Pacific Symposium on Biocomputing 14:39-50 (2009)

LEARNING THE STRUCTURE OF PROTEIN-PROTEIN INTERACTION NETWORKS*

OLEKSII KUCHAIEV
Department of computer science,
University of California, Irvine
CA, 92697-3425, USA
E-mail: oleksii.kuchaiev@uci.edu

NATAŠA PRŽULJ†
Department of computer science,
University of California, Irvine
CA, 92697-3425, USA
E-mail: natasha@ics.uci.edu

Modeling and analyzing protein-protein interaction (PPI) networks is an important problem in systems biology. Many random graph models were proposed to capture specific network properties or mimic the way real PPI networks might have evolved. In this paper we introduce a new generative model for PPI networks which is based on geometric random graphs and uses the whole connectivity information of the real PPI networks to learn their structure. Using only the high confidence part of yeast *S. cerevisiae* PPI network for training our new model, we successfully reproduce structural properties of other lower-confidence yeast, as well as of human PPI networks coming from different data sources. Thus, our new approach allows us to utilize high quality parts of currently available PPI data to create accurate models for PPI networks of different species.

1. Introduction

The majority of cellular functions are not carried out by single proteins, but by proteins acting together. Due to the recent advances in the experimental biological techniques such as yeast-2-hybrid[11], tandem affinity[12] purification and other high-throughput methods, there is a huge amount of protein-protein interaction (PPI) data publicly available. This amount

*This project was supported by the NSF CAREER grant IIS-0644424
†Corresponding author

Pacific Symposium on Biocomputing 14:39-50 (2009)

of data requires new mathematical and computational approaches to be developed in order to model and analyze the complex networks that they form. An accurate model of PPI networks will allow better estimating all types of network statistics as well as generating synthetic networks of species for which protein-protein interactions have not been experimentally determined. This could help understand cellular processes and lead future biological experiments. Therefore, analyzing and modeling PPI networks has become a vibrant research area.

A PPI network is a graph with nodes (vertices) corresponding to proteins and edges (links) corresponding to interactions between the proteins. Many random graph models were proposed to model PPI networks. Some of the commonly used models are: Erdös-Rényi random graphs[1], Erdös-Rényi random graphs with the same degree distribution as in data, scale-free graphs[2], geometric random graphs[4,3] and stickiness-index-based models[5]. These models were designed to capture specific network properties such as the degree distribution, the clustering coefficient, the average diameter etc., or to model the way these networks might have evolved (e.g., preferential attachment[6], or gene duplication and mutation[7,15] models); usually, they do not exploit the entire connectivity information from the real PPI networks to learn the model's topological structure. We adopt an alternative approach and exploit the entire connectivity information of a real world PPI network to learn its structure. Thus, we do not force the model to reproduce some specific network properties, such as the degree distribution, the clustering coefficient, the diameter and such. Instead, since our new model learns its structure from the real data, most of these network properties are captured automatically.

2. Methods

Several studies have shown that *geometric random graphs* represent a good model for PPI networks[3,14,9]. A geometric random graph is a graph $G = (V, E)$, where V is a set of nodes distributed uniformly at random in some metric space and for any pair of nodes $x, y \in V$, there is an edge $(x, y) \in E$ if and only if the distance between nodes x and y, $d(x, y)$, in the metric space is less than or equal to some fixed parameter ϵ according to a chosen metric. Our new model is based on geometric random graphs. In our model, henceforth called *trained geometric model*[a], we do not distribute

[a]The Matlab code implementing the model is available upon request.

Pacific Symposium on Biocomputing 14:39-50 (2009)

nodes in a metric space uniformly at random. Instead, we use available real world PPI data of high quality to learn the distribution of the points in a metric space. Having learned this distribution ($p_{learned}$), we generate model networks of arbitrary size by distributing points in the space according to the distribution $p_{learned}$ and connecting two nodes by an edge if they are close enough in space.

