PROTEOME SCIENCE

PROCEEDINGS Open Access

Protein complex detection with semi-supervised learning in protein interaction networks

Lei Shi[1*], Xiujuan Lei[2], Aidong Zhang[1]

From International Workshop on Computational Proteomics
Hong Kong, China. 18-21 December 2010

Abstract

Background: Protein-protein interactions (PPIs) play fundamental roles in nearly all biological processes. The systematic analysis of PPI networks can enable a great understanding of cellular organization, processes and function. In this paper, we investigate the problem of protein complex detection from noisy protein interaction data, i.e., finding the subsets of proteins that are closely coupled via protein interactions. However, protein complexes are likely to overlap and the interaction data are very noisy. It is a great challenge to effectively analyze the massive data for biologically meaningful protein complex detection.

Results: Many people try to solve the problem by using the traditional unsupervised graph clustering methods. Here, we stand from a different point of view, redefining the properties and features for protein complexes and designing a "semi-supervised" method to analyze the problem. In this paper, we utilize the neural network with the "semi-supervised" mechanism to detect the protein complexes. By retraining the neural network model recursively, we could find the optimized parameters for the model, in such a way we can successfully detect the protein complexes. The comparison results show that our algorithm could identify protein complexes that are missed by other methods. We also have shown that our method achieve better precision and recall rates for the identified protein complexes than other existing methods. In addition, the framework we proposed is easy to be extended in the future.

Conclusions: Using a weighted network to represent the protein interaction network is more appropriate than using a traditional unweighted network. In addition, integrating biological features and topological features to represent protein complexes is more meaningful than using dense subgraphs. Last, the "semi-supervised" learning model is a promising model to detect protein complexes with more biological and topological features available.

Background

High-throughput assay methodologies, such as microarrays and mass spectrometry, have resulted in the rapid growth of protein data sets, the analysis of which can potentially yield insights into the mechanisms of human diseases and the discovery of new therapeutic interventions [1][2]. Systematic analysis of the underlying relationships in these protein data sets can potentially provide useful insights into roles of proteins in biological processes [3][4][5][6].

PPI data sets provide us the good opportunity to systematically analyze the structure of a large living system and also allow us to use it to understand essential principles like essentiality, genetic interactions, functions, functional modules, protein complexes and cellular pathways [7]. Cellular functions and biochemical events are coordinately carried out by groups of proteins interacting with each other in functional modules, and the modular structure of complex networks is critical to functions [8]. Identifying such protein complexes in PPI networks is very important for understanding the structure and function of these fundamental cellular networks. Therefore, developing an effective computational

* Correspondence: lshi2@buffalo.edu
[1]Computer Science & Engineering Department, State University of New York at Buffalo, Buffalo, NY, USA
Full list of author information is available at the end of the article

approach to identify those protein complexes should be highly challenging but indispensable.

However, protein complexes are likely to overlap and the interaction data are very noisy. It is a great challenge to effectively analyze the massive data for biologically meaningful protein complex detection. Since most proteins form macromolecular complexes involving two or more proteins to perform biological functions, many people assume protein complexes should be dense subgraphs. Thus some graph clustering based algorithms could be applied to it. Molecular Complex Detection (MCODE) [9] is the first computational method to detect protein complexes from PPI networks. MCODE first identifies densely connected subgraphs and then uses another post-processing to filter non-dense subgraphs and generate overlapping clusters. Later, Spirin and Mirny [10] proposed a clique based algorithm, which exhaustively searches all the full cliques as protein complexes in the network. Since using clique is too constrained, they modified it by applying the Super-Paramagnetic Clustering (SPC) and a Monte Carlo (MC) simulation for the same purpose. Instead of adopting the over-constraining full cliques as the basis for protein complexes, Li *et al.*[*11*] devised an LCMA algorithm (Local Clique Merge Algorithm) that adopts a local clique merging method as an attempt to address the current incompleteness limitation of protein interaction data. Amin *et al.*[*12*] proposed a cluster periphery-tacking algorithm (DPCLus) to detect protein complexes by keeping track of the periphery of a detected cluster. Chua *et al.*[*13*] proposed an algorithm called PCP (ProteinComplexPrediction) for complex prediction, which utilized the filtered PPI network by FS-weight [14], clique finding and merging techniques. Ucar *et al.*[*15*] developed a refinement method, which uses hub protein duplication strategy to detect dense subgraphs in scale-free PPI networks with multi-functional hub proteins assigned to multiple clusters. Adamcsek *et al.*[*16*] proposed a CFinder algorithm to find complexes in the PPI networks. CFinder detects k-cliques as modules and then merges modules by calculating their similarities. Mete [17] extended the density-based clustering method DBSCAN [18] and used it in the PPI networks. SCAN first forms a cluster by a core node then iteratively merges the neighboring nodes one by one. Finally, the detected clusters are formed to become the predicted protein complexes.

