An integrative approach to modeling biological networks

Vesna Memišević[1], Tijana Milenković[1], and Nataša Pržulj[2,*]

[1]Department of Computer Science, University of California, Irvine, CA 92697-3435, USA
[2]Department of Computing, Imperial College London, London, SW7 2AZ, UK
[*]Corresponding author (e-mail: natasha@imperial.ac.uk)

Summary

Networks are used to model real-world phenomena in various domains, including systems biology. Since proteins carry out biological processes by interacting with other proteins, it is expected that cellular functions are reflected in the structure of protein-protein interaction (PPI) networks. Similarly, the topology of residue interaction graphs (RIGs) that model proteins' 3-dimensional structure might provide insights into protein folding, stability, and function. An important step towards understanding these networks is finding an adequate network model, since models can be exploited algorithmically as well as used for predicting missing data. Evaluating the fit of a model network to the data is a formidable challenge, since network comparisons are computationally infeasible and thus have to rely on heuristics, or "network properties." We show that it is difficult to assess the reliability of the fit of a model using any network property alone. Thus, we present an integrative approach that feeds a variety of network properties into five machine learning classifiers to predict the best-fitting network model for PPI networks and RIGs. We confirm that geometric random graphs (GEO) are the best-fitting model for RIGs. Since GEO networks model spatial relationships between objects and are thus expected to replicate well the underlying structure of spatially packed residues in a protein, the good fit of GEO to RIGs validates our approach. Additionally, we apply our approach to PPI networks and confirm that the structure of merged data sets containing both binary and co-complex data that are of high coverage and confidence is also consistent with the structure of GEO, while the structure of less complete and lower confidence data is not. Since PPI data are noisy, we test the robustness of the five classifiers to noise and show that their robustness levels differ. We demonstrate that none of the classifiers predicts noisy scale-free (SF) networks as GEO, whereas noisy GEOs can be classified as SF. Thus, it is unlikely that our approach would predict a real-world network as GEO if it had a noisy SF structure. However, it could classify the data as SF if it had a noisy GEO structure. Therefore, the structure of the PPI networks is the most consistent with the structure of a noisy GEO.

1 Introduction

1.1 Background

Large-scale biological network data are increasingly becoming available due to advances in experimental biology. We analyze protein-protein interaction (PPI) networks, where proteins are modeled as network nodes and interactions amongst them as network edges. Since it is the proteins that carry out almost all biological processes and they do so by interacting with other proteins, analyzing PPI network structure could lead to new knowledge about complex

biological mechanisms and disease. Additionally, we analyze network representations of protein structures, "residue interaction graphs" (RIGs), where residues are modeled as network nodes and inter-residue interactions as network edges; an inter-residue interaction exists between residues that are in close spatial proximity. Understanding RIGs might provide deeper insights into protein structure, binding, and folding mechanisms, as well as into protein stability and function.

To understand complex biological network data, one must be able to successfully reproduce them. Finding an adequate network model that generates networks that closely replicate the structure of real data is one of the first steps in this direction. Only a well-fitting network model that precisely reproduces the network structure and laws through which the network has emerged can enable understanding and replication of the biological processes and the underlying complex evolutionary mechanisms in the cell. The hope is that a good network model could provide insights into understanding of biological function, disease, and evolution.

Additionally, many graph theoretic problems are computationally hard that make the analyses of large biological networks infeasible. However, special graph classes often have well-known properties and solving many problems on such classes is feasible even though it is infeasible for graphs in general. Thus, finding a well-fitting graph class (i.e., network model) for biological networks could simplify their computational manipulation and enable easier extraction of biological knowledge that is encoded in their network topology.

Even though currently the PPI data is noisy and incomplete and the models are quite primitive, the models have already been used in practical biological applications to address realistic problems. For example, network motifs (which are believed to represent evolutionary conserved functional modules) are defined with respect to a random graph model [1, 2]. Similarly, network models are essential when motifs are used to classify real networks into super-families [3]. Furthermore, we used a well-fitting network model to de-noise PPI network data, i.e., to assign confidence levels to existing PPIs interactions, as well as to predict new interactions that were overlooked experimentally [4]. In 2004, a scale-free network model was used to guide biological experiments in a time- and cost-optimal way, thus minimizing the costs of interactome detection [5]. Finally, properties of a network model were used to develop computationally easy algorithms for PPI networks that are computationally intensive on graphs in general [6]. Since discovering PPI and other biological networks is in its infancy, it is expected that practical application of network models will increase and prove its value in the future.

Several network models have been proposed for biological networks. Starting with Erdös-Rényi random graphs [7], various network models have been designed to match certain properties of real-world networks. Early studies published largely incomplete yeast two-hybrid PPI data sets [8, 9] that were well modeled by scale-free networks [10, 11]. In a scale-free network, the distribution of degrees follows a power-law [12]. Modeling of the data by scale-free networks was based on the assumption that the degree distribution is one of the most important network parameters that a good network model should capture. However, networks of vastly different structures could have the same degree distributions [13]. Additionally, it has been argued that currently available PPI network data are samples of the full interactomes and thus the observed power-law degree distributions are artifacts of sampling properties of these networks [14, 15, 16]. As new biological network data becomes available, we need to ensure that our models continue to fit the data well. In the light of new PPI network data, several studies have started questioning the wellness of fit of scale-free models: an evidence has been presented that the

structure of PPI networks is closer to geometric random graphs, that model spatial relationships between objects, than to scale-free networks [17, 18, 19, 20, 21]. Similarly, geometric random graph model has been identified as an optimal network model for RIGs [22].

A well-fitting network model should generate graphs that closely resemble the structure of real-world networks. To evaluate the fit of a model to the data, one needs to compare model networks with real-world networks. However, network comparisons are computationally infeasible due to NP-completeness of the underlying subgraph isomorphism problem. Therefore, large network comparisons rely on heuristics, commonly called "network properties." These properties belong to two major classes: global and local. Global properties include the degree distribution, average clustering coefficient, clustering spectrum, average diameter, and the spectrum of shortest path lengths. Local properties include network motifs, small overrepresented subgraphs [2], and graphlets, small connected induced subgraphs of real-world networks (see Figure S1(a) in Supplementary Information (SI)) [17]. Based on graphlets, two highly sensitive measures of network local structural similarities were designed: the relative graphlet frequency distance ("RGF-distance") [17] and graphlet-based generalization of the degree distribution, called graphlet degree distribution agreement ("GDD-agreement") [18]. The choice of a network property for evaluating the fit of a network model to the data is non-trivial, since different models might be identified as optimal with respect to different properties. In general, global properties might not be constraining enough to capture complex topological characteristics of biological networks. For example, two networks with exactly the same degree distributions can have completely different underlying topologies (Figure S1(b) in SI). On the other hand, local properties, RGF-distance and GDD-agreement, impose a larger number of constraints, thus reducing degrees of freedom in which networks being compared can differ. The fit of model networks to real-world data can also be evaluated by using principal component analysis of the vector space whose coordinates are the statistics of network properties [23], as well as by counting the number of random walks of a given length in the network and feeding these counts into a machine learning classifier [24, 25].

1.2 Our Contribution

Previous studies evaluated the fit of a network model to the data with respect to a single network property [17, 18]. In this paper, we demonstrate that it might be difficult to assess the reliability of the fit of any particular network model to the data with respect to a single network property, since different models might be identified as optimal with respect to different properties (Figures S2(a) and S2(b) in SI). We also show that two networks with exactly the same value of one network property can have completely different network topologies (Figure S1(b) in SI). Thus, we introduce a novel approach that finds a consensus between network properties about the best fitting network model by integrating a variety of global and local network properties into the "network fingerprint." A "network fingerprint" is a vector whose coordinates are the following network properties: the average degree, the average clustering coefficient, the average diameter, and the frequencies of appearance of all 31 graphlets with 1 to 5 nodes (see Section 1 in SI). As such, our method imposes a large number of constraints on the networks being compared and reduces the number of degrees of freedom in which they can differ. The hope is that such an integrated approach will increase our confidence in the fit of a network model compared to when an individual network property is used for that purpose. Additionally, unlike previous studies [17, 18], our approach applies a series of machine learning classifiers (or just

"classifiers," for brevity) to network fingerprints to predict the best-fitting network model.