Hence, the crucial aspect of this model is the probabilistic distribution $p_{learned}$, which we use to generate model networks. To learn this distribution, we need to have the metric space and an example distribution of points in it that corresponds to real PPI data. Currently, it is hard even to hypothesize about the nature or dimensionality of space in which PPI networks reside. Thus, as a proof of concept, we choose 3-dimensional Euclidean unit cube as our metric space and the Euclidean metric. However, all of the techniques described below could be easily applied to any number of dimensions. Since real PPI networks contain only the connectivity information between the nodes, we need a technique that takes this information and embeds the network nodes into a metric space so that the topological structure of the network is preserved when viewed as a geometric graph constructed from the embedded nodes. That is, the spatial proximity of the nodes will correspond to the PPI network connectivity information as in a geometric random graph. For this purpose, we use the embedding algorithm introduced by Higham et al.[9].

Next, we model the distribution of the points in the space as a Mixture of Gaussians and learn the required parameters using Expectation-Maximization algorithm (EM)[10]. Mixture of Gaussians allows properties of complex distributions to be captured, and it allows easy sampling via the ancestral sampling technique[10]. The details are presented below.

2.1. *Embedding Algorithm*

The embedding algorithm introduced by D. Higham, M. Rašajski and N. Pržulj[9] is based on the ideas from Multi-Dimensional Scaling (MDS)[19]. It takes as an input a PPI network represented as an undirected unweighted simple graph $G = (V, E)$, and a dimensionality m of the Euclidean unit cube to be used for embedding. The algorithm outputs coordinates of each node in the space under the premise that the network connectivity information corresponds to Euclidean proximity, as in geometric random graphs. Conceptually, this algorithm consists of the following steps:

- $\forall$ pairs of nodes $i, j \in V$, calculate the shortest path distances be-

Pacific Symposium on Biocomputing 14:39-50 (2009)

tween them, $d_G(i,j)$. Then choose a cutoff value K and define the matrix of pairwise distances using the formula: $d(i,j) = \sqrt{d_G(i,j)}$ iff $d_G(i,j) \leq K$ otherwise $d(i,j) = K_{max}$, for some K_{max}.

- Construct matrix A using double-centering process:

$$a_{ij} = -\frac{1}{2}(d_{ij}^2 - \frac{1}{n}\sum_{k=1}^{n} d_{ik}^2 - \frac{1}{n}\sum_{k=1}^{n} d_{kj}^2 + \frac{1}{n^2}\sum_{k=1}^{n}\sum_{l=1}^{n} d_{kl}^2)$$

- It can be shown that A has a Schur decomposition and therefore, the resulting embedding into the m-dimensional space is then specified by the eigenvectors of A, corresponding to the m largest eigenvalues. That is, for each protein i, it's coordinates in the m-dimensional Euclidean space are then equal to the i-th coordinates of the eigenvectors $v_1, ..., v_m$ of the matrix A. $(m \leq n)$

2.2. *Learning Network Structure*

To learn the distribution $p_{learned}$ we need to have a learning set $X_{learning}$ of 3-dimensional coordinates of n nodes in the Euclidean unit cube. These coordinates are the output of the embedding algorithm described in section 2.1 applied to the PPI network data that we wish to use for learning. To model the density $p_{learned}$ of the distribution of points in the space, we use a Mixture of Gaussians model[10]:

$$p_{learned}(x) = \sum_{k=1}^{d} \pi_k N(x, \mu_k, \sigma_k^2)$$

It is a linear combination of multi-dimensional (in our case 3 dimensional) Gaussian distributions. Coefficients $\pi_k, k = 1, ..., d$, represent a marginal distribution $p(z)$ ($\sum_{k=1}^{d} \pi_k = 1$ and $\forall k : \pi_k \in [0,1]$) over a latent d-dimensional binary variable z, which has "1-of-d" representation (a particular coordinate z_k is equal to 1 while the rest are equal to 0). This hidden variable indicates what mixture component explains a particular data point x; that is $p(x|z_k = 1) = N(x, \mu_k, \sigma_k^2)$. The parameters to be learned from the data are: π_k, μ_k and σ_k^2 for $k = 1, ..., d$. The number of mixtures d, is a parameter which we do not learn, but choose manually to control the complexity of the model. We use Expectation-Maximization (EM) algorithm to learn the parameters of the Gaussian Mixture model[10].