The previous methods are suffering from a serious problem, that is, they all assume protein complexes as dense subgraphs. As Qi *et al.*[*19*] pointed out, not all protein complexes are clique-oriented and there are quite a large amount of protein complexes with shapes like star-shape or other forms. In this paper, we will solve the problem from another perspective, redefining the properties and features for protein complexes and using a semi-supervised learning method to build a model to detect those hidden protein complexes in the scale-free PPI networks. First, we choose several biological and topological features to represent the protein complexes. Then, we use the “semi-supervised” mechanism to recursively train the neural network and obtain the optimized parameters for the model. Last, we use the neural network to detect the protein complexes in the protein interaction network.

The paper is organized as follows. First, we identify the difficulties of the problem. Second, we propose some favorable properties for protein complexes. Third, we propose the multi-layer neural network. Fourth, we conduct extensive experiments to verify the effectiveness of the proposed method. Finally, we conclude the paper and propose the future work.

Challenges in protein complex detection

Through extensive observations, we found the following problems are the keys to detect protein complexes in the PPI networks.

- Protein interaction data are very noisy. Since a clustering method is based on the protein protein interactions in the graph, more reliable those interactions are, more accurate the clustering result will be. From the previous works [14][20][21], using a weighted and filtered graph instead of traditional unweighted graph to represent a PPI network is proven to be an effective way. Then the problem becomes how to obtain the reliable protein protein interactions in PPI data. Here we are using GO (Gene Ontology) to obtain the similarity between different proteins in the network and build a weighted graph with a setup threshold.
- Proteins may participate in multiple protein complexes. Therefore, protein complexes may overlap with each other. These overlaps correspond to proteins’ participation in multiple pathways and the crosstalk between different biological modules. Thus, the traditional paradigm for clustering and putting each protein into one single cluster doesn’t suit our problem well. Instead, we would prefer a method that can detect subgraphs with possible overlaps. Our proposed semi-supervised method overcomes this drawback that many existing graph clustering methods suffered and gives a promising result.
- How to represent protein complexes. Most existing clustering methods assume protein complexes as dense subgraphs, which is not always true for the protein complexes in the PPI networks [19]. In addition, all kinds of topologies present in protein complexes, and tremendous variation of the sizes of protein complexes pose a further problem for identifying the specific topologies. Traditional methods were all non-supervised methods

which didn't fully utilize the properties and features of protein complexes. Here we are trying to use both topological properties and biological properties of protein complexes to represent protein complexes and propose a multi-layer neural network based semi-supervised method to detect the hidden protein complexes.

Results and discussion

Data preparation

For our experiments, we built our weighted protein interaction networks from DIP data set [22], which contains 4935 proteins and 14162 interactions. The way to build the weighted network is illustrated in our previous paper [23][24]. In order to evaluate the predicted complexes, the set of real complexes are selected as the benchmarks. This benchmark set is from MIPS [25] and we only select those complexes that contain more than two proteins.

Validation criterion

In order to study the relative performance of different supervised learning algorithms, we need to define an evaluation criterion that determines if a predicted protein complex matches a complex in benchmark set. In [9], the authors used an overlapping score as the criterion:

$$overlap(A,B) = \frac{|V_A \cap V_B|}{|V_A| \cdot |V_B|}, \tag{1}$$

where A is the predicted complexes, B is the true protein complexes, V_A is the set of proteins in the subgraph A, and V_B is the set of proteins in the subgraph B. In this paper, we use an overlapping threshold of 0.20 to determine a match for all experiments. Predicted protein clusters that match one or more true protein complexes with overlapping scores higher than this threshold are identified as "matched clusters," and the corresponding true complexes are noted as "matched complexes."

To measure the accuracies of prediction, we calculate prediction, recall and *f*-measure for different algorithms. They are defined as:

$$Precision = \frac{M_{cluster}}{P_{cluster}}, \tag{2}$$

where $M_{cluster}$ is the number of the matched clusters and $P_{cluster}$ is the number of the predicted clusters, and

$$Recall = \frac{M_{complexes}}{T_{complexes}}, \tag{3}$$

where $M_{complexes}$ is the number of matched complexes and $T_{complexes}$ is the number of the true complexes. Because smaller size complexes and clusters have high probabilities of occurring by chance and they are not meaningful, here we only consider clusters and complexes whose size is 4 or larger.

$$f = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}, \tag{4}$$

where *f*-measure is defined as the harmonic mean of recall and precision. It reflects a combination of precision and recall.

Comparative evaluation

To evaluate the performance of our proposed method, we compared two different supervised learning methods SVM and Bayesian Network (BN) [19] with our method. In addition, we also compared it to a representative non-supervised learning method "MCODE" [9]. "MCODE" is a density based clustering method which is used to find highly interconnected subgraphs in the PPI networks. SVM [26] and Bayesian Network [19] use the same features as our method, showing the effectiveness of the neural network model. The comparison result is presented in Table 1 and in Figure 1. Each method is evaluated by the precision, recall and *f*-measure, separately. As we can see, the proposed method is superior in all measures. The recall rate of our method is around 49%, which is impressive. Our precision is not as high as recall but it is still better than the other existing methods. In addition, our *f*-measure is the highest among all 4 methods. Since many protein complexes are not included in the benchmark set, the predicted protein complexes could be correct protein complexes that are not in the available data. The recall value of MCODE is relatively low compared with other methods. This is because the protein complexes that MCODE found are relatively larger than the clusters found by the other approaches and thus affect the result [19]. In addition, the performances of SVM and Bayesian model are not as good as neural networks; perhaps this is due to the unique way that neural networks train the parameters of the model and the high tolerance to the noisy data.