Our method proceeds through the following steps. First, we represent each real-world and model network with its fingerprint. Second, we use fingerprints of model networks as input into classifiers to train them. Third, we validate the prediction accuracy of each classifier. Next, we use network fingerprints of real-world networks as input into trained classifiers to predict their best-fitting network models. Finally, we provide several validations of our model predictions.

2 Methods

2.1 Data Sets

We need to distinguish between two different types of PPIs: binary interactions obtained by yeast two-hybrid (Y2H) technique and co-complex data obtained by mass spectrometry of purified complexes. Since in co-complex data interactions are defined by using either the "spoke" or the "matrix" model, binary interaction networks are believed to have fewer false positives than co-complex data [26]; in the spoke model, edges exist between the bait and each of the preys in a pull-down experiment, but not between the preys, while in the matrix model, additional edges are formed between all preys. However, due to technological limitations of Y2H, binary interaction networks still contain many false negatives and are thus incomplete [26]. Networks from large databases contain both binary and co-complex PPIs; this makes them more complete, but at the same time, they have high levels of false positives. Also, they seem to contain a larger fraction of interactions supported by a single publication [27].

We analyze physical PPI networks of four eukaryotic organisms: yeast *Saccharomyces cerevisiae*, fruitfly *Drosophila melanogaster*, worm *Caenorhabditis elegans*, and human *Homo sapiens*. We analyze the total of 12 PPI networks, 5 of which are yeast, 3 of which are fruitfly, 1 of which is worm, and 3 of which are human. We denote PPI networks as follows. "YH1" and "YE1" are the high-confidence and the entire yeast PPI networks by Collins et al. [28], respectively. "YH2" is the yeast high confidence PPI network described by von Mering et al. [29]. "YE2" is the yeast PPI network containing top 11,000 high-, medium-, and low-confidence interactions from the same study [29]. "YE3" is the entire physical yeast protein interaction network from BioGRID[1]. "FH1" and "FE1" are the high-confidence and the entire fruitfly PPI networks by Giot et al. [30], respectively. "FE2" is the entire physical fruitfly protein interaction network from BioGRID. "WE1" is the entire worm PPI network from BioGRID. Finally, "HE1" is the entire human PPI network by Rual et al. [31], while "HE2" and "HE3" are entire human PPI networks from BioGRID and HPRD[2], respectively. All five yeast PPI networks, as well as FE2, WE1, HE2, and HE3, contain both binary and co-complex data. The remaining networks, i.e., FH1, FE1, and HE1, contain solely binary interactions.

In addition to PPI networks, we apply our approach to network representations of protein structures, residue interaction graphs (RIGs). In RIGs, nodes represent amino acids and edges exist between residues that are close in space. We analyze RIGs constructed for nine structurally

[1] http://www.thebiogrid.org/
[2] http://www.hprd.org/

and functionally different proteins [22]. For each of the nine proteins, multiple RIGs are constructed as undirected and unweighted graphs, with residues *i* and *j* interacting if any heavy atom of residue *i* is within a given distance cut-off of any heavy atom of residue *j*. Various distance cut-offs in [4.0, 9.0] Å are used, as well as three different representations of residues: (1) RIGs that contain as edges only residue pairs that have heavy *backbone* atoms within a given distance cut-off ("BB"), (2) RIGs that contain as edges only residue pairs that have heavy *side-chain* atoms within a given distance cut-off ("SC"), and (3) the most commonly used RIG model, in which *all* heavy atoms of every residue are taken into account when determining residue interactions ("ALL"). In total, these different RIG definitions result in 513 RIGs corresponding to nine different proteins (see [22] for details).

2.2 Techniques

We apply five commonly used classifiers: backpropagation method ("BP"), probabilistic neural networks ("PNN"), decision tree ("DT"), multinomial naïve Bayes classifier ("MNB"), and support vector machine ("SVM") (see Section 1 in SI). We evaluate the fit of three different network models to the real-world networks: Erdös-Rényi random graphs ("ER") [7], preferential attachment scale-free networks ("SF") [12], and 3-dimensional geometric random graphs ("GEO") [32, 17] (see Section 1 in SI). In an ER network, edges are placed between pairs of nodes uniformly at random with the same probability p [7]. The version of SF networks that we use are generated by Barabási-Albert peripheral attachment method [12]. In a GEO network, nodes correspond to points in a metric space that are distributed uniformly at random and edges are created between pairs of nodes if the corresponding points in space are close enough according to a chosen distance norm [32]. We construct geometric random graphs by using 3-dimensional Euclidean boxes and the Euclidean distance norm [17]. We choose the distance cut-off for the existence of edges so that the resulting GEO graph is of the same size as the data set that it is modeling. We do not consider other commonly used network models, such as random graphs with the same degree distribution as the data, or the stickiness index-based network model [33], because generating these models requires as input the degree distribution of real-world networks, while the training and testing sets of random networks need to be generated without any data input.

We start by generating the set of 8,220 random networks of different sizes belonging to the three network models: ER, SF, and GEO (see Section 1 in SI). We divide these random networks into two sets: the "training set," containing 20% of them, and the "testing set," containing the remaining 80% of them. We choose this ratio for the training and the testing sets to achieve good training and generalization of classifiers, as well as to avoid data over-fitting. Next, we find fingerprints for these model networks and provide them as input into classifiers. We train the five classifiers on random networks from the training set, so that classifiers could learn to distinguish between fingerprints of random networks belonging to different models. Then, we validate prediction accuracies of classifiers on the testing set. That is, we examine how well classifiers work on new, yet unseen data, by analyzing whether they classify random networks from the testing set into their correct models. We define the validation rate of a classifier as the percentage of random networks from the testing set that are correctly classified. Thus, the validation rate can be interpreted as the likelihood that a classifier will classify a network to its correct model. The validation rates over the entire testing data set for BP, PNN, DT, MNB, and SVM are 99.98%, 99.97%, 99.41%, 98.48%, and 94.72%, respectively (column 2 of Table S1

in SI). Model-specific validation rates are presented in columns 3-5 of Table S1 in SI. These high validation rates indicate that all five classifiers are able to successfully classify random networks into their correct models. We also verify that the classifiers are robust to noise (see below), which is important since we are dealing with noisy PPI data. For these reasons, we believe that our approach correctly classifies biological networks into their best-fitting network models.

3 Results and Discussion

3.1 Results

The best-fitting network models for RIGs identified by each of the five classifiers are presented in Figure S3 in SI. For more than 94% of all analyzed RIGs, all five classifiers predict GEO as the best-fitting network model. This result is encouraging, since GEO models spatial relationships between objects, and therefore, it is expected to replicate well the underlying nature of spatially packed residues in a protein. Our result is consistent with a recent study that demonstrated, by using a variety of individual network properties, that GEO is the optimal network model for RIGs [22]. The RIGs that are better modeled by SF and ER networks are those that were constructed by using the lowest distance cut-offs for "SC" contact type and the highest distance cut-offs for "ALL" contact type (Section "Data Sets"). This is consistent with our previous results [22], therefore additionally validating the correctness of this study.

Next, we apply our approach to PPI networks, which are, unlike RIGs, noisy and incomplete, and therefore the identification of their optimal network model could be more challenging. The best-fitting network models for PPI networks predicted by each of the five classifiers are presented in Table 1. Classifiers predict GEO as the best-fitting network model for most of the analyzed yeast PPI networks: YH1, YE1, YH2, and YE2 (Table 1). This is encouraging, since yeast has the most complete interactome, as indicated by high edge densities and clustering coefficients of its PPI networks (Table 1). Additionally, yeast PPI networks that are fitted the best by GEO were obtained by merging and de-noising multiple PPI networks that contained both binary and co-complex interaction data [28, 29]. Moreover, all five classifiers predict GEO as the best-fitting network model for YH1 network, that is comparable to small-scale experiments by the quality of its interactions [28]. These results are consistent with studies that demonstrated the superiority of the fit of GEO to PPI networks of various organisms obtained by various biological techniques [17, 18, 19, 20, 21].