As in many machine learning problems, more complex models will always explain the learning data better then more simple ones, but too complex models might result in over-fitting and, as a result, will perform badly

Pacific Symposium on Biocomputing 14:39-50 (2009)

in the modeling stage. Thus, we need to choose d big enough to fit the learning data well and small enough to avoid over fitting and other possible issues, such as singularities in the covariance matrices σ_k^2.

For these purposes we use Bayesian Information Criterion (BIC). During our experiments, we found that for our particular case, standard BIC does not penalize the model for its complexity strongly enough; therefore, we increase the penalty for having more parameters two times and use the formula: $BIC = log(p(X|\theta)) - MlogN$. Here, X is our learning set, $log(p(X|\theta))$ is a log-likelihood function, θ is the parameters we learn (μ, σ^2 and π), M is number of free parameters in the model (size of θ), and N is a size of the learning set. By choosing different numbers of mixtures for our model (choosing different values of d) we vary the number of free parameters M of the model, which in our case equals $M = m^2+m+(d-1)*(m+m^2+1)$, where m is the dimensionality of the space we used for the embedding (m=3 in all our experiments). According to this criterion, we need to choose the model which has the highest value of BIC[10].

2.3. *Generating Model Networks*

After the model has learned the density function $p_{learned}(x)$, it could be used for generating model networks of arbitrary size. Suppose that we want to generate a network $Net = (V, E)$ with $|V| = n$ and $|E| = m$. We use an *ancestral sampling* technique to sample from distribution $p_{learned}(x) = p(x|z) * p(z)$, where marginal distribution $p(z)$ is given by the coefficients π_k, the conditional distribution of x given z is $p(x|z) = N(x, \mu_k, \sigma_k^2)$ and to sample n nodes from $p_{learned}(x)$ we repeat the following two consecutive steps n times: 1) sample the value of z from $(\pi_1, ..., \pi_k)$, 2) sample 3-dimensional coordinates of point x_i from $p(x|z_k = 1) = N(x, \mu_k, \sigma_k^2)$.

After this sampling procedure, we have n points distributed in 3-dimensional Euclidean unit cube according to the density $p_{learned}(x)$. Next, we adjust the value of parameter ϵ to obtain m pairs of nodes with distances between them not greater then ϵ. Then, we connect these m pairs of the nodes by edges and obtain a model network of the required size.

2.4. *Network Comparisons*

We compare the following global network properties of our model and the PPI data networks: the degree distribution, the average clustering coefficient and the diameter. However, these properties themselves do not tell us much about the structure of a network. For example, it is trivial to

Pacific Symposium on Biocomputing 14:39-50 (2009)

construct two networks with the same degree distribution but with vastly different local structure. The same is also true for the clustering coefficient and the average diameter. An additional issue with using these network properties for evaluating the performance of network models is the noise and incompleteness present in PPI networks. For example, if we take two networks with the same structure, by removing edges from them we will decrease the degrees of the nodes, the clustering coefficients and the average diameters of both networks and therefore, while some structure between the two networks might have remained similar, all three of the above properties might differ substantially.

