Minimum curvilinearity to enhance topological prediction of protein interactions by network embedding

Carlo Vittorio Cannistraci, Gregorio Alanis-Lobato and Timothy Ravasi

Supplementary Information

Table of Contents

INTRODUCTION

This document serves as Supplementary Information for the article *Minimum curvilinearity to enhance topological prediction of protein interactions by network embedding* and is organised as four sections. Subsequent to this Introduction, Section I describes the network datasets we used and provides details on how they were assembled and prepared for our experiments. Section III expands the main article's explanation of what network embedding is, goes deep into the machinery behind Minimum Curvilinear Embedding and its variations, provides more details on the unsupervised and supervised embedding techniques against which we compared our proposed approach, and briefly summarises the means by which we addressed dimension determination to provide candidate interaction scores using network embedding. In Section IV, formulae and descriptions of the node-neighbourhood approaches for link prediction are presented and Section IV concludes this material with detailed information on our performance evaluation framework for link prediction.

I. NETWORK DATASETS

Four different yeast Protein-protein Interaction Networks (PPINs) are the main datasets analysed in this work. Yeast (*S. cerevisiae)* networks are the preferred benchmark to test algorithms for topological link-prediction because of the large amount of information available for yeast, in terms of both detected interactions and Gene Ontology (GO) associations (You et al., 2010).

One of the networks (Ben-Hur and Noble, 2005) comprises 10,411 physical interactions between 4036 proteins, presents a STRING overlap of 84% (i.e. 84% of this network's edges are reported in the STRING database), and is available at: http://noble.gs.washington.edu/proj/sppi/. The second network (J. Chen, Hsu, M.-L. Lee, et al., 2006) is composed of 12,234 interactions (mixture of physical, literature curated and functional) between 4385 proteins and presents a STRING overlap of 87%. The final two networks were those used by (You et al., 2010) corresponding to 12,934 physical interactions between 3645 proteins (considered sparse by You and colleagues) with a STRING overlap of 65% and 29,922 physical interactions between 3883 proteins (considered dense by You and colleagues) with STRING overlap of 67%. These last two networks are available at: http://home.ustc.edu.cn/~yzh33108/Manifold.htm. Details on the overall 4 networks are provided in the Table S1.

The embedding techniques used in this work rely on the shortest-path (SP) metric to generate a distance matrix that is later embedded into a space of reduced dimensions. In order to achieve the aforementioned, networks over which the embedding techniques are applied need to be connected. Otherwise, SPs between disconnected components would be, by definition, infinity and the embedding process could not be performed.

Due to this issue, we ensured all networks used in this work were connected. The original format of these datasets was a list of edges that was transformed into an adjacency matrix after assigning a numerical ID to each unique interactor that corresponds to adjacency matrix row and column indices. Matlab's function `graphconncomp` finds the connected components of a graph and returns a list indicating which component each node belongs to. To find the largest connected component, the mode of this list is computed and all nodes that are not part of the component indicated by the mode are discarded from the network. Note that this process shrinks the adjacency matrix and the list of interactors, which requires ID reassignments to match the new matrix dimensions.

Table S1 lists all networks used in this article along with their original number of nodes and edges and the reduced number of nodes and edges after extraction of the largest connected component.

Network		**Original dataset**		**Largest connected component**	
Organism	**Reference**	**Nodes**	**Edges**	**Nodes**	**Edges**
S. cerevisiae	Ben-Hur and Noble, 2005	4233	10517	4036	10411
S. cerevisiae	Chen et al., 2006	4385	12234	4385	12234
S. cerevisiae	You et al., 2010 Sparse	3645	12934	3645	12934
S. cerevisiae	You et al., 2010 Dense	3883	29922	3883	29922

Table S1. Network datasets used in this work. The table lists the number of nodes and edges of the original datasets and how these change when the largest connected component is extracted. Organism and references for each PPIN are also listed.

II. NETWORK EMBEDDING

One means to better visualise and interpret high-dimensional data is to assume that it lies on a manifold embedded in a space of high dimensions. In particular, given n high-dimensional data points $\{x_1, x_2, \dots, x_n\}$ with $x_i \in \mathbb{R}^D$ lying on a manifold embedded in a D-dimensional space, the objective of Manifold Embedding is to find a mapping $\mathcal{M}: \mathbb{R}^D \to \mathbb{R}^d$ such that the mapped points $\{y_1, y_2, \dots, y_n\}$ with $y_i \in \mathbb{R}^d$ preserve some of the topological properties of the original manifold.

If it is assumed that PPINs lie on a high, unknown metric space shaped by the biological properties of the proteins that form them, then it is possible to map a network to a reduced space, in which proteins that are close to each other are more likely to interact (Boguñá et al., 2008). This problem is called Network Embedding and is very similar to Manifold Embedding, however in this case we lack the coordinates of the points that we want to take to a low dimensional space (further details below).

Let us represent a PPIN by an undirected, unweighted graph $G = (V, E)$ with a set of $|V|$ nodes and a set of $|E|$ edges, which is a set of 2-element subsets of V. Network Embedding consists of finding a mapping $\mathcal{M}: V \to X$, where X is a set of points $\{x_1, x_2, \dots, x_{|V|}\}$ with $x_i \in \mathbb{R}^d$, i.e. each node of G is assigned a coordinate in a space of *d* dimensions such that the original topological properties of the network are preserved in this low dimensional space. Manifold Embedding algorithms can be easily modified for Network Embedding, however not all the algorithms that learn manifolds are applicable for this task, only those able to embed a topology starting from a distance or adjacency matrix can be used. The reason is that in this particular work, our goal is to perform the above described Network Embedding with no information other than the network topology itself. In this case, nodes have no properties other than their connections to other nodes and because of this, the embedding process necessarily has to start with the only information available: node distances based on their connectivity in the network, i.e. graph SPs.

Embedding techniques that start from proximity matrices P defined in the high-dimensional space (with entries $P_{i,j}$ storing the distance between the pair of points i,j), seek to find low-dimensional points such that their all pairwise distances in the reduced space are equal or very close to those defined in P. If p is the proximity matrix in the reduced space (with entries $p_{i,j}$ storing the distance between the pair of points i,j), embedding can be seen as the following minimisation problem:

$$\min \sum_{i<j} \left(P_{i,j} - p_{i,j}\right)^2$$

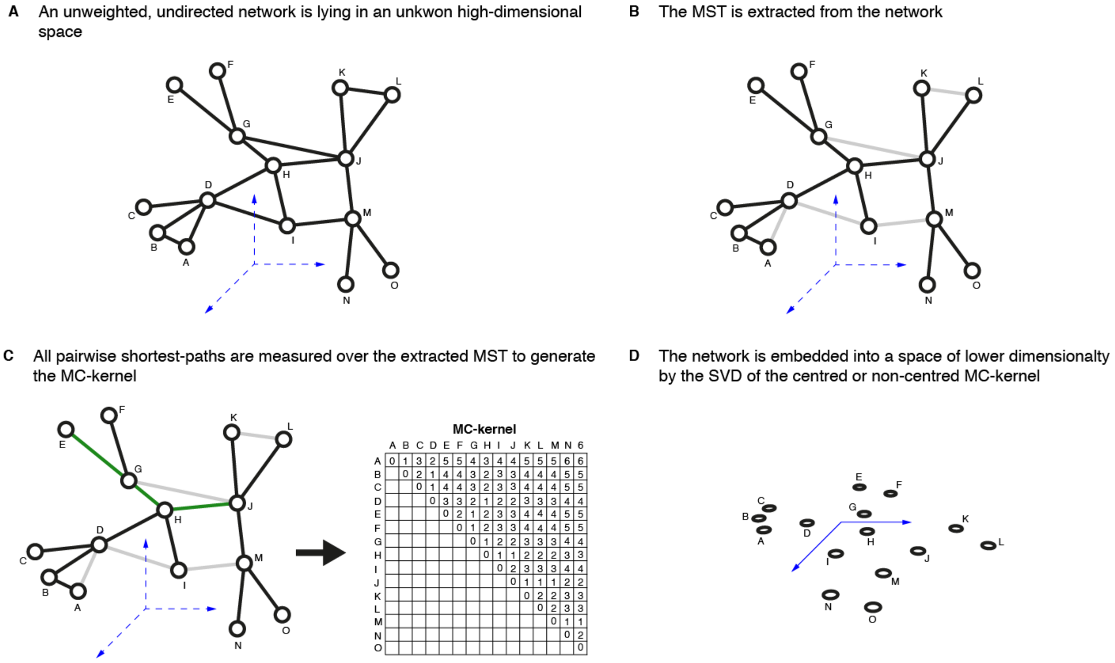
There are different strategies to solve the above problem (see following Sections II.1-II.3) and it is important to note that link prediction is now possible because, having coordinates for all network nodes in a space of low dimensionality, allows for the assignment of scores to pairs of nodes that are not connected in the original network topology. These scores are associated with distances in the reduced space and the list of non-adjacent pairs of nodes sorted by this measure of link likelihood is the output of the link prediction process by Network Embedding (further details in the following section).

II.1 Minimum curvilinear embedding

Minimum Curvilinear Embedding (MCE) is a parameter-free and time efficient unsupervised algorithm for nonlinear dimensionality reduction. It was originally introduced as a new form of nonlinear Multidimensional Scaling (MDS) in 2010 (Cannistraci et al., 2010) and has proven to be a powerful and robust tool in different applications (Zagar et al., 2011; Ryu et al., 2012).

MCE consists of a first step in which a nonlinear distance matrix (the minimum curvilinear distance matrix, MC-matrix) is calculated as pair-wise sample distances over the Minimum Spanning Tree (MST) in the feature space. The MST is computed according to the Euclidean distance or the Correlation distance (or any other preferred distance) in this space. The embedding transformation is then performed by classical MDS of the MC-matrix.