Table 1 Performance comparison of MCODE (Molecular Complex Detection), NN (Neural Network), SVM and BN (Bayesian Network).

Method	Supervised or Not	Precision	Recall	*f*-measure
MCODE	no	0.315	0.078	0.125
NN	yes	0.333	0.491	0.397
SVM	yes	0.239	0.412	0.302
BN	yes	0.273	0.473	0.346

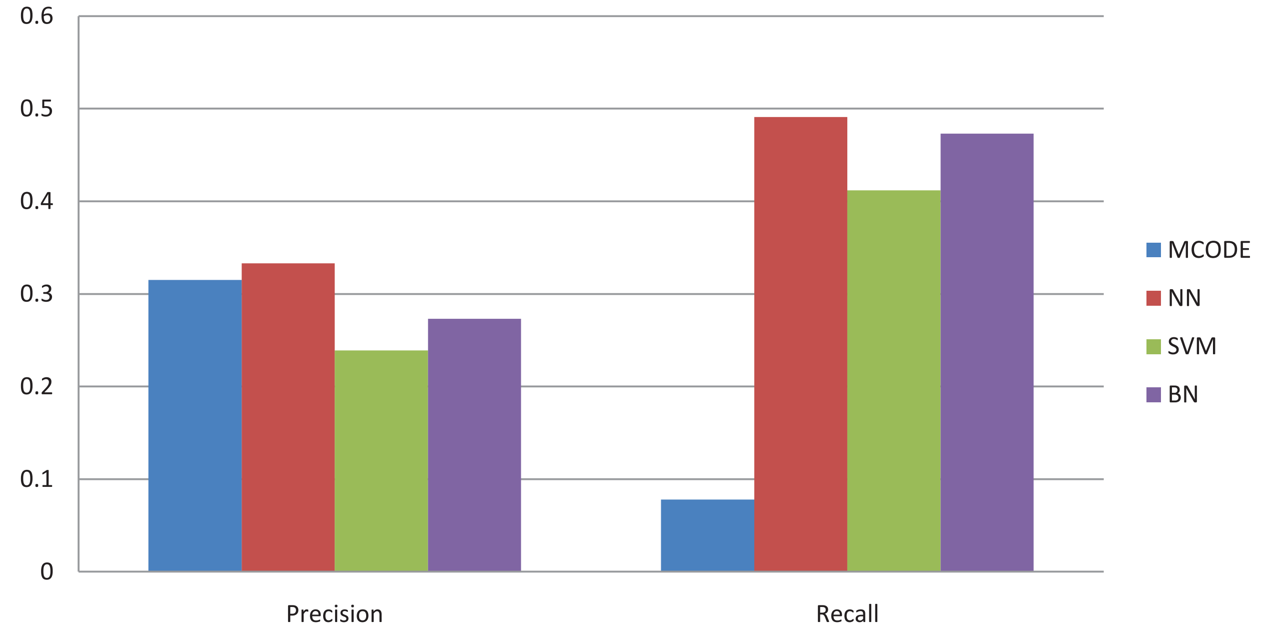

Figure 1 Comparative performance of several approaches in terms of precision and recall for DIP data.

Conclusions

In this paper, we analyzed and detected protein complexes in protein-protein interaction networks from a different perspective. Instead of using traditional non-supervised algorithms to find dense subgraphs in the PPI networks, we proposed a semi-supervised prediction model with neural network. Unlike previous methods that relied too much on the density of the subgraph, our algorithm utilizes topological and biological features from known protein complexes. With those characterized features, we could represent protein complexes better than the previous methods. Thus a more accurate prediction model can be built upon them. The comparison results show that our algorithm could identify complexes that are missed by other methods. We also have shown that our method achieves better precision and recall rates for the identified protein complexes. In addition, the framework we proposed is easy to be extended in the future. Since obtaining the features of protein complexes and building the prediction model are independent, we could add more representative features of protein complexes in the future work and adopt other similar prediction models that are similar to neural network. In the next step, we hope to find more representative features to formulate protein complexes either from topological manner or biological manner. Also, with more PPI networks of different species becoming available, we could apply the proposed method to the new emerging data sets.

Methods

While the existing methods identify protein complexes with strong assumptions about their topology (dense subgraph), our proposed method utilizes multiple features that define protein complexes in protein-protein interaction networks. Instead of only assuming the protein complexes as dense subgraphs, we derive several properties from known protein complexes and use these features to search for the new protein complex. Our algorithm first gains the weights for different features from the limited known protein complexes. Then it will assign a score to any subgraph in the graph. With a setup threshold, we could label some of the subgraphs as complexes. With more complexes, we could train the data again and get more suitable weights for the features, thus better prediction model. Recursively, we will find all protein complexes in the PPI network. Compared with the existing method, our proposed model found more accurate protein complexes in the protein-protein interaction network.