To tackle these issues we use the *Graphlet Degree Distribution (GDD) Agreement* measure[14,13] introduced by Pržulj[14] to provide a detailed evaluation of structural similarities of large networks. We give a few definitions to explain this similarity measure. *A graphlet* is a small induced subgraph of a network[3]. There are 30 possible non-isomorphic graphlets on 2, 3, 4 and 5 nodes. To calculate GDD agreement between two networks, we need to calculate for each node in the network how many times it touches each of the 30 graphlets. From a topological point of view, it is relevant to distinguish between *automorphism orbits* of each graphlet. For example, in a 3-node path, the "end-nodes" are identical from the topological point of view (i.e., can be mapped to each other by an *automorphism*, an *isomorphism* of a graph with itself – see[14] for details), whereas the "middle node" is different; therefore, a 3-node path has two different automorphism orbits. There are 73 automorphism orbits for the 30 graphlets on 2 to 5 nodes. *A Graphlet Degree Distribution (GDD)* is a 73-component distribution of the automorphism orbits in a network. Its j^{th} component, $d^j(k)$, is the sample distribution of the number of nodes in the network touching a particular graphlet (at automorphism orbit j) k times. Graphlet degree distribution has the degree distribution as its 1^{st} component, which corresponds to the only automorphism orbit of a 2-node path (edge). The *GDD Agreement* is a similarity measure between graphlet degree distributions of two networks. It is a number between 0 and 1, meaning that two networks have similar GDDs if their GDD agreement is close to 1, and otherwise, their GDDs are different. It is important to note that graphlet degree distribution measures the local structure of a network, because it is based on small local neighborhoods of the nodes. It is especially a very strong measure of structural similarities of small-world networks (those with small diameters), since in these networks, 5-node graphlets reach all parts of the network. Since PPI networks have small diameters, GDD is good for measuring their structure.

Pacific Symposium on Biocomputing 14:39-50 (2009)

We use GraphCrunch software package[13] to calculate GDD agreement and other network properties that evaluate the fit of model networks to the data.

3. Application

3.1. *Datasets and Learning*

We apply the above described approach to model PPI networks of yeast and human. These two particular species have been chosen because compared to other eukaryotic species, they have the most complete and accurate PPI network data available. For our learning set (henceforth $S_{learning}$), we use the largest connected component of the high confidence yeast network described in the study by Collins et al[16]. This part of the yeast *S. cerevisiae* PPI network is believed to be of quality comparable to the data produced by small-scale experiments. It has $1,004$ nodes and $8,323$ edges.

Table 1 presents real-world PPI networks that we modeled using our trained geometric model. All these networks contain only physical interactions between proteins. Using the notation from Table 1, note that $S_{learning}$ $\subset$ YCH $\subset$ YCOL; that is, YCH is a high confidence part of the YCOL and $S_{learning}$ is the largest connected component of the YCH PPI network.

Table 1. Real-world PPI networks used to evaluate the performance of the trained geometric model.

Network	Nodes	Edges	Origin
YCH	1622	9074	high confidence yeast PPI network from Collins et al.[16]
YCOL	2390	16127	yeast PPI network from Collins et al.[16]
YBG	4716	32747	yeast PPI network from BIOGRID[17]
HBG	7930	23543	human PPI network from BIOGRID[17]
HRAD	9141	42456	human PPI network from Radivojac et al.[18]

We embed $S_{learning}$ into a 3-dimensional cube using the embedding algorithm described above. After that, we have 3-dimensional coordinates of the $1,004$ nodes of $S_{learning}$ PPI network; we present these coordinates in $X_{learning}$, a 3×1004 matrix, and use $X_{learning}$ to learn the distribution $p_{learned}(x)$. We got the highest value of the BIC for 7 mixture components in the model and therefore we put $d = 7$ in the distribution $p_{learned}(x)$.

Pacific Symposium on Biocomputing 14:39-50 (2009)

3.2. *Analysis*

One of the most important network properties is the degree distribution. We compare the degree distributions of two networks using *Pearson correlation coefficient*. This coefficient ranges from -1 to 1: if it is close to 1 or -1, then there is a perfect linear correlation between two distributions (where nodes are ordered according to their degrees), and if it is close to 0, then there is no linear correlation. We measure this, not the difference such as sum of squared errors, because the latter is strongly affected by the amount of noise in PPI network data, while the correlation between distributions should be preserved even when some edges are missing (false negatives) or are mistakingly present (false positives). Figure 1 presents values of Pearson correlation coefficients of degree distributions between real PPI and model networks.

Table 2. Models used to model PPI networks.

