactL3; L3 - path of length-3; PR - precision-recall; AUC - area under curve; PR AUC - area under precision-recall curve; conf. score - confidence score; GO - gene ontology; GOSemSim - gene ontology semantic similarity

Declarations

Acknowledgements

The authors would like to acknowledge the Pilot High Performance Computing (HPC) Platform of The Hong Kong Polytechnic University for the free computing resources to run the experiments.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

Availability of data and materials

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/BMC_PPI_L3E

Ethics approval and consent to participate

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Consent for publication

Not applicable.

Authors' contributions

Both authors contributed to the conceptualization, problem formulation, methodology, formal analysis, and writing of the manuscript. H.Y. developed all the software, conducted the experiments, and compiled the results.

Authors' information

Not applicable.

Author details

[1]Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China. [2]Graduate School of Informatics, Kyoto University, 606-8501, Kyoto, Japan.

References

1. Eisenberg, D., Marcotte, E.M., Xenarios, I., Yeates, T.O.: Protein function in the post-genomic era. Nature **405**(6788), 823–826 (2000). doi:10.1038/35015694
2. Cusick, M.E., Klitgord, N., Vidal, M., Hill, D.E.: Interactome: gateway into systems biology. Human Molecular Genetics **14**(suppl_2), 171–181 (2005). doi:10.1093/hmg/ddi335
3. De Las Rivas, J., Fontanillo, C.: Protein–protein interactions essentials: Key concepts to building and analyzing interactome networks. PLOS Computational Biology **6**(6), 1–8 (2010). doi:10.1371/journal.pcbi.1000807
4. De Las Rivas, J., Fontanillo, C.: Protein–protein interaction networks: unraveling the wiring of molecular machines within the cell. Briefings in Functional Genomics **11**(6), 489–496 (2012). doi:10.1093/bfgp/els036. https://academic.oup.com/bfg/article-pdf/11/6/489/650213/els036.pdf