In the case of Network Embedding, we lack coordinates or properties of the samples (nodes) in the feature space (high-dimensional space where the network lies). Given a network (PPIN in the case of this work) lying in a space of high dimensions (see Fig. S1A), MCE extracts the MST directly from this available network (see Fig. S1B) and generates the MC-kernel by computing pairwise distances over the MST alone (see Fig. S1C). The embedding process by MDS would be the next step but in this work we propose a more versatile and efficient mapping procedure: embedding by Singular Value Decomposition (SVD) of the centred or non-centred MC-kernel (see Fig. S1D). We refer to the former possibility simply as MCE (because in fact it is theoretically equivalent to embedding by MDS) and to the latter as non-centred MCE (ncMCE). We refer the reader to the main article where pseudocode for MCE is listed in Algorithm 1 and a link to its Matlab implementation is provided.



A An unweighted, undirected network is lying in an unkwon high-dimensional space

B The MST is extracted from the network

C All pairwise shortest-paths are measured over the extracted MST to generate the MC-kernel

D The network is embedded into a space of lower dimensionalty by the SVD of the centred or non-centred MC-kernel

MC-kernel

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	6
A	0	1	3	2	5	5	4	3	4	4	5	5	5	6	6
B		0	2	1	4	4	3	2	3	3	4	4	4	5	5
C			0	1	4	4	3	2	3	3	4	4	4	5	5
D				0	3	3	2	1	2	2	3	3	3	4	4
E					0	2	1	2	3	3	4	4	4	5	5
F						0	1	2	3	3	4	4	4	5	5
G							0	1	2	2	3	3	3	4	4
H								0	1	1	2	2	2	3	3
I									0	2	3	3	3	4	4
J										0	1	1	1	2	2
K											0	2	2	3	3
L												0	2	3	3
M													0	1	1
N														0	2
O															0

Figure S1. MCE/ncMCE for Network Embedding. A. An unweighted, undirected network lies on a space of high dimensions (the dashed axes indicate that in principle the dimensions of this space are unknown). **B.** The MST is extracted from the network (edges in grey are those not considered by the MST). **C.** Pairwise distances are computed over the extracted MST to generate the MC-kernel (since it is symmetric, we show only the upper triangle). Note, for example, that the geodesic distance between nodes E and J is of 2 hops when measured over the graph but increases to 3 hops when measured over the MST (green path in the figure). **D.** The network is embedded into a reduced space by SVD of the centred (MCE) or non-centred (ncMCE) MC-kernel. Now the coordinates of the network nodes are known (now the axes are solid lines) and operations such as link prediction are possible.

The hypothesis behind link prediction by Network Embedding is that once the network is mapped to a low dimensional space, nodes that are close to each other are more likely to interact (Boguñá et al., 2008). Kuchaiev, You and their colleagues exploited this idea and assigned likelihood scores to candidate PPI by means of Euclidean distances (EDs) between nodes in the reduced space (see Fig. S2A).

In this paper we propose a different scoring scheme. Instead of simply computing EDs between nodes, we reconstruct the original network topology in the low dimensional space and, since the network is now weighted with the distances between directly connected nodes (see edge thickness in Fig. S2B), we can now compute “cleaner” shortest-paths over this network reconstruction to assign scores to candidate PPIs (see Fig. S2B).

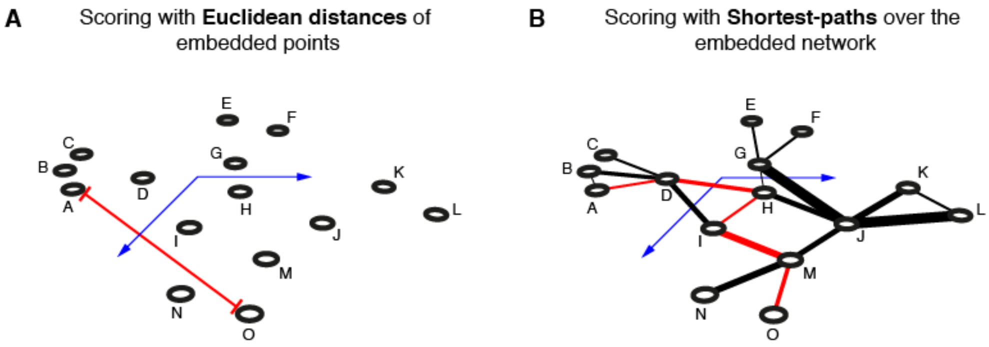

A
Scoring with **Euclidean distances** of embedded points
B
Scoring with **Shortest-paths** over the embedded network
A
B
C
D
E
F
G
H
I
J
K
L
M
N
O

Figure S2. Scoring approaches for link prediction by Network Embedding. A. Scores for candidate links by computing Euclidean distances in the reduced space. **B.** Our proposed scoring scheme requires the reconstruction of the original network in the low dimensional space. This network is now weighted (see edge thickness in the figure) and "cleaner" shortest-paths are the scores for the candidate interactions. Note how the score for candidate interaction A-O is different in the two schemes (red lines in the figure).

II.1.1 Testing the proposed innovations

In order to verify if the hypotheses posed above and the scoring scheme proposed in this work hold, we carried out a series of GO-free experiments.

In the first one, we generated 1000 random geometric graphs and counted the fraction of unique False Positives (FPs) visited when computing all pair SPs over the entire network (first step of Isomap algorithm) and over the MST (first step of MCE algorithm) out of the total number of FPs in the entire network. Fig. S3 shows that computing distances over the MST (i.e. using MC) takes into account a small amount of FPs, thus offering a denoised estimate of the network connectivity.

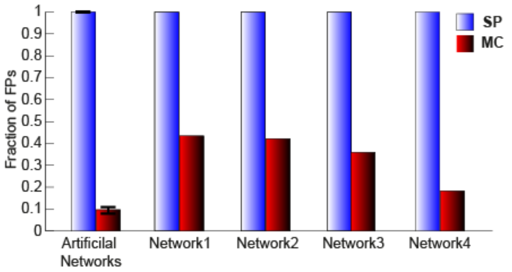


Fig. S3. Fraction of false positive interactions (FPs) used by shortest-path (SP) and minimum curvilinearity (MC). For the artificial networks (RGGs), mean and standard error values on 1000 iterations are provided.

In the second one, we study the power of non-centred MCE in solving the crowding problem in the complex Radar Signal dataset (all details and results appear in the main text).

In the third setup, we followed a methodology similar to that proposed in previous studies (Kuchaiev et al., 2009; You et al., 2010): we fit a nonparametric estimate to the distribution of low-dimensional distances between connected nodes from the network $p(distance|original)$ and another one to the distribution of distances between non-adjacent nodes $p(distance|candidate)$. If the hypothesis that nodes closer to each other in the reduced space are more likely to interact is true, $p(distance|original)$ should have higher peakedness (kurtosis) than

$p(distance|candidate)$ and the distribution itself should be shifted towards 0. Moreover, links from the original network topology whose distance is far from the origin are likely to represent false positives, while non-adjacent nodes whose distance is close to 0 are good candidates for interaction. On the other hand, if this experiment is carried out for EDs and SPs, the latter should provide better discrimination between good and bad candidate PPIs (more separation between distributions) than the former. In Fig. S4 we show that this is indeed the case and that there is a statistically significant difference between $p(distance|original)$ and $p(distance|candidate)$ over different dimensions (as measured by p-values obtained by a Mann-Whitney nonparametric test). This difference is much more significant when the scoring used is SP than when it is ED, reinforcing the idea that our proposed scoring technique is more powerful.

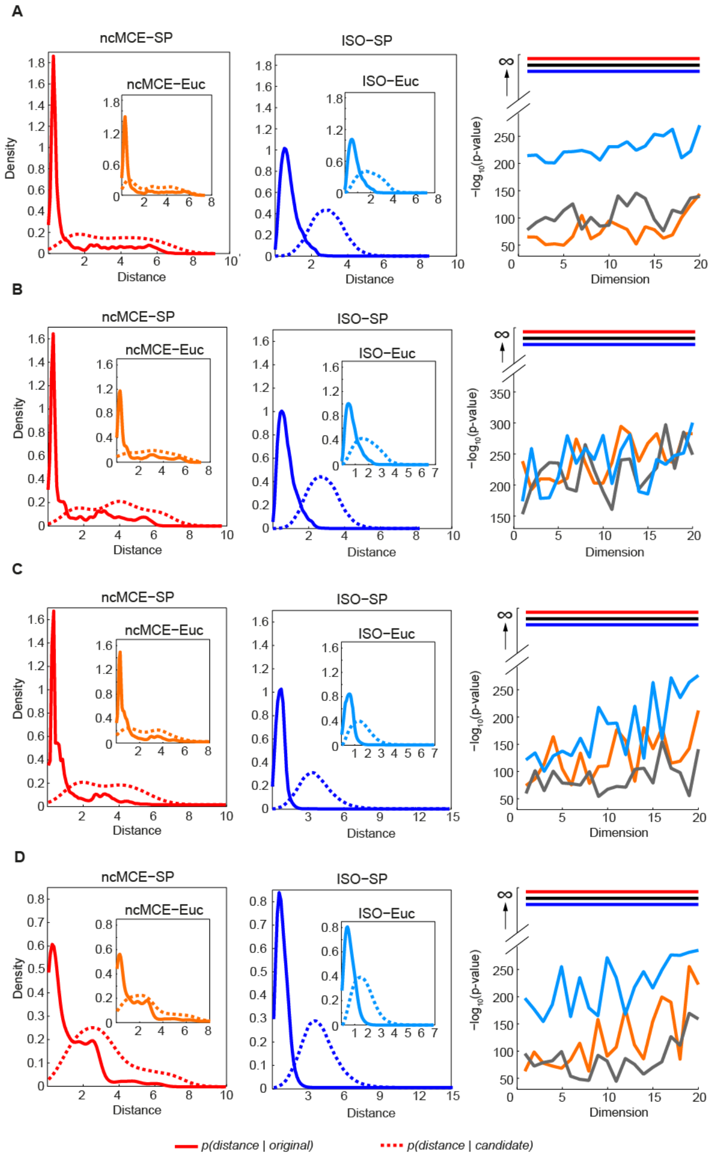


Figure S4. Discrimination between original network and candidate PPIs. Distribution of shortest-path scores in the reduced space (dimension 3 displayed) for the four datasets analysed in this work (**A** Ben-Hur and Noble, 2005 **B** Chen *et al.* 2006 **C** You *et al.* 2010 Sparse and **D** You *et al.* 2010 Dense). Network links $p(distance|original)$ (solid line) and candidate links $p(distance|candidate)$ (dashed line) after ncMCE (left) and Isomap (middle) network embedding. The insets show the distribution of Euclidean distance scores. The rightmost panel shows the resulting p-values from the statistical test for difference in the distributions over different dimensions.