Weighted undirected PPI network

Many previous works [20] have already pointed out that the PPI networks are very noisy. Table 2 shows the percentage of function-relevant interactions in three proteinprotein interaction data sets, namely, DIP [22], MIPS [25] and BioGrid [27]. An interaction is considered to be function-relevant if the two proteins involved in the interaction have at least one function in common.

Table 2 The percentage of function-relevant interactions in three protein interaction data sets

Data Set	Total number of interactions	Number of functional-relevant interactions	Percentage
DIP	14162	5216	36.83%
MIPS	13877	4189	30.18%
BioGrids	117675	36446	30.97 %

In this test, we adopt FunCat(version 20070316) [25] in the MIPS database as our annotation categories. From Table 2, we can see that only 30% – 40% observed interactions are relevant in functions. In other words, most of the observed interactions do not share functions. Among those sharing function pairs, some of them share more functions than the others. So there are a lot of false-positive and false-negative interactions in the PPI network and we would better use a weighted graph to represent it rather than the unweighted graph. Throughout the paper, we use a weighted, undirected graph $G = (W, V, E)$ to represent the protein interaction network where V represents the set of vertices (proteins), E represents the set of edges (interactions) and W represents the likelihoods for the interactions between vertices. The weights are obtained by using Gene Ontology (GO), which is used for consistent description of genes and gene products. The GO provides a collection of well defined and structured biological terms called GO terms, which is shown in Figure 2. By using GO structure, we could calculate the semantic similarity of different proteins in protein-protein interaction networks, therefore we use them as the weight of the edges. In the previous work [21], we have already successfully obtained a weighted, undirected DIP protein interaction network. Here we use the same network as before.

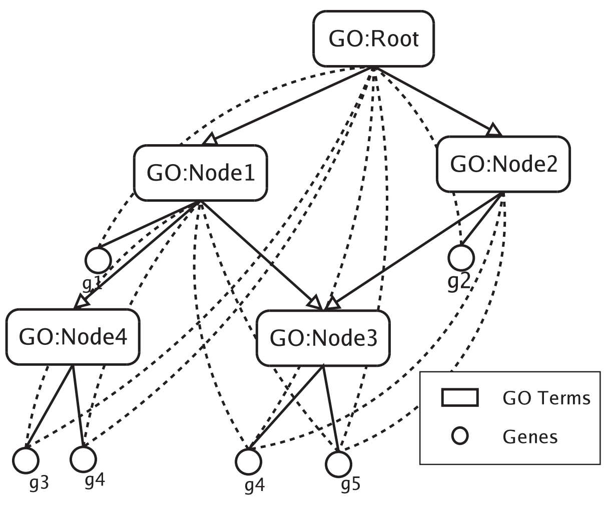

Figure 2 Gene Ontology structure

Complex features

Choosing the right features to representing the protein complexes is the key issue to our problem. So far, there is a lot of work that has been done in this area. In general, the existing approaches either use properties of nodes and edges or utilize non-trivial substructures. In this paper, we use both topological features of subgraphs and biological features of the proteins in the subgraphs. Most of the features are selected from the prior works on subgraph analysis [19][28][29]. The features that we used are listed in Table 3 and each of the feature types will be briefly discussed in the following.

Topological features:

1. Graph density: graph density is defined as $D = \frac{2|E|}{|V|(|V|-1)}$, where $|E|$ is the number of edges in the graph and $|V|$ is the number of vertices in the graph.

2. Degree statistics: these features are calculated from the degree of vertexes in the subgraph. A degree is defined as the number of neighbors of a vertex. Mean degree, variance of degrees, median degree and maximum degree are chosen for degree statistics.

3. Edge weight statistics: we only consider edges with nonzero weights here. Like degree statistics, mean and variance of weights are taken as features.

4. Topological change [28]: This group of features is gained by measuring the topological changes when different cutoffs of the weights are applied to the graph. Topological changes are measures as $T_i = (|E_i| - |E_{i+1}|)/|E_i|$, where E_i is the number of edges with different cutoffs i.

5. Clustering coefficient: the clustering coefficient is a measure of degree to which nodes in a graph tend to

Table 3 The distribution of features.

group ID	group name	number of features
1	Graph density	1
2	Degree statistics	4
3	Edge weight statistics	2
4	Topological change	7
5	Clustering coefficient	3
6	Topological coefficient	3
7	Protein length	2
8	Polarity of amino acids	21

cluster together. It is defined as $C_v = \dfrac{2\left|e(i,j)\right|}{d(v)(d(v)-1)} : i, j \in N(v), e(i,j) \in E$, where $|e(i, j)|$ gives the number of triangles that go through node v, whereas $d(v)(d(v) - 1)/2$ is the total number of triangles that could pass through node v.