Out of the remaining PPI networks in Table 1, three are binary interaction data sets (FH1, FE1, and HE1), and five originate from large PPI databases, BioGRID and HPRD, that contain both binary and co-complex data (YE3, FE2, WE1, HE2, and HE3). Binary PPI networks are less complete than networks from large databases (Section "Data Sets") [26]. However, large databases contain a large fraction of interactions obtained by literature curation (LC). It has recently been argued that LC can be error-prone and possibly of lower quality than commonly believed [27, 26]. Given that more than 75% (85%) of the LC yeast (human) PPIs in BioGRID are supported by a single publication [27], the quality of these interactions might be questionable [26]. Moreover, a considerably low overlap between high-throughput experimental and LC PPIs in BioGRID, as well as a surprisingly low overlap of interactions across different databases [27], might suggest that many interactions still remain to be validated and

Data	|V|	|E|	Diam	CC	BP	PNN	DT	NBM	SVM
YH1	1,622	9,074	5.53	0.55	GEO	GEO	GEO	GEO	GEO
YE1	2,390	16,127	4.82	0.44	GEO	ER	GEO	GEO	GEO
YH2	988	2,455	5.19	0.34	GEO	GEO	SF	GEO	GEO
YE2	2,401	11,000	4.93	0.30	GEO	ER	SF	GEO	GEO
YE3	4,961	39,434	3.48	0.18	SF	ER	SF	SF	ER
FH1	4,602	4,637	9.44	0.02	SF	ER	ER	SF	ER
FE1	6,985	20,007	4.47	0.01	SF	SF	SF	SF	SF
FE2	7,040	22,265	4.34	0.01	SF	SF	SF	SF	SF
WE1	3,524	6,541	4.32	0.05	SF	SF	SF	SF	SF
HE1	1,873	3,463	4.34	0.03	SF	SF	SF	SF	ER
HE2	7,941	23,555	4.69	0.11	SF	SF	SF	SF	SF
HE3	9,182	34,119	4.26	0.10	SF	SF	SF	SF	SF

Table 1: The best-fitting network models (out of ER, GEO, and SF) predicted by the five classifiers (BP, PNN, DT, NBM, and SVM) for the 12 PPI networks. The PPI networks are presented in the first column, denoted by "Data." Columns two to five contain the number of nodes ("|V|"), the number of edges ("|E|"), the average diameter ("Diam"), and the average clustering coefficient of a network ("CC"), respectively. Columns six to ten contain network models predicted by the five classifiers for each of the PPI networks.

discovered [27, 26]. For these reasons, it is not surprising that SF and ER are the best-fitting models for binary Y2H PPI networks and for PPI networks from large databases (Table 1). Since PPI networks are unlikely to be organized completely at random, the best fit of ER to some of them additionally verifies the presence of noise. A good fit of SF to networks that are smaller samples of complete interactomes (obtained only by Y2H) is consistent with previous studies arguing that power-law degree distributions in PPI networks are an artefact of sampling [14, 15, 16].

3.2 Robustness and Validation

To test the robustness of our approach to noise, we randomly add, remove, and rewire 10%, 20%, and 30% of edges in YH1 network and its corresponding model networks and examine how the classifiers classify them (Table 2). We test the robustness on YH1, since it has been argued that the quality of its interactions is comparable to that of interactions produced by small-scale experiments [28]. Clearly, there is no need to introduce noise in ER networks, since they cannot be made more random. Note that random edge rewirings of ER networks would result in ER networks of the same densities, while random edge deletions and additions of ER networks would result in ER networks of lower and higher densities, respectively. Nonetheless, since we train and test our classifiers on ER networks of different densities (see Section "Techniques" above and Section 1 in SI), and since their validation rates do not depend on densities of ER networks, there is no need to test the robustness of our method on "randomized" ER networks.

It is expected that with the introduction of more noise of ER type into the data and SF and GEO model networks, noisier networks will increasingly be classified as ER. Indeed, SVM classifies SF networks with 20-30% of edges deleted and rewired as ER (Table 2). At lower levels of noise, all classifiers predict noisy SF to still be SF (Table 2). Thus, noisy SF (and clearly, ER)

Network	Classifier	add10	add20	add30	del10	del20	del30	rew10	rew20	rew30
YH1	BP	GEO	GEO	GEO	GEO	SF	SF	SF	SF	SF
	PNN	GEO	GEO	GEO	GEO	GEO	GEO	GEO	GEO	GEO
	DT	GEO	GEO	SF	GEO	GEO	GEO	GEO	SF	SF
	MNB	GEO	GEO	GEO	GEO	GEO	GEO	GEO	GEO	GEO
	SVM	GEO	GEO	ER	GEO	GEO	GEO	GEO	GEO	ER
GEO_{YH1}	BP	GEO	GEO	GEO	GEO	GEO	GEO	GEO	GEO	SF
	PNN	GEO	GEO	GEO	GEO	GEO	SF	GEO	GEO	SF
	DT	SF	SF	SF	ER	ER	ER	SF	SF	SF
	MNB	GEO	GEO	GEO	GEO	GEO	GEO	GEO	GEO	GEO
	SVM	GEO	GEO	SF	SF	SF	SF	GEO	SF	SF
SF_{YH1}	BP	SF	SF	SF	SF	SF	SF	SF	SF	SF
	PNN	SF	SF	SF	SF	SF	SF	SF	SF	SF
	DT	SF	SF	SF	SF	SF	SF	SF	SF	SF
	MNB	SF	SF	SF	SF	SF	SF	SF	SF	SF
	SVM	SF	SF	SF	SF	ER	ER	SF	ER	ER

Table 2: The best-fitting network models (ER, GEO, SF) predicted by the five classifiers (BP, PNN, DT, NBM, and SVM) for noisy networks. The networks to which the noise is added are: YH1 network, as well as a GEO and an SF network of the same size as YH1, denoted by "GEO_{YH1}" and "SF_{YH1}", respectively (listed in column 1). We obtained noisy networks by randomly adding, deleting, and rewiring 10%, 20%, and 30% of edges (columns 3-11, respectively). For each of YH1, GEO_{YH1} and SF_{YH1} and for each of the randomization schemes, we constructed 10 instances of noisy (randomized) networks, resulting in the total of $3 \times 9 \times 10 = 270$ noisy networks. For each of YH1, GEO_{YH1} and SF_{YH1}, the classifiers predicted the same model for all instances of noisy networks in the same randomization scheme; predicted models are reported in columns 3-11.

are never classified as GEO. Similarly, increasing levels of noise in GEO networks cause their increasing miss-classification into ER or SF models. Thus, noisy GEO can be classified as either GEO, SF, or ER. This demonstrates that our approach is unlikely to classify a real-world network that has a noisy SF or ER topology as GEO. On the other hand, it might classify a real-world network that has a noisy GEO topology either as GEO, SF, or ER. Thus, the yeast PPI networks that are classified as GEO are unlikely to have SF or ER network structure. However, PPI networks of any organism that are classified as SF or ER could have noisy GEO structure.

The classifier that is the most robust to noise is MNB, since it always correctly predicts the model irrespective of the level of noise (Table 2). The least robust classifier seems to be DT, since it always predicts noisy GEO networks as SF or ER. Note however, that this is not surprising, since small changes in the input of a decision tree may cause large changes in its output due to a relative sensitivity of branching to the input values. For this reason, it is not surprising that DT incorrectly classifies YH2 and YE2 networks that are predicted to be GEO by most other classifiers (Table 1).