In yet another setup, we make use of random geometric graphs (RGGs). RGGs are important because there is indication that they can be good models for networks such as PPINs (Przulj et al., 2004). We generate RGGs by accommodating 1000 points uniformly at random in the 100-dimensional unitary cube and then connect them if and only if the dot product (similarity) between the vectors with tails in the origin and heads over these points is above a connectivity threshold *r*. We set such threshold by making sure that properties common to real biological networks (small-world and scale-free topologies) and connectivity were present. The advantage of using RGGs to test our innovations is that the sets of true and spurious interactions are clearly defined: true interactions are those that fulfil the connectivity threshold and spurious links are those that do not. Based on this, we added noise to 1000 different networks in amounts typical of PPINs: 40% False Negatives (FNs) and 60% FPs and performed a sparsification experiment in which the link predictors should rediscover the removed true interactions available in the generated RGG. In Fig. 4A-C in the main text, we show how the variations of MCE (especially ncMCE SP) are the strongest approaches in this framework when the networks are embedded into dimension 1-10.

Finally, we assessed the performance of the link predictors on a sparsification experiment over 1000 different RGGs without noise. Fig. 4D-F in the main text, shows how ISOMAP variations perform better in this framework (networks embedded into dimension 1-10), indicating that the use of MCE is encouraged on noisy networks such as PPINs.

It is important to mention that the low precision values obtained in these last two experiments are very common in link prediction due to the large amount of candidate interactions compared to the small amount of pruned interactions a technique is trying to rediscover (see for example (Liben-Nowell and Kleinberg, 2007), in which precision in the range of 0.0015-0.0048 are reported when predicting links in coauthorship networks).

We generate RGGs with 1000 nodes and around 12600 edges (we say *around* because RGGs are random and the number of edges is not fixed), which means that the number of candidate interactions is 1000(1000 – 1)/2 – 12600 = 486900. At the last sparsification level (right before the network loses connectivity and when the largest amount of pruned links should be rediscovered), the number of pruned links is ~11000, this means that the probability that a random prediction is correct is 11000/(486900 + 11000) = 0.02. The precision of, for example, ncMCE is 0.06 at this level for noisy RGGs (see Fig. 4A), which is a three-fold improvement over the analytical precision of a random predictor.

II.1.2 MCE's time performance

We claim ncMCE is a time efficient algorithm and here, we detail its computational complexity. Using the graph theory notation introduced above, we can say the MST extraction from the network of study has time complexity $\mathcal{O}(|E|\log|V|)$ (Kruskal, 1956). The pairwise shortest-path computation to generate the MC-kernel by Johnson's algorithm has time complexity $\mathcal{O}(|V|^2\log|V| + |V||E|)$ (Johnson, 1977). Finally, the economy size SVD performed to embed the network has time complexity $\mathcal{O}(d^2|V|)$ where d is the dimension of choice for the mapping. As it is clear, the step that consumes more time is the MC-kernel computation, thus the overall time complexity is in the order of the square of the number of nodes (this is also thanks to the non-centring approach because centred MCE is an $\mathcal{O}(|V|^3)$ algorithm). This is efficient when compared to, for example Tree Preserving Embedding (see the Radar Signal dataset experiment in the main article) or Isomap which are $\mathcal{O}(|V|^3)$ algorithms (due to greedy approximations and MDS respectively). Since parallel and highly optimised versions for MST extraction and SVD computation are available, efficient and fast version of MCE can be taken to production for the embedding of very large networks such as the World Wide Web. Time performance of MCE against other prediction techniques analysed in this work are listed in Table S2, these results confirm the above mentioned theoretical time complexities. The reported times correspond to the whole process of candidate interaction scoring (including dimension determination by AUC criterion, network embedding, and score designation). The computations were carried out in a Dell Precision T7500 workstation with Intel® Xeon® CPU X5650 @ 2.67GHz x 12 cores and 47.2 GB of RAM.

	Time in seconds				
Network Technique	Ben-Hur and Noble, 2005	Chen, et al., 2006	You et al., 2010 Sparse	You et al., 2010 Dense	Time robustness (max. time over all networks)
ncMCE SP	56.53	69.06	46.92	58.31	69.06
ncMCE Euc	56.94	230.94	54.20	110.93	230.94
MCE SP	168.30	69.72	105.96	147.89	168.30
MCE Euc	180.67	130.17	79.79	149.17	180.67
ISO SP	158.32	206.52	119.88	121.69	206.52
ISO Euc	130.78	155.42	175.77	233.74	233.74
FSW	529.04	627.04	443.00	509.95	627.04
CDD	365.29	433.48	303.04	349.96	433.48
IG1	109.08	129.51	90.16	112.82	129.51

Table S2. Time performance in seconds for the main link prediction techniques studied in this work. The time measured corresponds to the entire link prediction process: from dimension determination by AUC criterion and embedding (if applicable) to candidate PPI scoring. The last column reports Time Robustness, i.e. the maximum time spent by a technique to output the scored list of candidates links. Parallel implementations of the node-neighbourhood based predictors were used for this analysis.

II.1.3 Effect of the existence of multiple MSTs on MCE's performance

To conclude this section on MCE, we present the result of an important simulation regarding its intrinsic mechanism. Since the networks studied in this work are unweighted and undirected, more than one MST can be extracted from them. We studied the performance of ncMCE in the four main network datasets analysed in this article when different MSTs are extracted from the network at different levels of network sparsification (for more details on performance evaluation see the main article and Section IV of this document). As we can see in Fig. S5, the difference in ncMCE's performance when using different MST is so small (see standard error bars and performance curve practically overlapping) that it can be neglected. This is further proof that the MST is a powerful part of MCE's mechanism and that it is able to mine the most important information from the network topology.

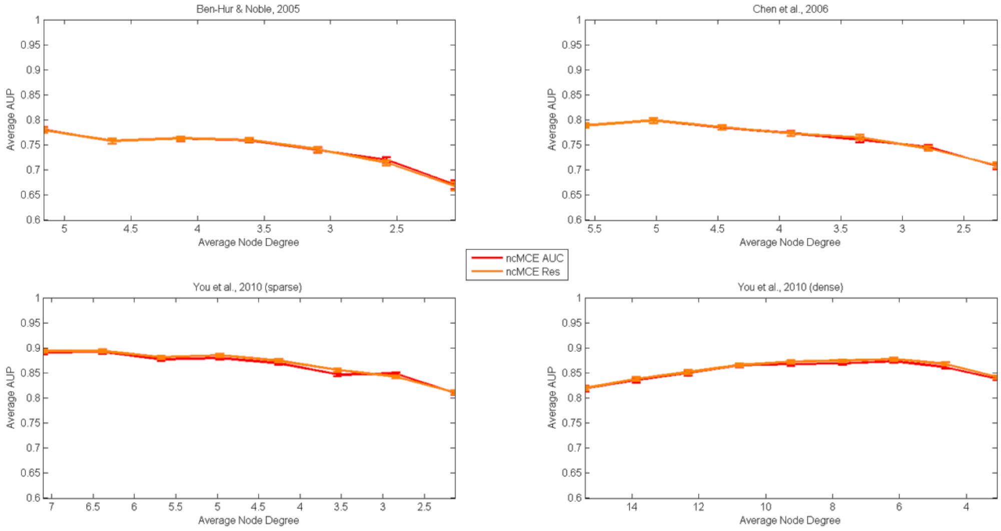


Figure S5. ncMCE's sparsification curves. At each percentage of link deletions, only one random sparsified network configuration is generated, but 100 different MSTs are extracted (by random initialization) from this configuration. The performance attained by the different ncMCEs (each of which uses a different MST) are averaged and their standard error is included as an error bar in the sparsification curve.

II.2 Unsupervised embedding

The unsupervised embedding techniques used in this work are Isomap, Sammon Mapping, Stochastic Neighbourhood Embedding (SNE), tSNE, MCE and its non-centred form named ncMCE (see Section II.1).

Isomap (Tenenbaum et al., 2000) is an embedding method that, after the construction of a neighbourhood graph over the points in the high dimensional space via the first *k* Euclidean-distance neighbours, approximates geodesic distances with shortest-path lengths over this graph. Later, it uses the generated distance matrix as an input for Multidimensional Scaling (MDS) that is the algorithm in charge of the embedding to lower dimensions. In (You et al., 2010), Isomap is used to embed a network to a space of low dimensions by skipping the construction of the neighbourhood graph and using the PPIN instead, they later used FSW (see Section III) to score candidate PPIs. In this work, we use the exact same methodology but without the use of FSW because in our experiments we did not find big differences between the two approaches (see Fig. S6).