6. Topological coefficient [19]: the topological coefficient is a relative measure of the extent to which a protein shares interaction partners with other proteins.

Biological features:

1. Protein length: the number of amino acids in a protein sequence.

2. Polarity of amino acids: the longer and more complementary the binding sites, the majority of which would be polar, of the protein, the stronger the proteins would be bound.

A two layers feed-forward neural network based model

A neural network is a set of connected input/output units in which each connection has a weight associated with it. In the last couple of years, many different kinds of neural networks and corresponding algorithms have been developed. Among the existing algorithms, the **backpropagation** algorithm [30] is the most well known one. The backpropagation algorithm performs learning on a multilayer feed-forward neural network. It iteratively learns a set of weights for prediction of the class label of tuples. Normally a multilayer feed-forward neural network consists of an input layer, multiple hidden layers, and an output layer. To avoid the long learning process, we choose to use the two layers model which contains one input layer, two hidden layers, and one output layer. The PPI network is notorious for its noisy behavior, which contains many false negative connections and false positive connections. Since the neural network is famous for its high tolerance of noisy data, it is an ideal prediction model for our problem [31]. An example of multiple-layer neural network is illustrated in Figure 3. In this example, there are one input layer, two hidden layers and one output layer in the network. $\mathbf{x}(x_1, x_2, x_3)$ is the input pattern and $\mathbf{y}(y_1, y_2)$ is the teaching or target vector. During the network training, the signals generated by the output layers are compared with the target vector $\mathbf{y}$, and any difference is used in training the weights throughout the network. Some key parameters in the neural network are set as follows. The number of units in the input layer is the number of features that we use (in our case, it is 43). The number of output layers is one in our case, since we only need to know if a subgraph is complex or not, which is a classic bi-class classification. The number of hidden layers is two, and we set 11 units for the first layer and 7 units for the second layer. Those numbers are obtained by experiments starting from a full connected network with a sufficiently large number of nodes to a smaller number of nodes. The activation function for each unit we use is the logistic function. Given the net input I_j to unit j, then O_j, the output of unit j, is computed as:

$$O_j = \frac{1}{1+e^{-I_j}}. \tag{5}$$

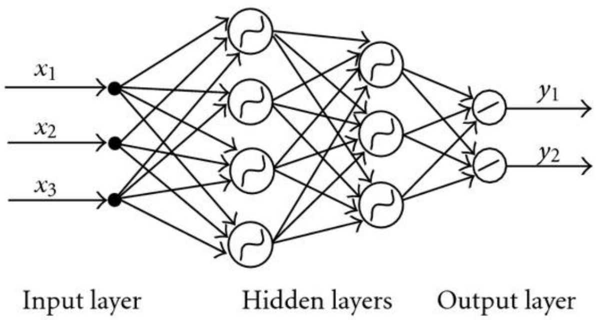

Figure 3 A multiple-layer neural network example

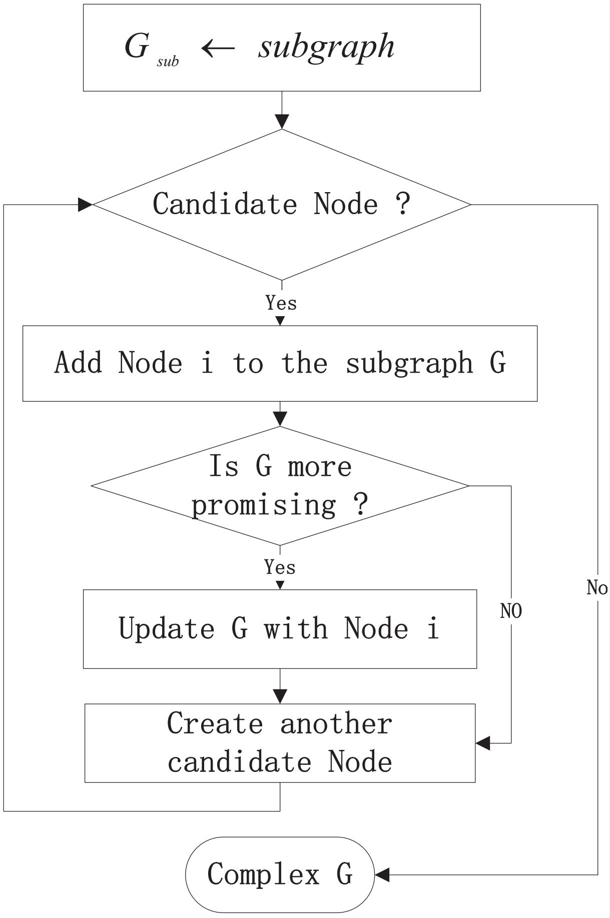
$G_{sub} \leftarrow subgraph$
Candidate Node ?
Yes
Add Node i to the subgraph G
Is G more promising ?
Yes
Update G with Node i
NO
No
Create another candidate Node
Complex G