We take a step further towards validating our results. We apply an algorithm that directly tests whether PPI networks have a geometric structure by embedding the proteins into a low-dimensional space given only their PPI network connectivity information [19]. We embed in 3-dimensional (3D) Euclidian space, simply as a proof of concept. The algorithm is based on multidimensional scaling, with shortest path lengths between protein pairs in a PPI network playing the role of Euclidean distances in space. After proteins are embedded in space, a radius r is chosen so that each node is connected to the nodes that are at most at distance r

from it; this procedure results in construction of a geometric graph (as defined in Section 1 in SI). Each choice of a radius thus corresponds to a different geometric graph. By varying the radius, specificity and sensitivity are measured to quantify the ability of each constructed geometric graph to recover the original PPI network. Then, the overall goodness of fit is judged by computing the areas under the Receiver Operator Characteristic (ROC) curves, with higher values indicating a better fit. For details, see [19].

We apply this algorithm to YH1 PPI network, as well as to ER, GEO, and SF model networks of the same size as YH1. As expected, the resulting areas under the ROC curve (AUCs) are low for ER and SF, with values of 0.65 and 0.56, respectively, since these networks do not have a geometric structure (Figure S4(a) in SI). On the other hand, AUCs are high for the data and GEO, with values of 0.89 and 0.98, respectively, suggesting that the data has a geometric structure (Figure S4(a) in SI). For each of the network models, the reported AUC is the average over 10 random graphs.

Since PPI networks are noisy, we test how robust the embedding algorithm is to noise in the data and model networks. We add noise to YH1 and its corresponding ER, SF, and GEO model networks by randomly deleting, adding, and rewiring 10%–50% of their edges. We embed these randomized networks into 3D Euclidian space and compute their AUCs. Noise barely improves the embedding of SF or ER (Figure S4(b) in SI) suggesting that the data is unlikely to have a noisy SF or ER structure; note that with edge deletions and additions the size of the networks changes affecting the quality of the embedding and thus, unlike above, we analyze "randomized" ER. However, noise has different effects on the embedding of GEO. Random edge deletions do not disturb the quality of the geometric embedding, since edge deletions have little effect on shortest path lengths in GEO networks. Therefore, AUCs for GEO networks obtained by random edge deletions are almost the same as AUCs for non-randomized GEO networks (Figure S4(b) in SI). On the other hand, shortest path lengths decrease with random edge additions and rewirings in GEO networks, resulting in worse embeddings and lower AUCs (Figure S4(b) in SI). Similar is observed for YH1: random edge deletions do not affect the quality of the embedding, whereas random edge additions and rewirings result in lower AUCs (Figure S4(b) in SI). The comparable behaviors of GEO and YH1 suggest that they have similar structures, thus additionally validating our network model predictions. Moreover, AUC value of 0.87 for GEO with 10% of randomly rewired edges is very close to AUC value of 0.89 for YH1 (Figure S4 in SI). Thus, the structure of the PPI data appears to be consistent with the structure of a noisy GEO.

3.3 Comparison with Other Studies

Filkov *et al.* use seven network properties to describe a real-world network and compare it with model networks [23]. In comparison, we use 34 properties, therefore decreasing the number of degrees of freedom in which networks being compared can vary. Also, the methodology used by Filkov *et al.* is different than ours. First, they evaluate the fit of a model to the data by using principal component analysis of the vector space whose coordinates are the statistics of the seven network properties that they analyzed. Second, they evaluate the fit of two scale-free network models to the data: SF and their new scale-free model of network growth via sequential attachment of linked node groups. In comparison, we use three network models that have very different network structure: ER, SF, and GEO.

Middendorf *et al.* measure the topological structure of a network by counting the number of random walks of a given length [24, 25] and giving those counts as input into classifiers. Random walks are different than graphlets in several ways. First, graphlets are induced and random walks are not. Second, nodes and edges can be repeated in a random walk, while a graphlet consists of a unique set of nodes and edges. Middendorf *et al.* use two classifiers, SVM [24] and DT [25] to discriminate different network models. That is, in each study, they use a single classifier to predict the best fitting network model for a real-world network. In comparison, in this study we use five different classifiers, all supporting GEO as the best-fitting model. We show that DT and SVM are the least robust out of the five classifiers that we analyzed (see Section "Robustness and Validation" and Table 2). Moreover, the training set of Middendorf *et al.* contains model networks of the size of the data only and thus it could be biased by the model properties that are enforced by the chosen network size. In comparison, we train our classifiers on random networks of different sizes (Section "Techniques" above and Section 1 in SI) to allow for a possibility to predict the best fitting network model for any yet unseen real-world network, independent of its size. Middendorf *et al.* consider SF, ER, and small-world networks and identify SF-based duplication-mutation models as the best-fitting models for biological networks. Given that Middendorf *et al.* did not consider GEO in their studies, and given a low robustness of DT and SVM that they used, their reported best fit of SF-based models to the data could be questioned.

3.4 Discussion

We further elaborate on the power of integration of different network properties as opposed to using individual ones to asses the fit of a network model to the data. We use our GraphCrunch software package [34] to evaluate the fit of ER, GEO, and SF models to all PPI networks described in Section "Data Sets." GraphCrunch evaluates the fit of the models to the data with respect to seven local and global network properties. When we evaluate the fit of the data to the models with respect to each of the seven properties, we obtain inconclusive results, because each of the properties favors a different model. For example, as illustrated in Figure S2(a) in SI, SF fits YH1 the best with respect to the degree distribution, but GEO is the best-fitting network model with respect to the clustering spectrum (Figure S2(b) in SI). This demonstrates the need for a method that finds a consensus between models suggested by different network properties. We propose such a method in this study. Since our method integrates a variety of network properties, it imposes a large number of constraints on the networks being compared and reduces the number of degrees of freedom in which they can differ, thus increasing the confidence in the fit of a network model. Inclusion of additional network properties could further increase the confidence at the expense of an increased computational complexity.

Although several studies proposed GEO as a well-fitting null model for PPI networks [17, 18, 19, 20, 21], a recent study questioned this [35]. Note however, that this conclusion was based on analyzing only one eukaryotic and one prokaryotic PPI network [35], each from DIP[3]. Thus, in the light of low quality and incompleteness of the data from large databases [27] (Section "Results"), no conclusions about the fit of GEO should have been made. The authors argued that low-dimensional geometric random graphs might not be able to capture high abundance of dense graphlets and bipartite subgraphs observed in real-world networks, neglecting two obvious alternatives for reconciling the differences in the abundance of subgraphs in the data

[3] http://dip.doe-mbi.ucla.edu/dip/Main.cgi

and in GEO: (1) they based their conclusion on the observation that bipartite graphlet 20 cannot exist in 2-dimensional (2D) GEO, even though it exists in 3D GEO as well as in all higher dimensions (Figure S2(c) in SI); and (2) GEO graphs with the same number of nodes, but 2.5 times more edges than the data have very similar abundance of all graphlets as the data (Figure S2(d) in SI). These observations are important since: (1) the optimal dimension for the space of PPI networks is unknown and finding it is a non-trivial research problem, but it is highly unlikely that PPI networks exist in a 2D space; and (2) the density of real-world data will continue to increase [36, 37], most likely in accordance with its network model. Also note that over-abundance of graphlets in the currently available PPI networks could be an artefact of the "matrix" and "spoke" models used to determine PPIs in affinity purification followed by mass spectrometry (AP/MS) pull-down experiments. In the matrix model, interactions are defined between all proteins in a purified complex, clearly resulting in over-abundance of dense graphlets. In the spoke model, interactions are defined between a bait and each of its preys, but not between the preys, clearly resulting in over-abundance of sparse graphlets; an overlap between preys in different purifications results in over-abundance of complete bipartite graphs.

Finally, it is possible that PPI networks are not completely geometric and that another random graph model would provide a better fit. Examining the fit of various other random network models to the data is the subject of future research. Additionally, it is possible that different parts of PPI networks have different structure. The two most commonly used high-throughput PPI detection methods, AP/MS and Y2H, are fundamentally different: AP/MS detects mostly stable protein complexes, whereas Y2H detects mostly transient signalling interactions [37]. Thus, the two methods examine different, complementary subspaces within the interactome, resulting in networks with different topological and biological properties [37]. Since proteins within a protein complex are close in the cell, it is possible that protein complexes have a geometric structure. In contrast, transient interactions in signalling pathways might have a different structure, such as that of bipartite graphs or scale-free networks.