SNE (Hinton and Roweis, 2002) and tSNE (Van der Maaten and Hinton, 2008) are considered force-based embedding techniques because they map points that are very similar close to each other by using attractive forces and points that are different far from each other by using repulsive forces (Shieh et al., 2011). Both SNE and tSNE transform Euclidean distances between the points in the high dimensional space into conditional probabilities that a certain point will pick another as a neighbour. Since we lack the high-dimensional coordinates of the proteins in the network, we compute the distances between them as shortest-paths over the PPIN and use this distance matrix as input for these embedding techniques. In order to transform these distances into a probability matrix, both SNE and tSNE use a parameter called 'perplexity'. SNE and tSNE are robust to changes on this parameter and its typical values range from 5 to 50 (Van der Maaten and Hinton, 2008). In fact, the authors propose a default perplexity value of 30 in the existing implementations of the techniques, which is the fixed value that we used in our experiments.

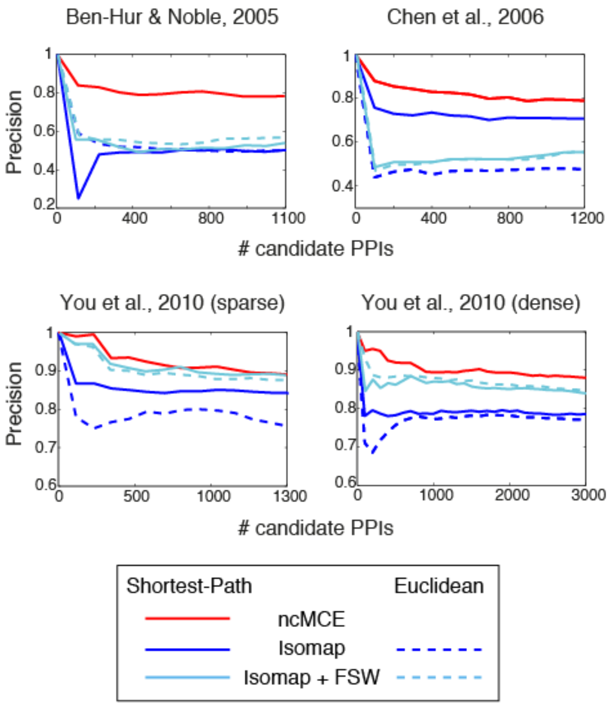


Figure S6. ncMCE vs Isomap and Isomap+FSW. Precision curves that highlight the difference in performance between Isomap and Isomap+FSW in the four network datasets used in this work. The X-axis indicates how many interactions are taken from the top of the candidate interaction list and the Y-axis indicates the precision of the technique for that portion of protein pairs.

Sammon Mapping (Sammon, 1969) (SM) is a type of nonlinear MDS that preserves small distances between data points in the reduced space better than classical MDS. It accepts as input the distance matrix for the points in the high-dimensional space and a random or PCA-based initialization of the mapping to the reduced space. It refines the low dimensional coordinates of the points using a steepest descent procedure to search for the minimum error defined as the difference between distances in the high and low dimensional spaces. As we did with SNE and tSNE, the distance matrix input to SM was that defined by the shortest-path lengths between proteins over the PPIN.

II.3 Supervised embedding

We tested two supervised embedding techniques: local MDS and Neighbour Retrieval Visualizer (NeRV). These methods can be considered force-based as well but instead of using forces based on kernels like SNE or tSNE do (Hinton and Roweis, 2002; van der Maaten and Hinton, 2008), they use forces based on neighbourhood graphs (Shieh et al., 2011). These two techniques require a parameter called 'effective number of neighbours' that resembles the perplexity of SNE and tSNE (Venna et al., 2010), however we did not work with it because the PPIN itself corresponds to the neighbourhood graph.

Both local MDS (Venna and Kaski, 2006) and NeRV (Venna et al., 2010) try to find a good trade-off between trustworthiness (points that are far from each other in the high dimensional space should not be part of the same neighbourhood) and continuity (points that are very close to each other should be part of the same neighbourhood). The user tells the algorithms what is more important between trustworthiness and continuity tuning a parameter λ. In this work, we applied these two techniques supervising the performance obtained with values of λ from 0 to 1 with a step of 0.1 and took the low dimensional coordinates that yielded the best prediction result.

II.4 Dimension determination for embedding

Previous work on PPIN embedding for interaction reliability assessing and link prediction showed that, in general, the embedding dimension does not affect the prediction or reliability assessing process (Kuchaiev et al., 2009; You et al., 2010). However, we wanted to propose a method for automatic determination of a good dimension of embedding, i.e. the one that allowed for the best technique's performance. Aside from making life easier for the user of embedding techniques by removing one parameter from the process, this automatic determination would put MCE/ncMCE to the test, because the best dimension was also used for all the embedding approaches analysed in this work. The dimension determination strategies we propose are described below.

II.4.1 The AUC criterion

The AUC-criterion (AUC after Area Under the Curve) is designed to work in combination with any algorithm for network embedding adopted for link prediction: it automatically determines the dimension into which we should embed the network. For a certain dimension of embedding, the prediction procedure (Fig. 1 in the main article) assigns a likelihood score to each interaction (low scores correspond to interactions that are likely to occur and high scores to interactions that are not). The scores are computed for both the original interactions in the network (O) and the candidate interactions (C), which are all those protein pairs that were not linked in the input network. The scored O and C generate two distributions of distances (see Fig. S7).

From this point we refer to the works of Kuchaiev et al. (Kuchaiev et al., 2009) and You et al. (You et al., 2010) with the acronym KY. As suggested in KY, we can vary a cut-off ε, from 0 up to the maximum distance of the two distributions so that all protein pairs with scores below ε are considered positives and all protein pairs with scores above ε are considered negatives (Fig. S7). KY suggest that taking the PPIs from the original network as our positive set, we can compute the number of True Positives (TP), FNs, FPs and True Negatives (TN) at each ε cut. This will yield a pair (1-Specificity, Sensitivity) that, measured for the entire ε range, generates a Receiver Operating Characteristic curve (ROC) and an Area Under the ROC Curve (AUC) that characterises the performance for the current dimension (Fig. S7). However, KY noticed that we have to handle the concept of positives and negatives in the PPI prediction framework very carefully because what we call "original network" and consider as our positive set contains some false interactions. Since we are interested in C that are likely to be real (hypothetically not yet detected by experimental methods), at each ε we consider a subset of the best ranked FPs, which - according to KY - are more likely real undetected interactions, and we call them Advocated Candidates (AC). On the other hand, protein pairs that are part of the original network but did not pass the ε cut, i.e. the worst ranked FNs according to KY, are considered Rejected original interactions (R). Given this conceptual framework proposed: the false positive rate (1-Specificity), that is originally FP/Negatives, is now AC/C; and the true positive rate (Sensitivity), that is originally TP/Positives, is now (O-R)/O (Fig. S7). We can now exploit the maximum AUC as criterion for optimization (AUC-criterion). You and colleagues showed that the AUC for different dimensions is very similar and the increase in its value tends to vanish for higher dimensions, thus they considered a fixed dimension 10 for embedding in their experiments (You et al., 2010). We take advantage of this finding (Fig. S7) by computing the AUC from dimension 1 up to the dimension where the difference between its AUC and the one of the previous dimension is less than 10^{-3}. In several tests we found that 10^{-3} is a such small difference between AUCs that we can consider it not significant, thus the last AUC is considered the appropriate to identify the dimensions for embedding. We then take the scored candidate interactions given by this dimension for the final evaluation of the method used.

II.4.2 The Res-criterion

One of the motivations to propose a second criterion is that the AUC-criterion considers the original network as a sort of gold-standard when in reality it includes several false interactions (You et al., 2010).

The new criterion for dimension determination is based on the idea that, the more different the likelihood score values are, the better they discriminate good candidates from bad candidates in the ranking. Thus, we have to define a measure of the resolution of the score values provided by each dimension, so that: the higher this measure, the higher the resolution; and the more we should consider this dimension as correct for embedding. The measure we used for dimension determination is defined in Equation S1.