Figure 4 Flow chart of the subgraph generation subroutine

Thus, the error of a hidden layer unit *j* is

$$\mathrm{Err}_j = O_j(1 - O_j)\sum_k \mathrm{Err}_k w_{jk}, \quad (6)$$

where w_{jk} is the weight of the connection from unit *j* to a unit *k* in the next higher layer, and Err_k is the error of *k*. The weights and biases are updated to reflect the propagated errors. Weights are updated by the following equations, where Δw_{ij} is the change in weight w_{ij}:

$$\Delta w_{ij} = (l)\mathrm{Err}_j O_i, \quad (7)$$

$$w_{ij} = w_{ij} + \Delta wij, \quad (8)$$

where *l* is the learning rate. Backpropagation learns using a method of gradient descent to search for a set of weights that fits the training data so as to minimize the mean squared distance between the network's class prediction and the known target value of the tuples [30]. Biases are updated by the following equations, where $\Delta\theta_j$ is the change in bias θ_j:

$$\Delta \quad _j = (l)\mathrm{Err}_j, \quad (9)$$

$$_j = \quad _j + \Delta \quad _j. \quad (10)$$

The whole updating process terminates when all Δw_{ij} get so small as to be below some specified threshold.

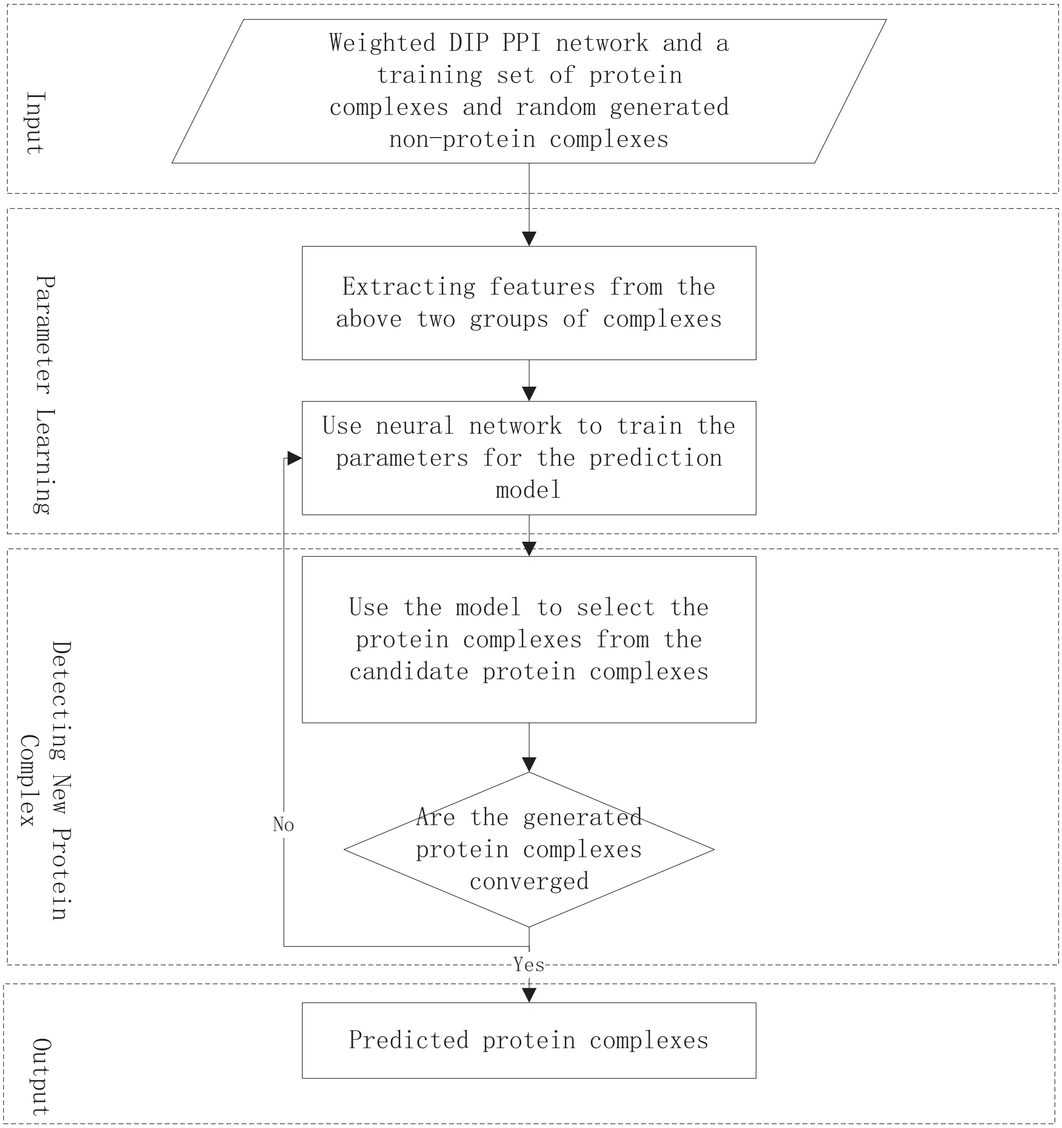
Input
Weighted DIP PPI network and a training set of protein complexes and random generated non-protein complexes
Parameter Learning
Extracting features from the above two groups of complexes
Use neural network to train the parameters for the prediction model
Detecting New Protein Complex
Use the model to select the protein complexes from the candidate protein complexes
Are the generated protein complexes converged
No
Yes
Output
Predicted protein complexes