Network	Model
ER	Erdös-Rényi random graph model[1]
ER_DD	ER model with the same degree distribution as in data
GEO	Geometric random graph model[3]
SF	Scale-free Barabasi-Albert preferential attachment model[2]
STICKY	Stickiness-index-based model[5]
TGEO	The new trained geometric model

Note that by construction, ER_DD model networks have exactly the same degree distribution as real PPI networks and therefore, their Pearson correlation coefficients with the PPI networks always equal to 1. It has been observed that PPI networks have power-law degree distributions[8], which gave rise to popularity of scale-free network models. However, as we can see in Figure 1, SF network model badly captures the degree distribution of the PPI network data compared to our new model, which has learned and captured the degree distribution from the part of real-world PPI network very well. As we can see in Figure 1, the new model significantly outperforms any other model in all cases. Note that we use only high quality part of yeast PPI network for learning, but this allowed us to generate model networks of high quality also for human PPI networks.

As illustrated in Figures 2 and 3, the trained geometric (TGEO) model is better fitting in all cases, except one, than the standard geometric random graph model (GEO). For BIOGRID[17] and Radivojac et al. data sets[18],

Pacific Symposium on Biocomputing 14:39-50 (2009)

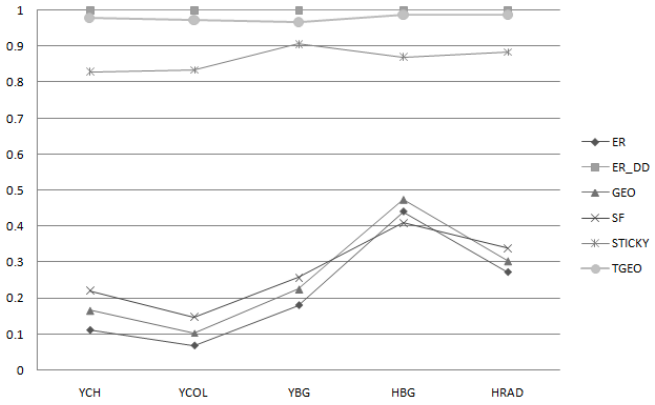

Figure 1. Pearson correlation coefficients of degree distributions between real PPI and model networks. Horizontal axis corresponds to the real PPI networks described in Table 1 and different labels correspond to different model networks described in Table 2.

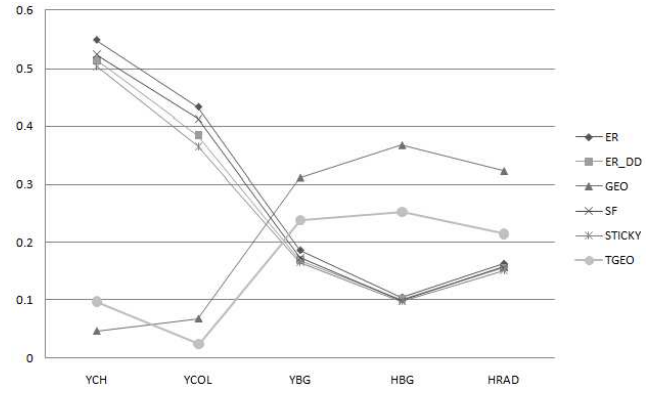

Figure 2. Differences in clustering coefficients of data and model networks. Horizontal axis corresponds to the real PPI networks described in Table 1 and different labels corresponds to different model networks described in Table 2.

GEO and TGEO models perform worse then other models. As described in section 2.4, this is likely caused by incompleteness and noise in the data. By incompleteness we do not mean the number of nodes (proteins) in the network, but presence or absence of real interactions among given nodes. Obviously, real yeast PPI network has some fixed diameter and clustering coefficient, which we can not measure now due to high noise levels in the data. Yeast networks coming from different sources (such as Collins et al.[16]