$$Resol_{All} = \frac{\sigma(unique(scores))}{Dim} \qquad \text{(S1)}$$

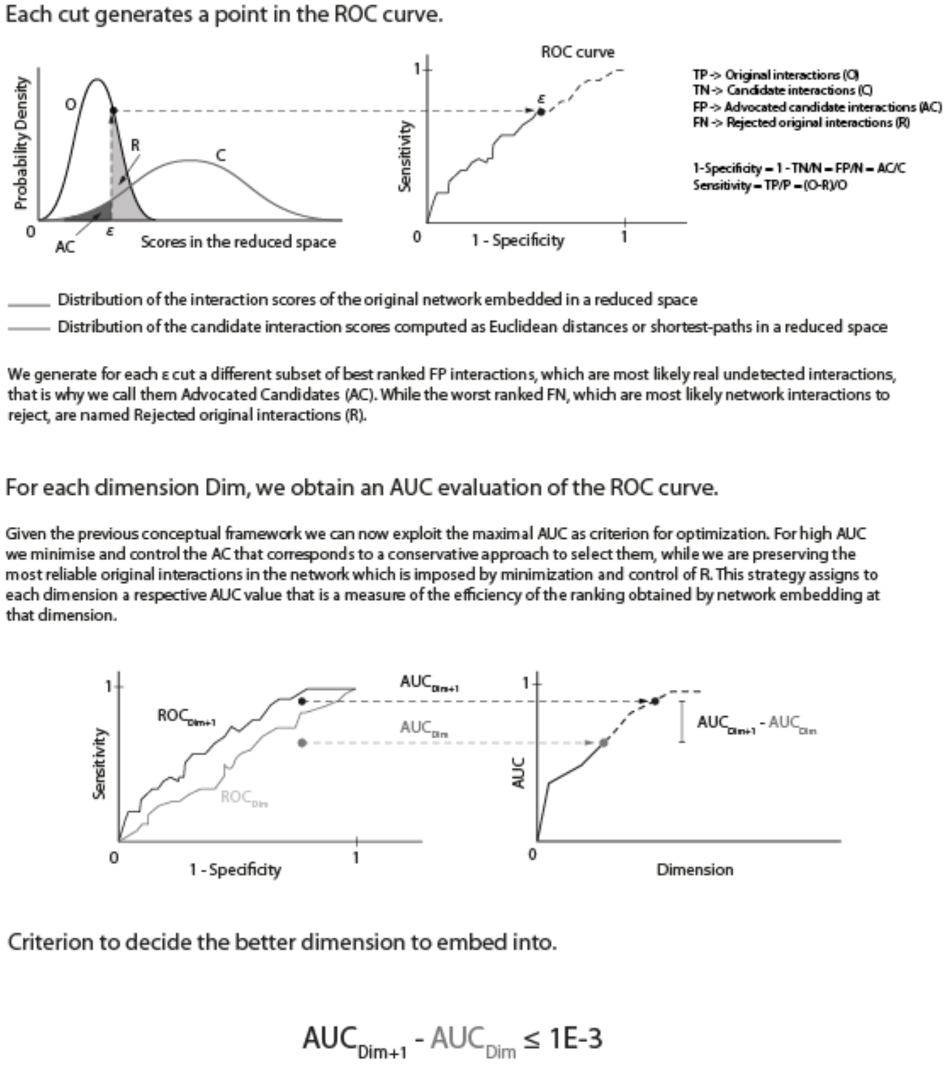
This formula takes all the unique score values of the candidate interactions in dimension *Dim*, computes its standard deviation *σ*, and divides it by *Dim*. By taking the unique score values we have an idea of the resolution that *Dim* is providing, then we determine the quality of that resolution by computing *σ* which quantify the variation between the unique score values. Finally the division by *Dim* penalises the higher dimensions, which were shown not to provide any relevant increase in performance (You et al., 2010). We specifically designed this criterion to fit with the quality of MCE, which provides more *soft-threshold effect* in the lowest dimensions. This explains why we tested the Res-criterion only in combination with MCE.

We also applied a variation of Equation S1 to check if dimension determination using only the top 100 ranked interactions would generate better area under the precision curve (AUP) in the range between 0 and the first 100 candidate PPIs only (for performance evaluation details see the main article and Section IV of this document).

The difference is that we compute σ on the unique scores from the top 100 candidate protein pairs (see Equation S2).

$$Resol_{100} = \frac{\sigma(unique(scores_{1\ to\ 100}))}{Dim} \tag{S2}$$

Results of the experiments showing performance differences when the three above described criteria are compared, are depicted in Fig. S8.



Each cut generates a point in the ROC curve.

Probability Density
O
R
C
0
AC
ε
Scores in the reduced space
ROC curve
1
Sensitivity
ε
0
1 - Specificity
1
TP -> Original interactions (O)
TN -> Candidate interactions (C)
FP -> Advocated candidate interactions (AC)
FN -> Rejected original interactions (R)
1-Specificity = 1 - TN/N = FP/N = AC/C
Sensitivity = TP/P = (O-R)/O

Distribution of the interaction scores of the original network embedded in a reduced space

Distribution of the candidate interaction scores computed as Euclidean distances or shortest-paths in a reduced space

We generate for each ε cut a different subset of best ranked FP interactions, which are most likely real undetected interactions, that is why we call them Advocated Candidates (AC). While the worst ranked FN, which are most likely network interactions to reject, are named Rejected original interactions (R).

For each dimension Dim, we obtain an AUC evaluation of the ROC curve.

Given the previous conceptual framework we can now exploit the maximal AUC as criterion for optimization. For high AUC we minimise and control the AC that corresponds to a conservative approach to select them, while we are preserving the most reliable original interactions in the network which is imposed by minimization and control of R. This strategy assigns to each dimension a respective AUC value that is a measure of the efficiency of the ranking obtained by network embedding at that dimension.

Criterion to decide the better dimension to embed into.

$$AUC_{Dim+1} - AUC_{Dim} \leq 1E\text{-}3$$

Figure S7. Explanation of how the AUC-criterion determines the correct dimension for embedding. First of all, the list of scores for the interactions is considered as two separate distributions of distances in the reduced space: the distribution of interaction scores of the original network referred to as O (which, in the reduced space, are always the Euclidean distances between the proteins because they are connected directly by one edge) and the distribution of candidate interaction scores referred to as C (which can be Euclidean distances as in You et al., 2010 or shortest-path lengths, our alternative). We cut these distributions from 0 up to the maximum available score. At each step, we consider everything to the left of ε as positive and everything to the right as negative and count the number of TPs, FPs, TNs, and FNs to generate a pair (1-Specificity, Sensitivity) in the ROC curve. However, given the nature of the Prediction Problem in PPINs, where the original network (considered as the positive set) includes also false interactions, we pose a new conceptual framework where the FPs that pass ε are considered Advocated Candidates (AC), and FNs that do not pass it are considered Rejected original interactions (R). Thus, for each dimension we obtain a value for the area under the ROC curve and when the difference between the AUC of dimension Dim + 1 and dimension Dim is less than 10^{-3}, we take the current dimension as the optimal to embed the network into.

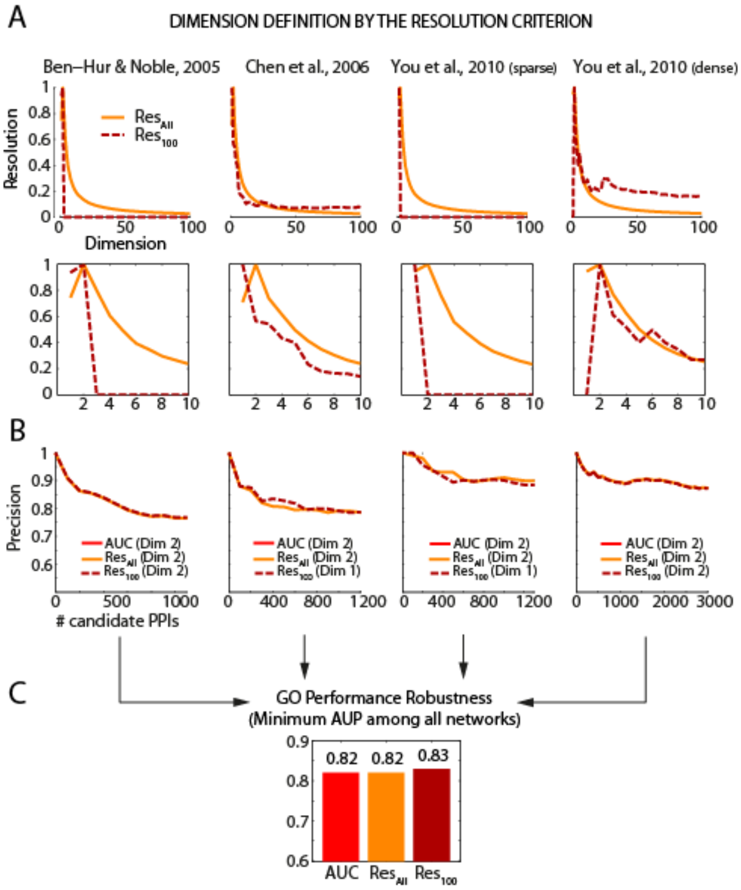


Figure S8. AUC-criterion and resolution-criterion. A. Dimension determination curves for ncMCE-SP over all the candidate interaction scores ($\mathrm{Res}_{\mathrm{All}}$) and the top 100 ranked ones (Res_{100}). The X-axis indicates the different dimensions tested and the Y-axis the measure of resolution for a specific dimension. The upper panel shows the curves for up to 100 dimensions, and the lower panel is the zoomed-in portion of the plot for dimensions 1 through 10. **B.** Precision curves that show the performance achieved by the dimensions determined using the AUC-criterion (AUC), $\mathrm{Res}_{\mathrm{All}}$ and Res_{100}. **c.** Performance Robustness (minimum AUP among all networks) for the different criteria combined with ncMCE-SP.

III. NODE-NEIGHBOURHOOD PREDICTORS

When prior functional biological knowledge about the proteins that form a PPIN is not available, the only biological resource we are left with is the information allocated in the PPIN topology itself. Several techniques have been proposed to exploit the topology of a PPIN in order to assess the reliability of the network interactions or to predict protein function.