Acknowledgements

This project was supported by the NSF CAREER IIS-0644424 grant.

References

[1] S. S. Shen-Orr, R. Milo, S. Mangan, and U. Alon. Network motifs in the transcriptional regulation network of escherichia coli. *Nature Genetics*, 31:64–68, 2002.

[2] R. Milo, S. S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network motifs: simple building blocks of complex networks. *Science*, 298:824–827, 2002.

[3] R. Milo, S. Itzkovitz, N. Kashtan, R. Levitt, S. Shen-Orr, I. Ayzenshtat, M. Sheffer, and U. Alon. Superfamilies of evolved and designed networks. *Science*, 303:1538–1542, 2004.

[4] O. Kuchaiev, M. Rasajski, D.J. Higham, and N. Pržulj. Geometric de-noising of protein-protein interaction networks. *PLoS Computational Biology*, 5(8): e1000454. doi:10.1371, 2009.

[5] M. Lappe and L. Holm. Unraveling protein interaction networks with near-optimal efficiency. *Nature Biotechnology*, 22(1):98–103, 2004.

[6] N. Pržulj, D. G. Corneil, and I. Jurisica. Efficient estimation of graphlet frequency distributions in protein-protein interaction networks. *Bioinformatics*, 22(8):974–980, 2006.

[7] P. Erdös and A. Rényi. On random graphs. *Publicationes Mathematicae*, 6:290–297, 1959.

[8] T. Ito, K. Tashiro, S. Muta, R. Ozawa, T. Chiba, M. Nishizawa, K. Yamamoto, S. Kuhara, and Y. Sakaki. Toward a protein-protein interaction map of the budding yeast: A comprehensive system to examine two-hybrid interactions in all possible combinations between the yeast proteins. *Proc Natl Acad Sci U S A*, 97(3):1143–7, 2000.

[9] P. Uetz, L. Giot, G. Cagney, T. A. Mansfield, R. S. Judson, J. R. Knight, E. Lockshon, V. Narayan, M. Srinivasan, P. Pochart, A. Qureshi-Emili, Y. Li, B. Godwin, D. Conover, T. Kalbfleish, G. Vijayadamodar, M. Yang, M. Johnston, S. Fields, and J. M. Rothberg. A comprehensive analysis of protein-protein interactions in saccharomyces cerevisiae. *Nature*, 403:623–627, 2000.

[10] A.L. Barabási and Z. N. Oltvai. Network biology: Understanding the cell's functional organization. *Nature Reviews*, 5:101–113, 2004.

[11] H. Jeong, S. P. Mason, A.-L. Barabási, and Z. N. Oltvai. Lethality and centrality in protein networks. *Nature*, 411(6833):41–2, 2001.

[12] A.L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

[13] R. Tanaka, T.-M. Yi, and J. Doyle. Some protein interaction data do not exhibit power law statistics. *FEBS letters*, 579:5140–5144, 2005.

[14] M. P. H. Stumpf, C. Wiuf, and R. M. May. Subnets of scale-free networks are not scale-free: Sampling properties of networks. *Proceedings of the National Academy of Sciences of the United States of America*, 102(12):4221–4224, March 2005.

[15] J. D. H. Han, D. Dupuy, N. Bertin, M. E. Cusick, and Vidal. M. Effect of sampling on topology predictions of protein-protein interaction networks. *Nature Biotechnology*, 23:839–844, 2005.

[16] E. de Silva, T. Thorne, P. J. Ingram, I. Agrafioti, J. Swire, C. Wiuf, and M. P. H. Stumpf. The effects of incomplete protein interaction data on structural and evolutionary inferences. *BMC Biology*, 4:39+, November 2006.

[17] N. Pržulj, D. G. Corneil, and I. Jurisica. Modeling interactome: Scale-free or geometric? *Bioinformatics*, 20(18):3508–3515, 2004.

[18] N. Pržulj. Biological network comparison using graphlet degree distribution. *Bioinformatics*, 23(2):e177–e183, 2007.

[19] D.J. Higham, M. Rasajski, and N. Pržulj. Fitting a geometric graph to a protein-protein interaction network. *Bioinformatics*, 24(8):1093–1099, 2008.

[20] O. Kuchaiev and N. Pržulj. Learning the structure of protein-protein interaction networks. *Pacific Symposium on Biocomputing*, pages 39–50, 2009.

[21] N. Pržulj, O. Kuchaiev, A. Stevanović, and W. Hayes. Geometric evolutionary dynamics of protein interaction networks. *Pacific Symposium on Biocomputing*, 15:178-189, 2010.

[22] T. Milenković, I. Filippis, M. Lappe, and N. Pržulj. Optimized null model for protein structure networks. *PLOS One*, 4(6):e5967, 2009.

[23] V. Filkov, Z.M. Saul, S. Roy, R.M D'Souza, and P.T. Devanbu. Modeling and verifying a broad array of network properties. *EPL*, 86(doi: 10.1209/0295-5075/86/28003), 2009.

[24] M. Middendorf, E. Ziv, C. Adams, J. Hom, R Koytcheff, C. Levovitz, G. Woods, L. Chen, and C. Wiggins. Discriminative topological features reveal biological network mechanisms. *BMC Bioinformatics*, 5:181(doi:10.1186/1471-2105-5-181), 2004.

[25] M. Middendorf, E. Ziv, and C. Wiggins. Inferring network mechanisms: The drosophila melanogaster protein interaction network. *PNAS*, 102(9):3192–3197, 2005.

[26] K. Venkatesan, J.F. Rual, A. Vazquez, U. Stelzl, I. Lemmens, T. Hirozane-Kishikawa,T. Hao, M. Zenkner, X. Xin, K.I. Goh, M.A. Yildirim, N. Simonis, K. Heinzmann, F. Gebreab, J.M. Sahalie, S. Cevik, C. Simon, A.S. de Smet,E. Dann, A. Smolyar, A. Vinayagam, H. Yu, D. Szeto, H. Borick, A. Dricot, N. Klitgord, R.R. Murray, C. Lin, M. Lalowski, J. Timm, K. Rau, C. Boone, P. Braun, M.E. Cusick, F.P. Roth, D.E. Hill, J. Tavernier, E.E. Wanker, A.L. Barabsi, and M. Vidal. An empirical framework for binary interactome mapping. *Nature Methods*, 6(1):83–90, 2009.

[27] M. E. Cusick, H. Yu, A. Smolyar, K. Venkatesan, A. R. Carvunis, N. Simonis, J. F. Rual, H. Borick, P. Braun, M. Dreze, J. Vandenhaute, M. Galli, J. Yazaki, David E. Hill, J. R. Ecker, F. P. Roth, and M. Vidal. Literature-curated protein interaction datasets. *Nature Methods*, 6(1):39–46, December 2008.

[28] S.R. Collins, P. Kemmeren, X.C. Zhao, J.F. Greenblatt, F. Spencer, F.C. Holstege, J.S. Weissman, and N.J. Krogan. Toward a comprehensive atlas of the physical interactome of saccharomyces cerevisiae. *Molecular and Cellular Proteomics*, 6(3):439–450, 2007.

[29] C. von Mering, R. Krause, B. Snel, M. Cornell, S. G. Oliver, S. Fields, and P. Bork. Comparative assessment of large-scale data sets of protein-protein interactions. *Nature*, 417(6887):399–403, May 2002.

[30] L Giot, JS Bader, C Brouwer, A Chaudhuri, B Kuang, Y Li, YL Hao, CE Ooi, B Godwin, E Vitols, G Vijayadamodar, P Pochart, H Machineni, M Welsh, Y Kong, B Zerhusen, R Malcolm, Z Varrone, A Collis, M Minto, S. Burgess, L McDaniel, E Stimpson, F Spriggs, J Williams, K. Neurath, N Ioime, M Agee, E Voss, K Furtak, R Renzulli, N Aanensen, S Carrolla, E Bickelhaupt, Y Lazovatsky, A DaSilva, J Zhong, CA Stanyon, RL Jr Finley, KP White, M Braverman, T Jarvie, S Gold, M Leach, J Knight, RA Shimkets, MP McKenna, J Chant, and JM Rothberg. A protein interaction map of drosophila melanogaster. *Science*, 302(5651):1727–1736, 2003.