5. Steffen, M., Petti, A., Aach, J., D'haeseleer, P., Church, G.: Automated modelling of signal transduction networks. BMC Bioinformatics **3**(1), 34 (2002). doi:10.1186/1471-2105-3-34
6. Enright, A.J., Van Dongen, S., Ouzounis, C.A.: An efficient algorithm for large-scale detection of protein families. Nucleic Acids Research **30**(7), 1575–1584 (2002). doi:10.1093/nar/30.7.1575. https://academic.oup.com/nar/article-pdf/30/7/1575/7039591/gkf245.pdf
7. Silverman, E.K., Schmidt, H.H.H.W., Anastasiadou, E., Altucci, L., Angelini, M., Badimon, L., Balligand, J.-L., Benincasa, G., Capasso, G., Conte, F., Di Costanzo, A., Farina, L., Fiscon, G., Gatto, L., Gentili, M., Loscalzo, J., Marchese, C., Napoli, C., Paci, P., Petti, M., Quackenbush, J., Tieri, P., Viggiano, D., Vilahur, G., Glass, K., Baumbach, J.: Molecular networks in network medicine: Development and applications. WIREs Systems Biology and Medicine **12**(6), 1489 (2020). doi:10.1002/wsbm.1489. https://onlinelibrary.wiley.com/doi/pdf/10.1002/wsbm.1489
8. Liu, C., Ma, Y., Zhao, J., Nussinov, R., Zhang, Y.-C., Cheng, F., Zhang, Z.-K.: Computational network biology: Data, models, and applications. Physics Reports **846**, 1–66 (2020). doi:10.1016/j.physrep.2019.12.004. Computational network biology: Data, models, and applications
9. Michaut, M., Kerrien, S., Montecchi-Palazzi, L., Chauvat, F., Cassier-Chauvat, C., Aude, J.-C., Legrain, P., Hermjakob, H.: InteroPORC: automated inference of highly conserved protein interaction networks. Bioinformatics **24**(14), 1625–1631 (2008). doi:10.1093/bioinformatics/btn249. https://academic.oup.com/bioinformatics/article-pdf/24/14/1625/16882191/btn249.pdf
10. Pitre, S., Dehne, F., Chan, A., Cheetham, J., Duong, A., Emili, A., Gebbia, M., Greenblatt, J., Jessulat, M., Krogan, N., Luo, X., Golshani, A.: Pipe: a protein-protein interaction prediction engine based on the re-occurring short polypeptide sequences between known interacting protein pairs. BMC Bioinformatics **7**(1), 365 (2006). doi:10.1186/1471-2105-7-365
11. 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 (2003). doi:10.1126/science.1087361. https://science.sciencemag.org/content/302/5644/449.full.pdf
12. Zhang, Q.C., Petrey, D., Deng, L., Qiang, L., Shi, Y., Thu, C.A., Bisikirska, B., Lefebvre, C., Accili, D., Hunter, T., Maniatis, T., Califano, A., Honig, B.: Structure-based prediction of protein–protein interactions on a genome-wide scale. Nature **490**(7421), 556–560 (2012). doi:10.1038/nature11503
13. Lü, L., Zhou, T.: Link prediction in complex networks: A survey. Physica A: Statistical Mechanics and its Applications **390**(6), 1150–1170 (2011). doi:10.1016/j.physa.2010.11.027
14. Kovács, I.A., Luck, K., Spirohn, K., Wang, Y., Pollis, C., Schlabach, S., Bian, W., Kim, D.-K., Kishore, N., Hao, T., Calderwood, M.A., Vidal, M., Barabási, A.-L.: Network-based prediction of protein interactions. Nature Communications **10**(1), 1240 (2019). doi:10.1038/s41467-019-09177-y. A preliminary version of this article appeared in bioRxiv (2018). 10.1101/275529.
15. Granovetter, M.S.: The strength of weak ties. American Journal of Sociology **78**(6), 1360–1380 (1973). doi:10.1086/225469. https://doi.org/10.1086/225469
16. do Valle, I.F., Roweth, H.G., Malloy, M.W., Moco, S., Barron, D., Battinelli, E., Loscalzo, J., Barabási, A.-L.: Network medicine framework shows proximity of polyphenol targets and disease proteins is predictive of the therapeutic effects of polyphenols. bioRxiv (2021). doi:10.1101/2020.08.27.270173. https://www.biorxiv.org/content/early/2021/02/08/2020.08.27.270173.full.pdf
17. Liu, B., Zhu, Y., Yan, K.: Fold-LTR-TCP: protein fold recognition based on triadic closure principle. Briefings in Bioinformatics **21**(6), 2185–2193 (2019). doi:10.1093/bib/bbz139. https://academic.oup.com/bib/article-pdf/21/6/2185/34672124/bbz139.pdf
18. Muscoloni, A., Abdelhamid, I., Cannistraci, C.V.: Local-community network automata modelling based on length-three-paths for prediction of complex network structures in protein interactomes, food webs and more. bioRxiv, $\mathrm{doi}{:}10.1101/346916$ (2018). doi:10.1101/346916. https://www.biorxiv.org/content/early/2018/06/18/346916.full.pdf
19. Yuen, H.Y., Jansson, J.: Better link prediction for protein-protein interaction networks. In: 2020 IEEE 20th International Conference on Bioinformatics and Bioengineering (BIBE), pp. 53–60 (2020). doi:10.1109/BIBE50027.2020.00017
20. Sharan, R., Ulitsky, I., Shamir, R.: Network-based prediction of protein function. Molecular Systems Biology **3**(1), 88 (2007). doi:10.1038/msb4100129. https://www.embopress.org/doi/pdf/10.1038/msb4100129
21. Zhou, T., Lü, L., Zhang, Y.-C.: Predicting missing links via local information. The European Physical Journal B **71**(4), 623–630 (2009). doi:10.1140/epjb/e2009-00335-8
22. Adamic, L.A., Adar, E.: Friends and neighbors on the web. Social Networks **25**(3), 211–230 (2003). doi:10.1016/S0378-8733(03)00009-1
23. Lei, C., Ruan, J.: A novel link prediction algorithm for reconstructing protein–protein interaction networks by topological similarity. Bioinformatics **29**(3), 355–364 (2012). doi:10.1093/bioinformatics/bts688. https://academic.oup.com/bioinformatics/article-pdf/29/3/355/17102727/bts688.pdf
24. Nakajima, N., Hayashida, M., Jansson, J., Maruyama, O., Akutsu, T.: Determining the minimum number of protein-protein interactions required to support known protein complexes. PLOS ONE **13 no.4 article e0195545**(4), 1–17 (2018). doi:10.1371/journal.pone.0195545
25. Cannistraci, C.V., Alanis-Lobato, G., Ravasi, T.: From link-prediction in brain connectomes and protein interactomes to the local-community-paradigm in complex networks. Scientific Reports **3**(1), 1613 (2013). doi:10.1038/srep01613
26. Chen, Y., Wang, W., Liu, J., Feng, J., Gong, X.: Protein interface complementarity and gene duplication improve link prediction of protein-protein interaction network. Frontiers in Genetics **11**, 291 (2020). doi:10.3389/fgene.2020.00291
27. Jaccard, P.: The distribution of the flora in the alpine zone.1. New Phytologist **11**(2), 37–50 (1912). doi:10.1111/j.1469-8137.1912.tb05611.x. https://nph.onlinelibrary.wiley.com/doi/pdf/10.1111/j.1469-8137.1912.tb05611.x