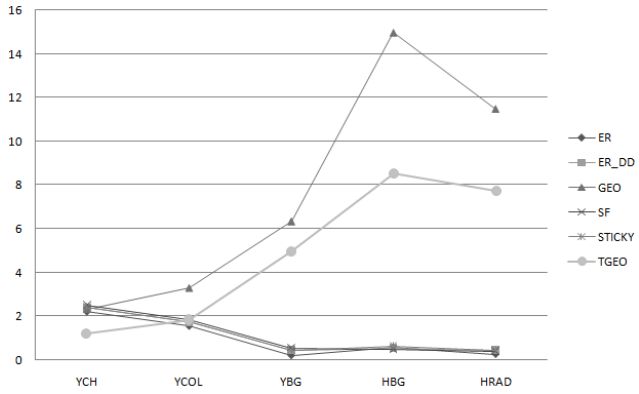

Figure 3. Differences in diameters of data and model networks. Horizontal axis corresponds to the real PPI networks described in Table 1 and different labels correspond to different model networks described in Table 2.

and BIOGRID[17]) have substantially different clustering coefficients and diameters (YCOL has the clustering coefficient of 0.44 and the diameter of 4.81, whereas YBG has the clustering coefficient of 0.19 and the diameter of 3.71). Thus, these two characteristics at the moment can not tell us much about the real structure of a PPI network.

Although global network properties are important, they might not capture the local structure of networks and are are also very sensitive to high levels of noise. Therefore, we are more interested in a more structurally constraining measure of network similarity, the GDD agreement[14]. Figure 4 presents GDD agreements between real PPI and model networks. It shows that our new trained geometric model (TGEO) fits all but one PPI network better than other models; the exception is yeast PPI network taken from BIOGRID[17], for which TGEO performs almost the same as STICKY model, but it captures the degree distribution much better then STICKY model (see Figure 1). Surprisingly, learning network structure from yeast PPI data gives better modeling insights into modeling human PPI networks than do other network models. This suggests that PPI networks of even such distant organisms as yeast and human exhibit similarities in their structures.

4. Discussion and Conclusions

Many network models have previously been introduced attempting to capture specific sets of PPI network properties, or to mimic the way in which

Pacific Symposium on Biocomputing 14:39-50 (2009)

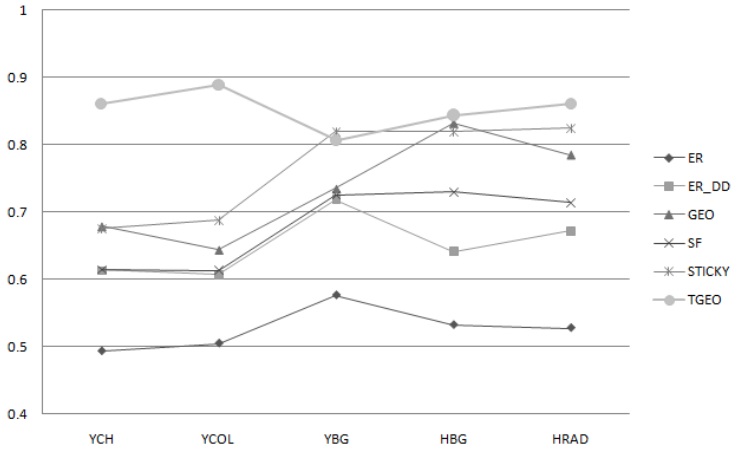

Figure 4. GDD-agreement between the data and model networks. Horizontal axis corresponds to the real PPI networks described in Table 1 and different labels correspond to different model networks described in Table 2.

these networks might have evolved. However, it is difficult to say to what extent network properties really describe network structure. Therefore, network models should use all of the available real network information, i.e., the entire network connectivity information, not only some of the network properties, to learn the structure of PPI networks. Using our new network embedding algorithm[9] and our previous observations that geometric random graphs provide a good model for PPI networks[3,14,9], we have reduced the problem of constructing a well-fitting model for PPI networks to a standard machine learning problem of density estimation. Our new *trained geometric model* uses parts of PPI networks of high quality to learn the network structure and uses this learned knowledge to generate model networks with arbitrary numbers of nodes and edges. Our experiments show that even if we use a learning PPI network from a simple eukaryotic organism (i.e., yeast) and then use our model trained in this way to model a PPI network of another higher eukaryotic organism (i.e., human), we get a substantial improvement in the fit of our new network model to PPI networks over all other currently commonly used random graph models. This suggest that PPI networks of even such distant species as yeast and human exhibit similar structural properties.