[31] J.F. Rual, K. Venkatesan, T. Hao, T. Hirozane-Kishikawa, A. Dricot, N. Li, G. F. Berriz, F. D. Gibbons, M. Dreze, N. Ayivi-Guedehoussou, N. Klitgord, C. Simon, M. Boxem,

S. Milstein, J. Rosenberg, D. S. Goldberg, L. V. Zhang, S. L. Wong, G. Franklin, S. Li, J. S. Albala, J. Lim, C. Fraughton, E. Llamosas, S. Cevik, C. Bex, P. Lamesch, R. S. Sikorski, J. Vandenhaute, H. Y. Zoghbi, A. Smolyar, S. Bosak, R. Sequerra, L. Doucette-Stamm, M. E. Cusick, D. E. Hill, F. P. Roth, and M. Vidal. Towards a proteome-scale map of the human protein-protein interaction network. *Nature*, 437:1173–78, 2005.

[32] M. Penrose. *Geometric Random Graphs*. Oxford University Press, 2003.

[33] N. Pržulj and D.J. Higham. Modelling protein-protein interaction networks via a stickiness index. *Journal of the Royal Society Interface*, 3(10):711–716, 2006.

[34] T. Milenković, J. Lai, and N. Pržulj. Graphcrunch: a tool for large network analyses. *BMC Bioinformatics*, 9(70), 2008.

[35] R. Colak, F. Hormozdiari, F. Moser, A. Schonhuth, J. Holman, M. Ester, and S. C. Sahinalp. Dense graphlet statistics of protein interaction and random networks. *Pac Symp Biocomput*, pages 178–89, 2009.

[36] M. P. H. Stumpf, T. Thorne, E. de Silva, R. Stewart, H. J. An, M. Lappe, and C. Wiuf. Estimating the size of the human interactome. *Proceedings of the National Academy of Sciences*, 105(19):6959–6964, May 2008.

[37] H. Yu, P. Braun, M. A. Yildirim, I. Lemmens, K. Venkatesan, J. Sahalie, T. Hirozane-Kishikawa, F. Gebreab, N. Li, N. Simonis, T. Hao, J.F. Rual, A. Dricot, A. Vazquez, Ryan R. Murray, C. Simon, L. Tardivo, S. Tam, N. Svrzikapa, C. Fan, A. S. de Smet, A. Motyl, M. E. Hudson, J. Park, X. Xin, M. E. Cusick, T. Moore, C. Boone, M. Snyder, F. P. Roth, A. L. Barabasi, J. Tavernier, D. E. Hill, and M. Vidal. High-quality binary protein interaction map of the yeast interactome network. *Science*, 322(5898):104–110, October 2008.

Supplementary information for:

An integrative approach to modeling biological networks

Vesna Memišević[1], Tijana Milenković[1], and Nataša Pržulj[2,*]

[1]Department of Computer Science, University of California, Irvine, CA 92697-3435, USA
[2]Department of Computing, Imperial College London, London, SW7 2AZ, UK
[*]Corresponding author (e-mail: natasha@imperial.ac.uk)

1 Methods

1.1 Network fingerprint

We summarize the structure of a complex network by the notion of the "network fingerprint" (or just "fingerprint," for brevity). We define the *fingerprint* to be a 34-dimensional vector whose coordinates contain the following network properties: the average degree, average clustering coefficient, average diameter, and frequencies of the appearance of all 31 1-5-node graphlets. The *degree* of a node is the number of edges incident to the node; the *average degree* of a network is the average of degrees over all nodes in the network. The *clustering coefficient* of a node is defined as the probability that two neighbors of the node are themselves connected. The average of clustering coefficients over all nodes in a network is the *average clustering coefficient* of the network. The smallest number of links that have to be traversed in a network to get from one node to another is called the *distance* between the two nodes and a path through the network that achieves this distance is called the *shortest path* between the nodes; the average of shortest path lengths over all pairs of nodes in a network is called the *average network diameter*. *Graphlets* are small connected non-isomorphic induced subgraphs of a large network [1]; we count the occurrences of the only 1-node graphlet, a node, the only 2-node graphlet, an edge, and all 29 3-5-node graphlets (shown in Figure S1(a)). Because different coordinates of a network fingerprint can differ by several orders of magnitude, we normalize each coordinate to avoid domination of coordinates having large values. We normalize the i^{th} coordinate x_i of the network fingerprint *x* as $log(x_i + 1)$, for $i = 1, ..., 34$; we add 1 to x_i to avoid the logarithm function to go to infinity when $x_i = 0$.

1.2 Random network models

We consider three random network models: Erdös-Rényi (ER) random graphs [2], scale-free Barabási-Albert (SF) networks [3], and geometric (GEO) random graphs [4]. In *Erdös-Rényi random graphs*, edges between pairs of nodes are distributed uniformly at random with the same probability p [2]. *Scale-free networks* are networks that have power-law degree distributions. The version of SF networks that we use are generated by Barabási-Albert peripheral attachment method [3], in which newly added nodes preferentially attach to existing nodes with

probabilities proportional to their degrees. In *geometric random graphs*, nodes correspond to uniformly distributed points in a metric space and edges are created between pairs of nodes if the corresponding points are close enough in the metric space according to some distance norm [4]. We construct geometric random graphs by using 3-dimensional Euclidean boxes and the Euclidean distance norm [1].

For each of the three random network models, we generate 10 instances of random networks per model. We generate random networks of different sizes, both in terms of the number of nodes (n) and the number of edges (m). We use the following 28 values for n: 100, 200, 300, 400, 500, 600, 700, 800, 900, 1,000, 1,100, 1,200, 1,300, 1,400, 1,600, 2,100, 2,600, 3,100, 3,600, 4,100, 4,600, 5,000, 6,000, 7,000, 8,000, 9,000, 10,000, and 11,000. For each of the 26 values of n below 10,000, we vary $k = m/n$ from 1 to 10, in increments of 1. Due to the increase in computational complexity with the increase in the number of nodes and edges, for the 2 largest values of n, $n = 10,000$ and $n = 11,000$, we only use $k = 1, ..., 7$. Thus, we analyze $26 \times 10 + 2 \times 7 = 274$ different network sizes. In total, for the 3 network models, 10 random network instances per model, and 274 network sizes, we create $3 \times 10 \times 274 = 8,220$ model networks.

1.3 Machine learning classifiers

We use five well-known machine learning classifiers: backpropagation method (BP), probabilistic neural networks (PNN), decision tree (DT), multinomial naïve Bayes classifier (MNB), and support vector machine (SVM).

Both **BP** [5] and **PNN** [6] are based on artificial neural networks (ANNs). ANNs are simplified mathematical models of biological nervous systems built of processing units called neurons. Neurons in ANNs have many input signals and they produce one output signal. They are organized into the following layers: the input layer, one or more hidden layers, and the output layer. Neurons in the input layer do not perform any processing; instead, they only distribute the input data to all neurons in the first hidden layer. The number of hidden layers depends on implementation of an ANN. We use the standard implementations of BP and PNN from Neural-Network Toolbox in Matlab[1]. For the completeness of the manuscript, we briefly outline them below.

In our implementation of **BP** [5], the input layer consists of 34 neurons corresponding to the 34 coordinates of the network fingerprint input vector. To match the length of our input vector, we implement one hidden layer with 15 neurons; varying the number of neurons in the hidden layer between 10 and 20 had marginal effect on the results. The output layer contains three neurons, according to the "1-of-N encoding of the output classes" principle [7]: the number of neurons in the output layer matches the number of possible "output classes," i.e., random network models (ER, SF, and GEO). Thus, for a given output class, the neuron corresponding to the class is set to 1, whereas the remaining two neurons are set to -1. After BP computes the values on the three output neurons for an input vector, it classifies the input into an output class that corresponds to the neuron with the largest value.