Figure 5 The flow chart of the whole algorithm

A semi-supervised learning method for new complexes

Based on the above model, we could use it to evaluate the candidate subgraphs. If the evaluating value exceeds the threshold, the candidate subgraph is predicted to be a complex. So the problem becomes finding subgraphs with high evaluating values in the weighted PPI network. However, as proved in [19], identifying the set of maximally scoring subgraphs in large graph is NP-hard. Thus, heuristic algorithms are needed here. There are several approaches that have already been used to solve this problem, such as hill climbing, simulated annealing, and tabu-search heuristic [32].

Here we propose a new heuristic method using an evaluation value as the objective function. At the beginning, each cluster starts at a deterministic single node which we call seed node. In our method, we choose the highest weight node as our seed node. The weights of nodes are determined by summing up the weights of the incident edges. Then we add the neighbor nodes of the cluster one by one to the new cluster and the order is based upon their impact on the evaluation score. The adding process stops when there are no more proteins that could be added and our new protein complex is created. The whole subgraph generation routine is illustrated in Figure 4. Also, we keep tracking the overlapping between the existing clusters and the current investing cluster. If the overlapping rate is over a threshold, we will merge those two complexes.

The completely proposed algorithm for protein complexes identification is described below and the flow chart of the whole algorithm is shown in Figure 5.

- **Input**

– Weighted DIP PPI network and a training set of protein complexes and randomly generated non-protein complexes.

- **Learning parameters step**

– Extract features from the above two groups of complexes.

– Use neural network to train the parameters for the prediction model.

- **Identifying for complexes**

– Start from the seed nodes, add neighboring proteins of the cluster one by one based on the priority and the impact on the cluster.

– Output the complexes when there is no more proteins to satisfy the criterion given above.

– Use the newly generated complexes to recursively update the parameters of the model in the second step and find the new complex.

- **Output**

– Predict protein complexes.

Our input is the weighted PPI graph and a set of known complexes and non-complexes as training data. The known protein complexes are drawn from MIPS protein complexes and the non-complexes are generated randomly from the DIP protein interaction dataset. First, we use the neural network model to learn model parameters from the training data. Once we get the prediction model, we will start searching for the protein complex. Next, when we have more protein complexes, we recursively train our prediction model and find new protein complexes until there are no more proteins that could be added. The final output complexes are those detected clusters which have a higher evaluation score than the threshold.

Acknowledgements
This work was partly supported by NSF grand DBI-0234895.
This article has been published as part of *Proteome Science* Volume 9 Supplement 1, 2011: Proceedings of the International Workshop on Computational Proteomics. The full contents of the supplement are available online at http://www.proteomesci.com/supplements/9/S1.

Author details
[1]Computer Science & Engineering Department, State University of New York at Buffalo, Buffalo, NY, USA. [2]Computer Science Department, Shanxi Normal University, Xi'an, Shanxi Province, 710062, China.

Authors' contributions

LS designed and implemented the algorithm and the framework, analyzed the results and drafted the manuscript. XL participated experiments of the algorithm. AZ coordinated the project and revised the manuscript.

Competing interests

The authors declare that they have no competing interests.

Published: 14 October 2011

References

1. Hwang W, Cho YR, Zhang A, Ramanathan M: **A novel functional module detection algorithm for protein-protein interaction networks.** *Algorithms for Molecular Biology* 2006, **1(24)**.
2. Zhang A: **Protein Interaction Networks: Computational Analysis.** Cambridge University Press; 2009.
3. Hartwell LH, Hopfield JJ, Leibler S, Murray AW: **From molecular to modular cell biology.** *Nature* 1999, **402**:c47-c52.
4. Ito T, Chiba T, Ozawa R, Yoshida M, Hattori M, Sakaki Y: **A comprehensive two-hybrid analysis to explore the yeast protein interactome.** *Proc. Natl. Acad. Sci. USA* 2001, **98(8)**:4569-4574.
5. Aloy P, Russell RB: **Interrogating protein interaction networks through structural biology.** *Proc. Natl. Acad. Sci. USA* 2002, **99(9)**:5896-5901.
6. Chen J, Yuan B: **Detecting functional modules in the yeast protein-protein interaction network.** *Bioinformatics* 2006, **22(18)**:2283-2290.
7. Hwang W, Kim T, Cho YR, Zhang A, Ramanathan M: **SIGN: reliable protein interaction identification by integrating the similarity in GO and the similarity in protein interaction networks.** *Proceedings of 7th IEEE Symposium on Bioinformatics and Bioengineering (BIBE)* 2007, 1384-1388.
8. Hishigaki H, Nakai K, Ono T, Tanigami A, Takagi T: **Assessment of prediction accuracy of protein function from protein-protein interaction data.** *Yeast* 2001, **18**:523-531.
9. Bader GD, Hogue CW: **An automated method for finding molecular complexes in large protein interaction networks.** *BMC Bioinformatics* 2003, **4(2)**.
10. Spirin V, Mirny LA: **Protein complexes and functional modules in molecular networks.** *Proc. Natl. Acad. Sci. USA* 2003, **100(21)**:12123-12128.
11. Li Xiao-Li, *et al*: **Interaction graph mining for protein complexes using local clique mergeing.** *Genome Informatics* 2005, **16(2)**:260-269.
12. Altaf-Ul-Amin M, Shinbo Y, Mihara K, Kurokawa K, Kanaya S: **Development and implementation of an algorithm for detection of protein complexes in large interaction networks.** *BMC Bioinformatics* 2006, **7(207)**.