Currently, noise is one of the biggest challenges in analyzing and modeling PPI networks. We intentionally used only high confidence data from the study by Collins et al.[16] to learn real structural properties of PPI networks

Pacific Symposium on Biocomputing 14:39-50 (2009)

and thus to avoid learning the properties of the noise in PPI networks. Once the data of such quality becomes available for human and other species, it should be used to train species-specific models.

Our new network modeling approach can be applied not only to model PPI networks, but also to model other real-world networks. Since the distribution of points in a metric space is crucial for properties of geometric graphs, by learning this distribution from real-world networks we can generate model networks with different structural properties and in that way model networks arising in different applications.

References

1. Erdös,P. and Rényi,A., *Publ. Math.* **6**, 290-297, (1956).
2. Barabasi,A.L. and Albert,R., *Science* **286**, 509-512, (1999).
3. N. Pržulj, D. G. Corneil, and I. Jurisica, *Bioinformatics* **20**, 3508-3515, (2004).
4. M. Penrose. Geometric Random Graphs, *Oxford University Press*, (2003).
5. N. Pržulj and Des Higham, *Journal of the Royal Society Interface*, **3**, 10 (2006).
6. R. Albert and A. Barabási, *Reviews of Modern Physics*, **74**, 47-97 (2002).
7. A. Vaźquez, A. Flamminia, A Maritana, A. Vespignani, *Complexus*, **1**, 38-44 (2003).
8. Barabási AL, Oltvai ZN, *Nat Rev Genet.*, **2**, 5, (2004).
9. D. J. Higham, M. Rašajski, and N. Pržulj, *Bioinformatics*, **24**, 8 (2008).
10. C. Bishop, Pattern Recognition and Machine Learning, *Springler*, (2006).
11. T. Ito, K. Tashiro, S. Muta, R. Ozawa, T. Chiba, M. Nishizawa, K. Yamamoto, S. Kuhara, and Y. Sakaki, *Proc Natl Acad Sci U S A*, **97**, 3 (2000).
12. Gavin, A. C., Bosche, M., Krause, R., Grandi, P., Marzioch, M., Bauer, A., Schultz, J., Rick, J. M., Michon, A. M., Cruciat, C. M., Remor, M., Hofert, C., Schelder, M., Brajenovic, M., Ruffner, H., Merino, A., Klein, K., Hudak, M., Dickson, D., Rudi, T., Gnau, V., Bauch, A., Bastuck, S., Huhse, B., Leutwein, C., Heurtier, M. A., Copley, R. R., Edelmann, A., Querfurth, E., Rybin, V., Drewes, G., Raida, M., Bouwmeester, T., Bork, P., Seraphin, B., Kuster, B., Neubauer, G., and Superti-Furga, G. *Nature* **415** 6868 (2002).
13. T. Milenković, J. Lai, and N. Pržulj, *BMC Bioinformatics*, **9**, 70, (2008).
14. N. Pržulj, *Proceedings of the 2006 European Conference on Computational Biology, ECCB '06*, **23**, e177-e183, (2007).
15. N. Pržulj, *Proceedings of the 2007 IEEE Information Theory Workshop (ITW)* (2007).
16. Collins SR, Kemmeren P, Zhao XC, Greenblatt JF, Spencer F, Holstege FC, Weissman JS, Krogan NJ, *Mol Cell Proteomics.*, **3**, 6, (2007).
17. Stark, C. and Breitkreutz, B.J. and Reguly, T. and Boucher, L. and Breitkreutz, A. and Tyers, M., *Nucleic Acids Research*, **34**, D535-D539, (2006).
18. Radivojac, P., Peng, K., Clark, W.T., Peters, B.J., Mohan, A., Boyle, S.M. and Mooney, S.D. *Proteins*, **3**, 72, (2008).
19. Cox T. F. and Cox M. A. A. Multidimensional Scaling. *Chapman and Hall*, London. (1994).