All neurons in the input layer are connected with all neurons in the hidden layer. Similarly, all neurons in the hidden layer are connected with all neurons in the output layer. All of these

[1] http://www.mathworks.com/access/helpdesk/help/toolbox/nnet/

connections are weighted. Each neuron in the hidden and the output layer produces output by applying a non-linear "transfer function" to calculate a weighted sum of its inputs. We use *logsig* and *tansig* transfer functions in the hidden layer and the output layer, respectively. Initially, all weights are assigned randomly. Weights are adjusted gradually trough a training (learning) process: BP keeps adjusting the weights until the error between the value of each output neuron and its desired value (i.e., the value of the class that the input that we are training BP on belongs to) is $\leq 10^{-5}$. We use *trainscg* "training function" and set the learning rate to 0.01. Given these parameters, BP is successfully trained on the training set (defined in Techniques section above) in 588 epochs.

PNN that we use consists of the radial basis layer and the competitive layer. The radial basis layer further consists of the input and pattern sublayers. Similarly, the competitive layer consists of the summation and the output sublayers. The number of neurons in the input sublayer corresponds to the 34 dimensions of the input vector. The pattern sublayer consists of three pools of "pattern" neurons, where each pool corresponds to one of the three output classes. The number of neurons in each pool is determined as follows. As each network fingerprint from the training set is provided as input vector into PNN during the training process, a new neuron is added to the pool that corresponds to the output class (i.e., network model) of the input vector. After the training phase, when an input vector is presented to the trained PNN, the pattern sublayer computes how close the input vector is to each of the vectors from the training set in each pool. This information is sent to the summation sublayer. The summation sublayer consists of three neurons, where each neuron corresponds to one of the three output classes. Input into each neuron in the summation sublayer is the collection of outputs from the corresponding pool in the pattern sublayer. The output of each summation sublayer neuron is a weighted sum of all its inputs. Each of the three sums represents the probability that the input vector belongs to the corresponding class. Given these probabilities, the output sublayer, consisting of a single neuron, outputs the class having the highest probability.

We use a standard implementation of **DT** [8, 9] from Statistics Toolbox in Matlab[2]. Interior nodes in the decision tree are queries on certain attributes; in our case, attributes are the coordinates of the fingerprint vector. Each leaf in the tree corresponds to one of the three output classes. Branches in the tree represent conjunctions of attributes that lead to classification into the output classes. DT recursively splits the training set of input vectors into subsets based on the values of their coordinates; this corresponds to branching in the tree. DT continues to do so until the training input vectors are assigned to their correct classes.

We use a standard implementation of **MNB** [9] from WEKA [10], a publicly available collection of machine learning algorithms for data mining. MNB classifies the input data based on the Bayes' rule by selecting a class that maximizes the posterior probability of the class, given the training set. MNB does not use the assumption of a naïve Bayes classifier, that all data attributes are independent of each other.

We use a standard implementation of **SVM** [9, 11] from WEKA. SVM maps our 34-dimensional input vectors into a high dimensional space; the space dimension is automatically determined by WEKA. During the training phase, SVM finds an optimized data division within this space by constructing a hyperplane that optimally separates the data into two classes; since there are many hyperplanes that might classify the data, the hyperplane is chosen so that the distance from the hyperplane to the nearest data point is maximized. We generalize this binary classi-

[2]http://www.mathworks.com/access/helpdesk/help/toolbox/stats/

fication to the multiclass classification, with three classes corresponding to the three random network models. We do so by using three binary "one-versus-all" SVMs: for each of the three classes, its corresponding SVM either classifies the input data as belonging to the class ("positive classification"), or not belonging to the class ("negative classification") [9, 11]. Each of these three binary SVMs produces an output function that gives a relatively large value for a positive classification and a relatively small value for a negative classification. The input data is classified into the class with the highest value of the output function.

2 Supplementary figures

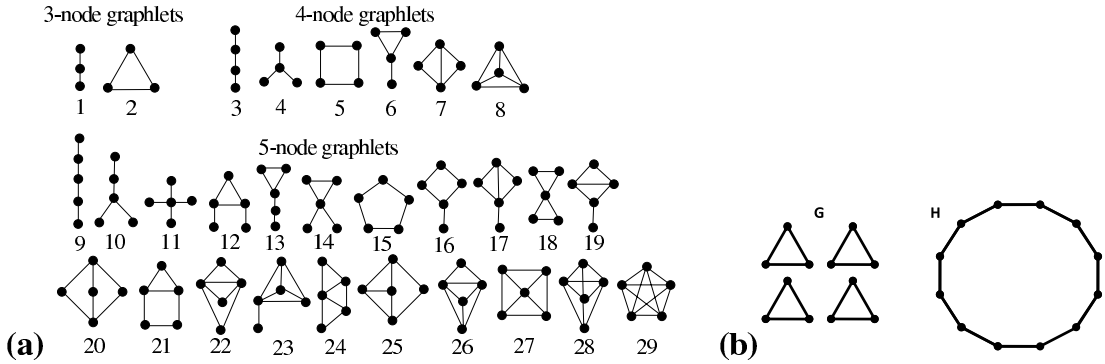


Figure S1: (a) All 3-node, 4-node, and 5-node graphlets [1]; (b) an example of two networks of the same size, G and H, that have the same degree distribution, but very different network structure.

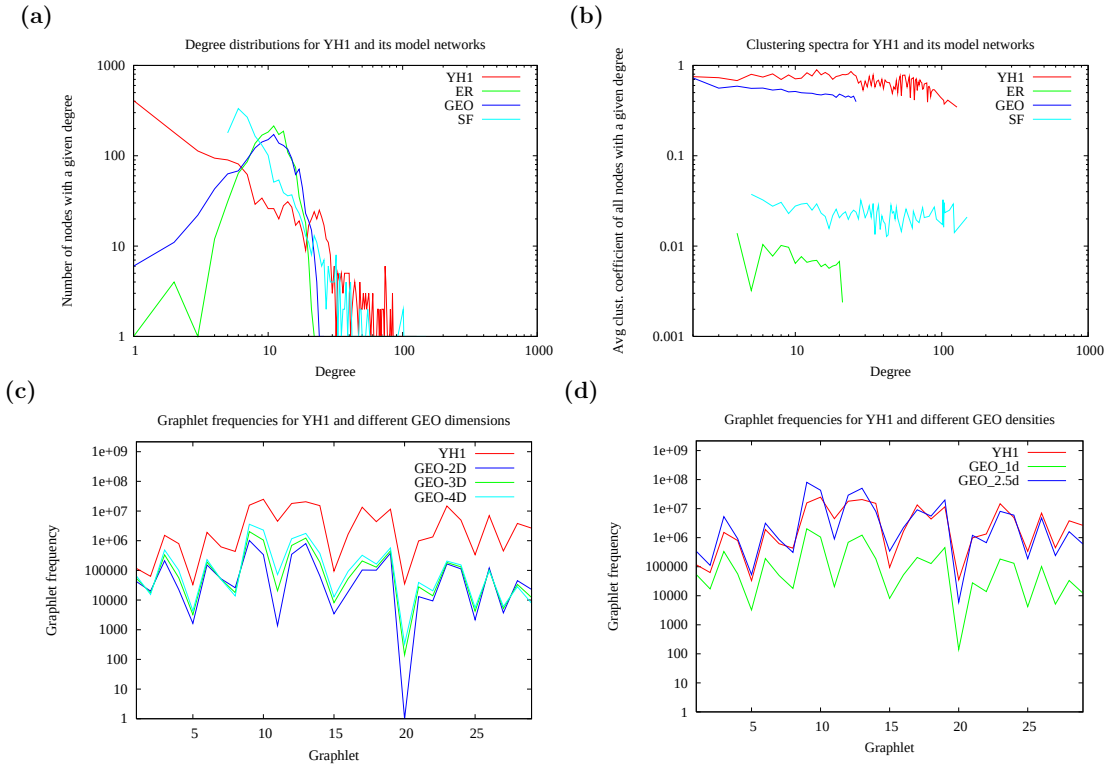


Figure S2: (a) Degree distributions and (b) clustering spectra for YH1, and an ER, a GEO, and an SF model network of the same size as YH1. (c) Graphlet frequencies for YH1, a 2-dimensional GEO ("GEO-2D"), a 3-dimensional GEO ("GEO-3D"), and a 4-dimensional GEO ("GEO-4D") network of the same size as YH1. (d) Graphlet frequencies for YH1, a GEO network with the same number of nodes and edges as YH1 ("GEO_1d"), and a GEO network with the same number of nodes, but 2.5 times as many edges as YH1 ("GEO_2.5d"). On horizontal axes in panels c and d, graphlets are numbered as in Figure S1.