The pioneers of PPI reliability assessment are Saito and his team. In 2002, they proposed the so-called Interaction Generality (IG1) Index which, given that partners of ‘sticky’ proteins and self-activators do not interact with anything else in the network (J. Chen, Chua, et al., 2006), assigns high index values to potential false positives (interactions whose seed proteins x and y have a lot of neighbours that do not interact with anything else) and low values to more reliable PPIs (see Equation S3, where *G* is a network, and *x*, *y*, *x*', and *y*' are nodes in *G*).

$$IG1(x,y) = 1 + |\{\{x',y'\} \in G \,|\, \boldsymbol{x'} \in \{\boldsymbol{x'},\boldsymbol{y'}\}, \boldsymbol{y'} \notin \{\boldsymbol{x},\boldsymbol{y}\}, \boldsymbol{\Gamma}(\mathbf{y'}) = \mathbf{1}\}| \tag{S3}$$

Later, further indices were proposed, such as Interaction Generality Two (IG2) (Saito et al., 2003) or Interaction Reliability by Alternative Path (IRAP) (J. Chen et al., 2005), but their minimal comparative performance makes them very computationally expensive (J. Chen et al., 2005). On the other hand, indices to predict functions such as the Czekanowski-Dice Dissimilarity (CDD) (Brun et al., 2003) and Functional Similarity Weight (FSW) (Chua et al., 2006) were additionally and successfully employed for reliability assessment (J. Chen, Chua, et al., 2006) and link prediction in PPINs (You et al., 2010). Equations S4 and S5 respectively represent the formulae of these last two indices . In these equations $\gamma(x)$ is the set of neighbours of *x* including itself, and n_{avg} is the average node degree of the network.

$$CDD(x,y) = \frac{|\gamma(x)\Delta\gamma(y)|}{|\gamma(x) \cup \gamma(y)| + |\gamma(x) \cap \gamma(y)|} \tag{S4}$$

$$FSW(x,y) = \frac{2|\gamma(x) \cap \gamma(y)|}{|\gamma(x) - \gamma(y)| + 2|\gamma(x) \cap \gamma(y)| + \lambda_{x,y}} \cdot \frac{2|\gamma(x) \cap \gamma(y)|}{|\gamma(y) - \gamma(x)| + 2|\gamma(x) \cap \gamma(y)| + \lambda_{y,x}} \tag{S5}$$

where $\lambda_{x,y} = \max(0, n_{avg} - (|\gamma(x) - \gamma(y)| + |\gamma(x) \cap \gamma(y)|))$.

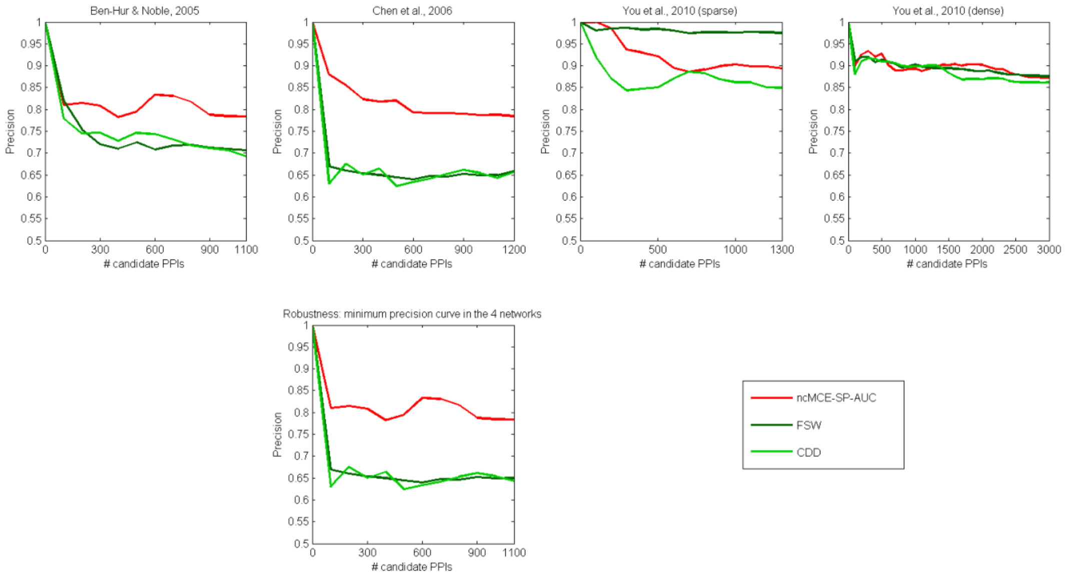


Figure S9. Performance comparison between ncMCE-SP-AUC, FSW and CDD, using the classical approach based on precision curves. The X-axis indicates how many interactions are taken from the top of the candidate interaction list (sorted decreasingly by score), and the Y-axis indicates the precision of the technique for that portion of protein pairs.

Given that ncMCE ended up being the best embedding approach to link prediction, we compared it against the above described node-neighbourhood based prediction techniques. In Fig. S9 we show the performance of ncMCE (using the AUC-criterion and shortest-paths to score) against CDD and FSW (see main article and Section IV of this document for details on performance evaluation). On the other hand, Fig. S10 shows the results of the in-silico validation of the PPIs predicted by the best variation of MCE when compared to those predicted by FSW. The main results of this analysis are performed for the top 100 candidate PPIs, however, in Fig. S10E we also include results from different thresholds of top candidates (20, 40, 60, 80 and the top 100).

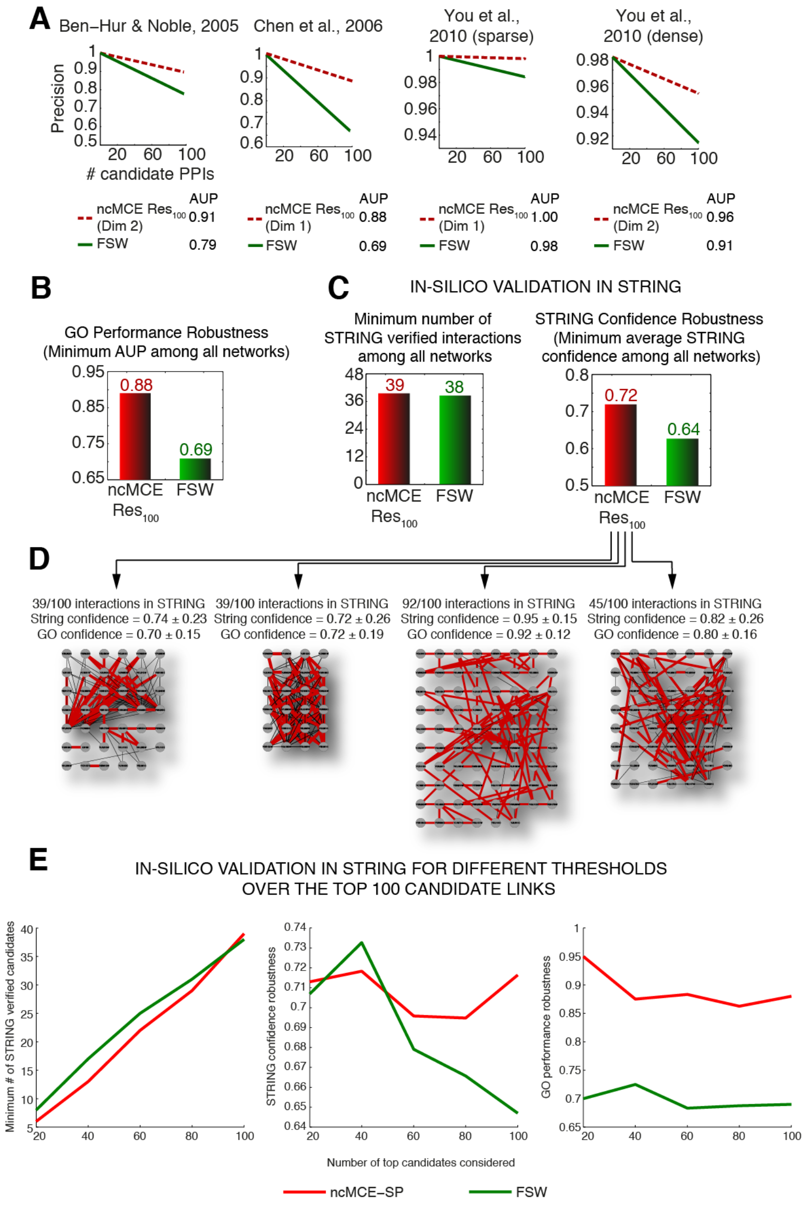


Figure S10. In-silico STRING validation. A. GO precision curves for the top 100 candidate PPIs proposed by ncMCE-SP combined with Res_{100} (referred to as ncMCE-Res_{100} in the figure) and FSW. The X-axis indicates how many interactions are taken from the top of the candidate interaction list (sorted decreasingly by score), and the Y-axis indicates the precision of the technique for that portion of protein pairs. **B.** GO performance robustness for the above methods. **C.** In-Silico validation of the top 100 candidate PPIs proposed by the above methods. **D.** Sub-networks formed by the top 100 candidate PPIs proposed for each network by ncMCE-Res_{100}. The red edges indicate links validated in STRING database. The number of validated PPIs, their average STRING confidence along with standard deviation, and their average GO confidence along with standard deviation, appear on top of each network. **E.** Validation in STRING for different thresholds over the top 100 candidates.

IV. PERFORMANCE EVALUATION FRAMEWORK

IV.1 Problems in the evaluation of link predictors

The evaluation of the topological link-prediction in PPIN presents several pitfalls. Adding random interactions to the PPINs in order to generate false links is theoretically ill-posed. Creation of links uniformly at random between nodes generates Erdős–Rényi network models (Erdős and Rényi, 1960; Watts and Strogatz, 1998; Barabási and Albert, 1999), and this is in contrast with the nature of PPINs that instead follow a scale-free topological organisation, typical of the Barabási-Albert model (Barabási and Albert, 1999). The absence of complete databases of true-negative PPIs for the most important species is another reason for the absence of negative sets of interactions for the evaluation. This is a general issue that afflicts link-prediction also in other contexts such as social networks (Liben-Nowell and Kleinberg, 2007). Thus, simulations oriented to assess the precision (ratio of true-positive-predicted links to all positive links) of predictors are preferred (Liben-Nowell and Kleinberg, 2007; You et al., 2010). The standard strategy is to randomly remove interactions from the network and to assess the precision of the methods in identifying these missing links. The number of links removed from the network can be considered a parameter to vary in the simulation to test the performance of the predictors at different levels of network sparsification. However, this strategy is well-posed only in the case of networks, such as social, that are not relevantly noisy - in the sense that they present few false-positive interactions – and that present a sufficient number of links to remove. PPINs instead are very noisy because they present high false-positive rate (Hart et al., 2006) - ~50% in the case of high-throughput yeast two-hybrid assays (You et al., 2010) - and incomplete because they present high false-negative rate (Hart et al., 2006; You et al., 2010). A consequence is that the application of such removal-strategy on a PPIN is not recommended and might be misleading. In fact, the interactions removed from the network are used as the true-positive set for the precision assessment, but in the case of PPINs, a significant amount of them can represent spurious information. A second issue with the removal-strategy is that it does not really assess the prediction of missing and unknown interactions (here called candidate-interactions), but technically it only evaluates the method's performance to recover the original network topology, which paradoxically contains several false-positives. This can cause precision overestimation, in particular for those predictors based on greedy strategies to explore the network topology (such as the shortest path).