13. Chua HN, Sung WK, Leong HW, Wong L: **Using indirect protein-protein interactions for protein complex prediction.** *CSB* 2007, 97-109.
14. Chua HN, Sung WK, Wong L: **Exploiting indirect neighbours and topological weight to predict protein function from protein-protein interactions.** *Bioinformatics* 2006, **22(13)**:1623-1630.
15. Ucar D, Asur S, Catalyurek UV, Parthasarathy S: **Improving Functional Modularity in Protein-Protein Interactions Graphs using Hub-Induced Subgraphs.** *PKDD* 2006, **363**:28-42.
16. Adamcsek B, Palla G, Farkas IJ, Derenyi I, Vicsek T: **CFinder: locating cliques and overlapping modules in biological networks.** *Bioinformatics* 2006, **22(8)**:1021-1023.
17. Mete M, Tang F, Xu X, Yuruk N: **A structural approach for finding functional modules from large biological networks.** *BMC Bioinformatics* 2008, **9(Suppl 9)**:SI9.
18. Ester M, Kriegel HP, Sander J, Xu X: **A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise.** *KDD* 1996, 226-231.
19. Qi Yanjun, Balem Fernanda, Faloutsos Christos, Klein-Seetharaman Judith, Bar-Joseph Ziv: **Protein complex identification by supervised graph local clustering.** *Bioinformatics* 2008, **24(13)**:i250-268.
20. Shi L, Cho YR, Zhang A: **ANN Based Protein Function Prediction Using Integrated Protein-Protein Interaction Data.** *Bioinformatics, Systems Biology and Intelligent Computing, International Joint Conference on* 2009, **0**:271-277.
21. Cho YR, Hwang W, Zhang A: **Modularization of protein interaction networks by incorporating Gene Ontology annotations.** *Proceedings of IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)* 2007, 233-238.
22. Salwinski L, Miller CS, Smith AJ, Pettit FK, Bowie JU, Eisenberg D: **The database of interacting proteins: 2004 update.** *Nucleic Acid Research* 2004, **32**:D449-D451.
23. Pei P, Zhang A: **A topological measurement for weighted protein interaction network.** *Proceedings of 16th IEEE Computational Systems Bioinformatics Conference (CSB)* 2005, 268-278.
24. Cho YR, Hwang W, Zhang A: **Efficient modularization of weighted protein interaction networks using k-hop graph reduction.** *Proceedings of 6th IEEE Symposium on Bioinformatics and Bioengineering (BIBE)* 2006, 289-298.
25. Mewes HW, *et al*: **MIPS: analysis and annotation of proteins from whole genome in 2005.** *Nucleic Acid Research* 2006, **34**:D169-D172.
26. Joachims T: **Learning to classify text using support vector machines.** *PhD thesis* Cornell University; 2001.
27. Stark C, Breitkreutz BJ, Reguly T, Boucher L, Breitkreutz A, Tyers M: **BioGRID: a general repository for interaction datasets.** *Nucleic Acids Research* 2006, **34**:D535-D539.
28. Chen L, Shi X, Kong X, Zeng Z, Cai YD: **Identifying protein complexes using hybrid properties.** *Journal of Proteome Research* 2009, **8(11)**:5212-8.
29. Barabási AL, Oltvai ZN: **Network biology: understanding the cell's functional organization.** *Nature Reviews: Genetics* 2004, **5**:101-113.
30. Han JW, Kamber M: **Data Mining concepts and techniques.** Morgan Kaufmann; 2006.
31. Albert R, Jeong H, Barabási AL: **Error and attack tolerance of complex networks.** *Nature* 2000, **406**:378-482.
32. Virtanen SE: **Properties of nonuniform random graph models.** *Research Report. Helsinki University of Technology, Laboratory for Theoretical Computer Science* 2003.

doi:10.1186/1477-5956-9-S1-S5
Cite this article as: Shi *et al.*: Protein complex detection with semi-supervised learning in protein interaction networks. *Proteome Science* 2011 **9**(Suppl 1):S5.