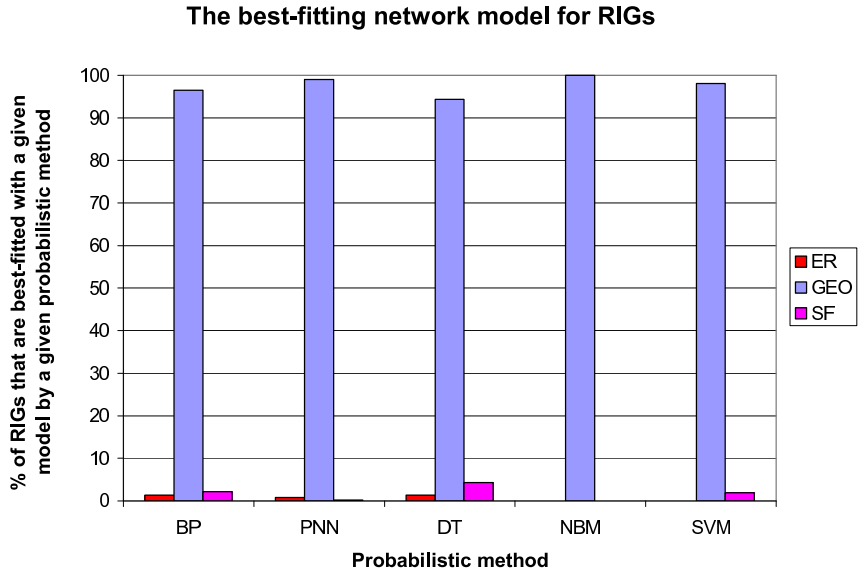
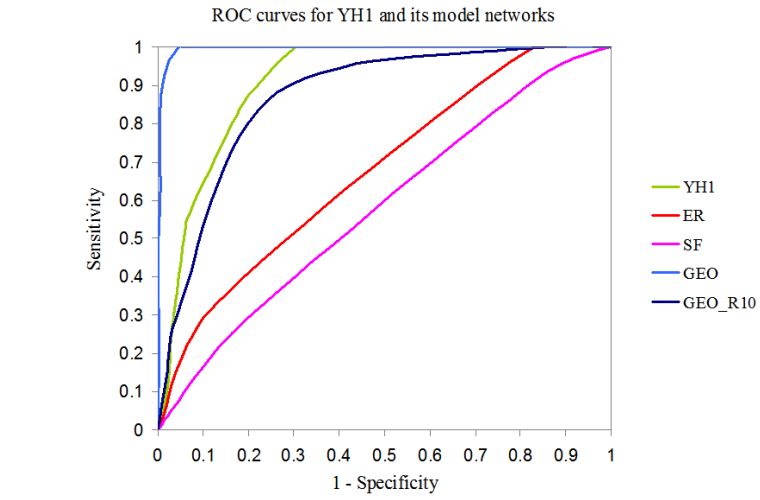
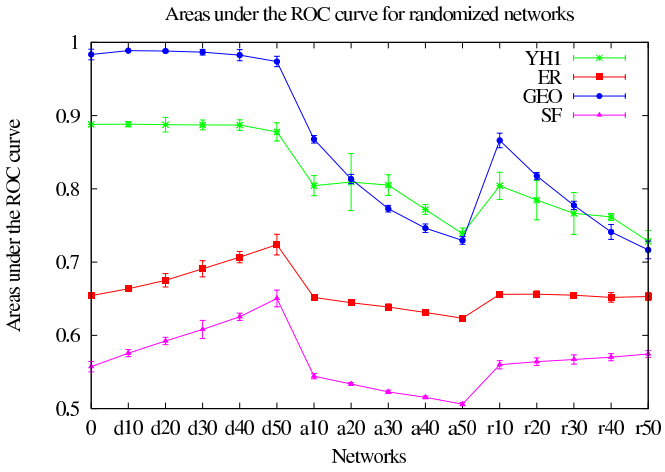


Figure S3: The best-fitting network model out of the three models (ER, GEO, and SF) predicted by the five classifiers (BP, PNN, DT, NBM, and SVM) for the 513 analyzed RIGs.



(a)



(b)

Figure S4: (a) ROC curves illustrating the performance of the embedding algorithm for YH1 PPI network and one network belonging to each of the following model networks that are of the same size as YH1: ER, SF, GEO, and randomized GEO network ("GEO_R10") obtained by randomly rewiring 10% of edges. (b) Areas under the ROC curve (AUCs) for YH1 and its ER, GEO, and SF model networks (denoted by "0" on x-axis), as well as for their randomized versions obtained by randomly deleting, adding, and rewiring (denoted by "d", "a", and "r" on x-axis, respectively) 10%, 20%, 30%, 40%, and 50% of their edges (denoted by "10", "20", "30", "40", and "50" on x-axis, respectively). For each of the network models and each of the randomization schemes, points in the panel represent averages of AUCs over 10 networks. The error bar around a point is one standard deviation below and above the point.

3 Supplementary tables

Classifier	VR-Total	VR-ER	VR-GEO	VR-SF
BP	99.98% (6,575/6,576)	100% (2,192/2,192)	100% (2,192/2,192)	99.96% (2,191/2,192)
PNN	99.97% (6,574/6,576)	100% (2,192/2,192)	100% (2,192/2,192)	99.91% (2,190/2,192)
DT	99.41% (6,537/6,576)	99.41% (2,179/2,192)	99.64% (2,184/2,192)	99.18% (2,174/2,192)
MNB	98.48% (6,476/6,576)	98.18% (2,152/2,192)	100% (2,192/2,192)	97.26% (2,132/2,192)
SVM	94.72% (6,229/6,576)	94.85% (2,079/2,192)	100% (2,192/2,192)	89.33% (1,958/2,192)

Table S1: The validation rates ("VR") for the five classifiers, BP, PNN, DT, MNB, and SVM (column 1), over the entire testing set of 6,576 ER, GEO, and SF networks (column 2), as well as within each individual testing subset of 2,192 ER, 2,192 GEO, or 2,192 SF networks (columns 3–5, respectively).

Acknowledgements

This project was supported by the NSF CAREER IIS-0644424 grant.

References

[1] N. Pržulj, D.G. Corneil, and I. Jurisica. Modeling interactome: Scale-free or geometric? *Bioinformatics*, 20(18):3508–3515, 2004.

[2] P. Erdös and A. Rényi. On random graphs. *Publicationes Mathematicae*, 6:290–297, 1959.

[3] A.L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

[4] M. Penrose. *Geometric Random Graphs*. Oxford University Press, 2003.

[5] D.E. Rumelhart and J. McClelland. *Parallel Data Processing*. The M.I.T. Press, Cambridge, MA, 1986.

[6] P.D. Wasserman. *Advanced Methods in Neural Computing*. Van Nostrand Reinhold, New York, 1993.

[7] J.A. Stegemann and N.R. Buenfeld. A Glossary of Basic Neural Network Terminology for Regression Problems. *Neural Computing and Applications*, 8:290–296, 1999.

[8] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. CRC Press, Boca Raton, FL, 1984.

[9] S. Chakrabarti. *Mining The Web - Discovering Knowledge From Hypertext Data*. Morgan Kaufmann, San Francisco, CA 94104-3205, 2003.

[10] I.H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2nd Edition, San Francisco, 2005.

[11] V.N. Vapnik. *The Nature of Statistical Learning Theory (Information Science and Statistics)*. Springer, 1999.