The use of all the GO terms in this precision evaluation (molecular function or MF, biological process or BP and cellular compartment or CC) has been motivated by the guilt-by-association principle, which states that interacting proteins are likely to share function or be located in the same cellular compartment (Oliver, 2000): ~63% of interacting proteins have at least one common function and ~76% of them share a cellular compartment in yeast (Saito et al., 2002, 2003) and there are several studies that have employed the GO to reduce the amount of false positive PPIs resulting from computational predictions (Mahdavi and Y.-H. Lin, 2007; Zeng et al., 2008) and to detect functional interaction patterns (Turanalp and Can, 2008). In particular, Zeng and colleagues showed that GO semantic similarity is a very strong tool to discriminate between true and spurious PPIs. In their experiments they are able to detect 99.61% FPs and 83.03% TPs by means of the GO only. Yet another study showed that an important amount of human embryonic stem cell PPI share GO terms

(Zuo et al., 2009), which is another confirmation of the validity of our evaluation framework. Generally, past studies on evaluation of interactions adopted all the three GO terms to assess precision (Chen, et al., 2006; Chen, et al., 2006; Chen, et al., 2005; Saito, et al., 2002; Saito, et al., 2003; You, et al., 2010). Other studies (Qi et al., 2006) came to the conclusion that the three GO terms are very important features that allow for the accurate prediction of PPIs (MF is the second most important, preceded by BP and followed by CC). However, from a different standpoint, it might be considered arguable to include the GO molecular function category when evaluating interactions. In fact, one might observe that: while it makes sense that two proteins working in the same biological process (or being in the same cellular compartment) interact, expecting that two proteins with the same molecular function (an example two enzymes) interact is biologically weak-posed. For this reason we repeated the precision assessments presented in Fig. 5, excluding molecular function from the evaluation. The curves displayed in Fig. S11 offer an evaluation similar to the one presented in Fig. 5 of the main article, from which we can gather that the good performances of ncMCE/MCE approaches are fairly robust, since they are confirmed also under a reasonable change of the evaluation framework.

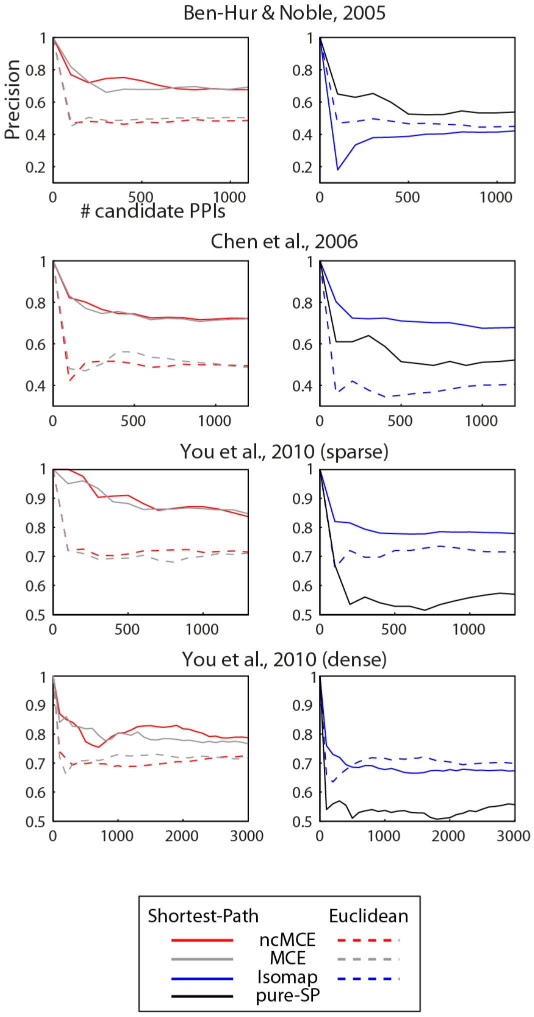


Figure S11. Comparison between ncMCE, MCE, Isomap, and SP without MF ontology for evaluation. The X-axis indicates how many interactions are taken from the top of the candidate interaction list, and the Y-axis indicates the precision of the technique for that portion of protein pairs.

IV.2 Gene Ontology evaluation

PPIs that are involved in the same BP, have similar MF or are located in the same CC are very likely to occur (Saito et al., 2002, 2003; J. Chen et al., 2005; J. Chen, Chua, et al., 2006; J. Chen, Hsu, M. L. Lee, et al., 2006; You et al., 2010). We annotate and measure the similarity between GO terms for all proteins in the PPINs using the R package GOSemSim by (G. Yu et al., 2010) and the Wang GO Semantic Similarity method (J. Wang et al., 2007). The GOSemSim function that we used takes as input the list of proteins that form the PPIN, annotates them, computes the Wang GO semantic similarity between proteins and outputs a matrix whose entries are the GO similarities for every PPI. There are several GO semantic similarity indices (Jiang and Conrath, 1997; D. Lin, 1998; Resnik, 1999) that were originally developed for natural language taxonomies and it is not known if they are 100% suitable for GO. Wang's measure was created from the ground up especially for the GO and its values (with a range between 0 if there is not information in the GO for one or both proteins or if they do not have a similar MF, BP or CC and 1 if the proteins share one or more identical GO terms) are more consistent with the human perspective and the manual gene clustering into GO terms (J. Wang et al., 2007). Whenever the Wang similarity is in the high end of the range, the proteins being analysed can be considered analogous in their MF, BP, or CC (J. Wang et al., 2007). Thus, as suggested in previous studies (J. Chen et al., 2005; J. Chen, Chua, et al., 2006; You et al., 2010) we decided to consider only those pairs with Wang similarity above 0.5. Finally, to obtain the precision curve that measures the performance of the indices, we take 10% of the number of interactions in the original network from the candidate list (as done in (You et al., 2010)) and check, by considering from one PPI up to the 10% taken, the proportion of pairs with relevant MF, BP or CC in the different GO matrices which is a measure of Precision (i.e. for each candidate PPI, the maximum of its GO Wang similarities in the three GO categories is computed and if it is greater or equal to 0.5, the link is considered as TP). To summarise the performance of the technique in one number, we report the Area Under the generated Precision curve (AUP). As an additional test of the ability of prediction techniques to push poor candidate PPIs to the bottom of the ranking list, we performed the above described experiment but considering 100% of the original network links instead of 10% only. The results show the power and robustness of our proposed techniques and also point out that, in general, topological link prediction is able to push the best candidate interactions to the top of the ranking list (note the low precision values of the bottom candidates, represented by the right tails of the curves shown in Fig. S12).

IV.3 Evaluation discarding large complexes

Given the fact that large protein complexes tend to be the most numerous in available PPI datasets, the performance of prediction techniques can be biased if the predictor mostly recognizes interactions between proteins taking part in these complexes (Reyes, 2009). To study the behaviour and robustness of our proposed approaches when proteins that are part of large complexes are discarded from the analysis, we annotated all interactors from each of our four main network datasets by means of DAVID Bioinformatics and detected all proteins that are part of large complexes such as the ribosome, the proteasome, and the exosome. We then removed these proteins from our networks and repeated the experiments shown in Figs. 5, S11, and S12 (see Figure S13 below). The results confirm that our proposed approach outperforms in general the others even in this new scenario. Nevertheless, since the largest complexes were removed from the networks, the crowding problem in the embedding space is significantly reduced and, in this particular condition: 1) the use of ncMCE does not give a clear advantage in respect to MCE; 2) the performance of Isomap-SP is enhanced in comparison with its performance in the other experiments.

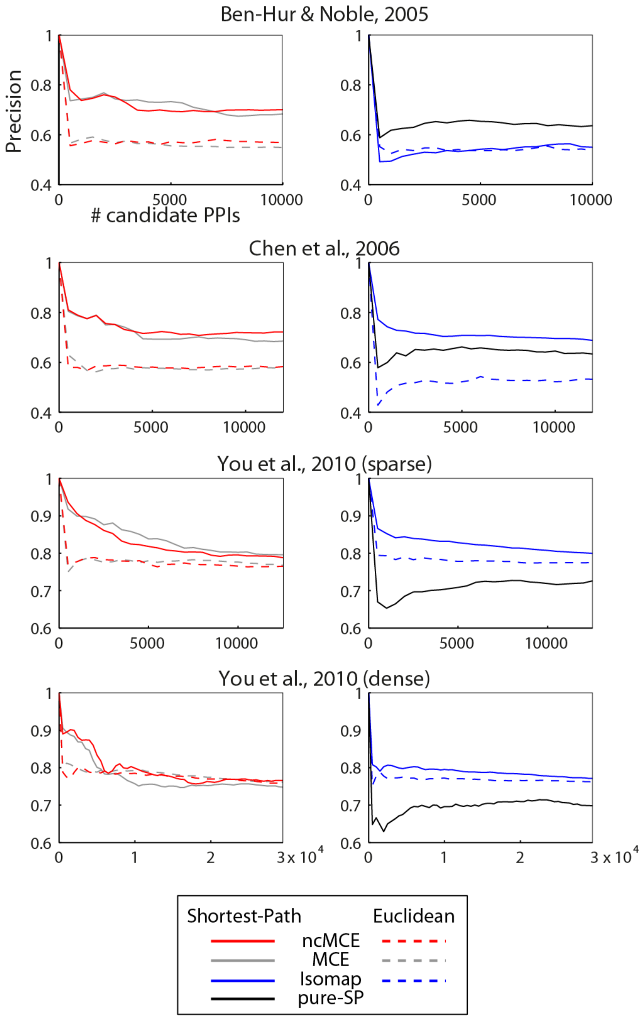


Figure S12. Comparison between ncMCE, MCE, Isomap, and SP for 100% of the original links in the network. The X-axis indicates how many interactions are taken from the top of the candidate interaction list, and the Y-axis indicates the precision of the technique for that portion of protein pairs.