28. Lü, L., Jin, C.-H., Zhou, T.: Similarity index based on local paths for link prediction of complex networks. Phys. Rev. E **80**, 046122 (2009). doi:10.1103/PhysRevE.80.046122
29. Lehne, B., Schlitt, T.: Protein-protein interaction databases: keeping up with growing interactomes. Human Genomics **3**(3), 291 (2009). doi:10.1186/1479-7364-3-3-291
30. Oughtred, R., Stark, C., Breitkreutz, B.-J., Rust, J., Boucher, L., Chang, C.e.l.: The BioGRID interaction database: 2019 update. Nucleic Acids Research **47**(D1), 529–541 (2018). doi:10.1093/nar/gky1079. https://academic.oup.com/nar/article-pdf/47/D1/D529/27436717/gky1079.pdf
31. Szklarczyk, D., *et al.*: STRING v10: protein–protein interaction networks, integrated over the tree of life. Nucleic Acids Research **43**(D1), 447–452 (2014). doi:10.1093/nar/gku1003. https://academic.oup.com/nar/article-pdf/43/D1/D447/7311163/gku1003.pdf
32. Licata, L., Briganti, L., Peluso, D., Perfetto, L., Iannuccelli, M., Galeota, E.e.l.: MINT, the molecular interaction database: 2012 update. Nucleic Acids Research **40**(D1), 857–861 (2011). doi:10.1093/nar/gkr930. https://academic.oup.com/nar/article-pdf/40/D1/D857/16957240/gkr930.pdf
33. Luck, K., *et al.*: A reference map of the human binary protein interactome. Nature **580**(7803), 402–408 (2020). doi:10.1038/s41586-020-2188-x
34. van Rijsbergen, C.J.: Information Retrieval. Butterworth, Oxford, UK (1979)
35. Boyd, K., Eng, K.H., Page, C.D.: Area under the precision-recall curve: Point estimates and confidence intervals. In: Blockeel, H., Kersting, K., Nijssen, S., Železný, F. (eds.) Machine Learning and Knowledge Discovery in Databases, pp. 451–466. Springer, Berlin, Heidelberg (2013)
36. Yu, G., Li, F., Qin, Y., Bo, X., Wu, Y., Wang, S.: GOSemSim: an R package for measuring semantic similarity among GO terms and gene products. Bioinformatics **26**(7), 976–978 (2010). doi:10.1093/bioinformatics/btq064. https://academic.oup.com/bioinformatics/article-pdf/26/7/976/564042/btq064.pdf
37. Wang, J.Z., Du, Z., Payattakool, R., Yu, P.S., Chen, C.-F.: A new method to measure the semantic similarity of GO terms. Bioinformatics **23**(10), 1274–1281 (2007). doi:10.1093/bioinformatics/btm087. https://academic.oup.com/bioinformatics/article-pdf/23/10/1274/497100/btm087.pdf

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- BMCSuppleBetterLinkPredictionforPPI.pdf