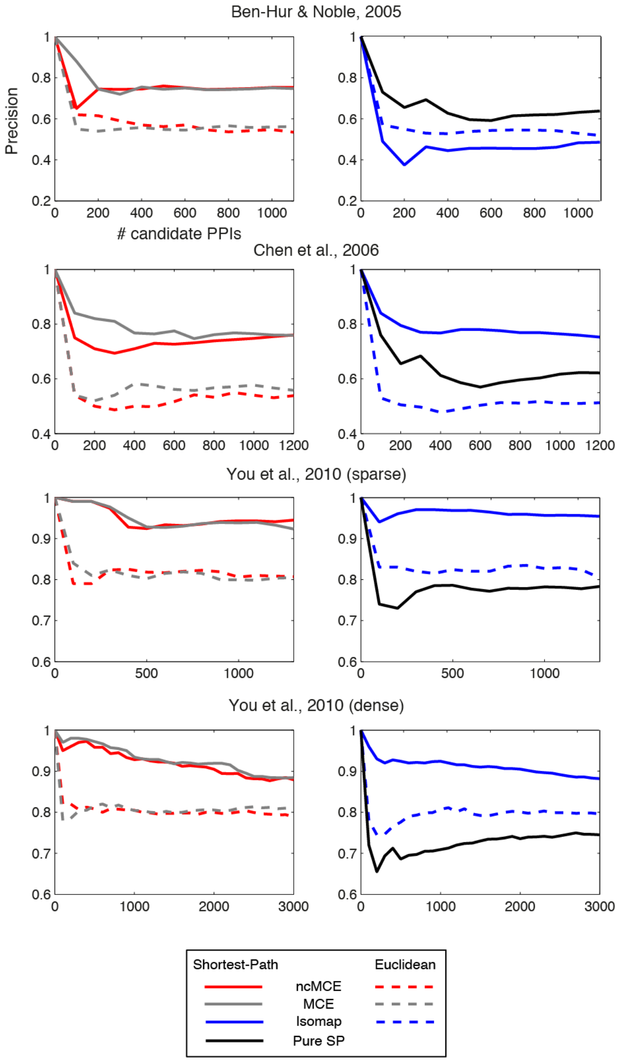


Figure S13. Comparison between ncMCE, MCE, Isomap, and SP when proteins that are part of large complexes are discarded from the analysis. The X-axis indicates how many interactions are taken from the top of the candidate interaction list, and the Y-axis indicates the precision of the technique for that portion of protein pairs.

REFERENCES

Barabási,A.-L. and Albert,R. (1999) Emergence of Scaling in Random Networks. *Science*, **286**, 509–512.

Ben-Hur,A. and Noble,W.S. (2005) Kernel methods for predicting protein-protein interactions. *Bioinformatics*, **21 Suppl 1**, i38–46.

Boguñá,M. et al. (2008) Navigability of complex networks. *Nature Physics*, **5**, 74–80.

Brun,C. et al. (2003) Functional classification of proteins for the prediction of cellular function from a protein-protein interaction network. *Genome Biology*, **5**, R6.

Cannistraci,C.V. et al. (2010) Nonlinear dimension reduction and clustering by Minimum Curvilinearity unfold neuropathic pain and tissue embryological classes. *Bioinformatics*, **26**, i531–9.

Chen,J. et al. (2005) Discovering reliable protein interactions from high-throughput experimental data using network topology. *Artificial intelligence in medicine*, **35**, 37–47.

Chen,J., Hsu,W., Lee,M.L., et al. (2006) Increasing confidence of protein interactomes using network topological metrics. *Bioinformatics*, **22**, 1998–2004.

Chen,J., Chua,H.N., et al. (2006) Increasing confidence of protein-protein interactomes. *Genome Informatics. International Conference on Genome Informatics*, **17**, 284–97.

Chen,J., Hsu,W., Lee,M.-L., et al. (2006) NeMoFinder: Dissecting genome-wide protein-protein interactions with meso-scale network motifs. *KDD '06 Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, 106–115.

Chua,H.N. et al. (2006) Exploiting indirect neighbours and topological weight to predict protein function from protein-protein interactions. *Bioinformatics*, **22**, 1623–30.

Erdős,P. and Rényi,A. (1960) On the evolution of random graphs. *Publications of the Mathematical Institute of the Hungarian Academy of Sciences*, **5**, 17–61.

Hart,G.T. et al. (2006) How complete are current yeast and human protein-interaction networks? *Genome Biology*, **7**, 120.

Hinton,G. and Roweis,S. (2002) Stochastic Neighbor Embedding. *Advances in Neural Information Processing Systems*, **15**, 833–840.

Jiang,J.J. and Conrath,D.W. (1997) Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. *Proceedings of 10th International Conference on Research In Computational Linguistics*.

Johnson,D.B. (1977) Efficient Algorithms for Shortest Paths in Sparse Networks. *Journal of the ACM*, **24**, 1–13.

Kruskal,J.B. (1956) On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society*, **7**, 48–50.

Kuchaiev,O. et al. (2009) Geometric de-noising of protein-protein interaction networks. *PLoS Computational Biology*, **5**, e1000454.

Liben-Nowell,D. and Kleinberg,J. (2007) The Link-Prediction Problem for Social Networks. *Journal of the American Society for Information Science*, **58**, 1019–1031.

Lin,D. (1998) An Information-Theoretic Definition of Similarity. *Proceedings of the 15th International Conference on Machine Learning*, 296–304.

Liu,G. et al. (2009) Complex discovery from weighted PPI networks. *Bioinformatics*, **25**, 1891–7.

Van der Maaten,L. and Hinton,G. (2008) Visualizing Data using t-SNE. *Journal of Machine Learning Research*, **9**, 2579–2605.

Mahdavi,M. a and Lin,Y.-H. (2007) False positive reduction in protein-protein interaction predictions using gene ontology annotations. *BMC Bioinformatics*, **8**, 262.

Oliver,S. (2000) Guilt-by-association goes global. *Nature*, **403**, 601–3.

Przulj,Natasah et al. (2004) Modeling interactome: scale-free or geometric? *Bioinformatics*, **20**, 3508–15.

Qi,Y. et al. (2006) Evaluation of Different Biological Data and Computational Classification Methods for Use in Protein Interaction Prediction. *PROTEINS: Structure, Function, and Bioinformatics*, **500**, 490–500.

Resnik,P. (1999) Semantic Similarity in a Taxonomy : An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language. *Journal of Artificial Intelligence Research*, **11**, 95–130.

Reyes,J.A. (2009) Machine Learning for the Prediction of Protein-Protein Interactions. 2–202.

Ryu,T. et al. (2012) The evolution of ultraconserved elements with different phylogenetic origins. *BMC evolutionary biology*, **12**, 236.

Saito,R. et al. (2003) Construction of reliable protein-protein interaction networks with a new interaction generality measure. *Bioinformatics*, **19**, 756–763.

Saito,R. et al. (2002) Interaction generality, a measurement to assess the reliability of a protein-protein interaction. *Nucleic Acids Research*, **30**, 1163–8.

Sammon,J.W. (1969) Sammon Mapping.pdf. *IEEE Transaction on Computation*, **C-18**, 401–409.

Shieh,A.D. et al. (2011) Tree preserving embedding. *Proceedings of the National Academy of Sciences of the United States of America*, **108**, 16916–21.

Tenenbaum,J.B. et al. (2000) A global geometric framework for nonlinear dimensionality reduction. *Science*, **290**, 2319–23.

Turanalp,M.E. and Can,T. (2008) Discovering functional interaction patterns in protein-protein interaction networks. *BMC Bioinformatics*, **9**, 276.

Venna,J. et al. (2010) Information Retrieval Perspective to Nonlinear Dimensionality Reduction for Data Visualization. *Journal of Machine Learning Research*, **11**, 451–490.

Venna,J. and Kaski,S. (2006) Local multidimensional scaling. *Neural Networks*, **19**, 889–899.

Wang,J. et al. (2007) A new method to measure the semantic similarity of GO terms. *Bioinformatics*, **23**, 1274–81.

Watts,D.J. and Strogatz,S.H. (1998) Collective dynamics of “small-world” networks. *Nature*, **393**, 440–2.

You,Z.-H. et al. (2010) Using manifold embedding for assessing and predicting protein interactions from high-throughput experimental data. *Bioinformatics*, **26**, 2744–51.

Yu,G. et al. (2010) GOSemSim: an R package for measuring semantic similarity among GO terms and gene products. *Bioinformatics*, **26**, 976–8.

Zagar,L. et al. (2011) Stage prediction of embryonic stem cell differentiation from genome-wide expression data. *Bioinformatics*, **27**, 2546–53.

Zeng,E. et al. (2008) Estimating support for protein-protein interaction data with applications to function prediction. *Proceedings of Comput Syst Bioinformatics Conference*, 73–84.

Zuo,C. et al. (2009) Enriching protein-protein and functional interaction networks in human embryonic stem cells. *International Journal of Molecular Medicine*, **23**, 811–819.