Peer-reviewed version available at Appl. Sci. 2017, 7, , 412; doi:10.3390/app7040412

Article

Simplified Swarm Optimization Based Function Modules Detection in Protein-to-Protein Interaction Networks

Xianghan Zheng [1,2,*], Lingting Wu [1,2], Shaozhen Ye [1], Riqing Chen [3]

1 College of Mathematics and Computer Science, Fuzhou University, Fuzhou, China
2 Fujian Key Laboratory of Network Computing and Intelligent Information Processing, Fuzhou, China
3 Fujian Agriculture and Forestry University, Fuzhou University, Fuzhou, China
* Correspondence: xianghan.zheng@fzu.edu.cn

Abstract. Proteomics research has become one of the most important topics in the fields of life science and natural science. At present, research on protein–protein interaction networks (PPINs) mainly focuses on detecting protein complexes or function modules. However, existing approaches are either ineffective or incomplete. In this paper, we investigate function module detection mechanisms in PPIN, including open databases, existing detection algorithms and recent solutions. After that, we describe the proposed solution based on simplified swarm optimization (SSO) algorithm and gene ontology knowledge. The proposed solution implements SSO algorithm for clustering proteins with similar function, and imports biological gene ontology knowledge for further identifying function complexes and improving detection accuracy. Furthermore, we use four different categories of species dataset for experiment: Fruitfly, Mouse, Scere, and Human. The testing and analysis result show that the proposed solution is feasible, efficient and could achieve a higher accuracy of prediction than existing approaches.

Keywords: protein–protein interaction networks; protein function module; simplified swarm optimization

1. Introduction

Proteomics is one of the most important topics in the fields of life science and natural science [4–6]. Considering that proteins alone rarely exhibit their biological functions in independent individuals, the understanding of protein–protein interactions (PPI) is the basis to reveal the activity of protein and promotes the study of various diseases and development of new drug.

In the past 10 years, substantial work was conducted to promote the research in the field of PPI, such as publications in Nature [3] and Science [4], proceedings of the National Academy of Sciences [5], and nucleic acid research [6]. Available data on PPI are greatly enriched because of the fast development of high-throughput screening [7] and data mining technologies [8]. Some widely used and most complete open dataset are also released, for instance, Biomolecular Interaction Network Database (BIND) [9], Database of Interaction Proteins (DIP) [10], IntAct [11], Human Protein Reference Database (HPRD) [12], and Molecular Interaction Database (MINT) [13].

However, the existing solution is incomplete or inaccurate due to the following technical challenges. On one hand, high-throughput screening technology generates a huge amount of noise data and higher false positive rate, while the experimental method loses lots of real interactions (false negative) [1]. On the other hand, the existing computation approaches (Graph theory-, machine learning-, and intelligent algorithm-based described in section 2) are inefficient, computational complex, or lack of convincible result in PPI network with a huge amount of nodes and dynamic structure. In this paper, we investigate the function module detection in PPIN and describe the proposed a lightweight and efficient simplified swarm optimization based PPI function module detection, with the contributions in the following points:

1. We investigate PPI dataset and the existing function module detection methods as well as select

Peer-reviewed version available at Appl. Sci. 2017, 7, , 412; doi:10.3390/app7040412

the interaction data of four typical species of protein from the DIP database as our research data. A specific data crawler is developed to extract the data feature from the four species.

2. We describe the proposed PPIN function module detection from a few aspects: system model, feature selection, mathematic description, and model optimization, etc. The proposed solution implements SSO algorithm for clustering proteins with similar function and imports biological gene ontology knowledge for further identification of the function complexes.
3. We conduct experiments to validate the feasibility and efficiency of the proposed solution. The evaluation of "Degree of polymerization" and "Similarity between classes" further proves the precision improvement and correctness of our proposed solution.

The paper is organized as follows. Section 2 introduces the existing research, including the graph theory-, machine learning-, and intelligent algorithm-based approaches. Section 3 describes our solution, including the system model, dataset pre-processing, and SSO-based solution. Experiments, including dataset description and result evaluation, are explained in Section 4. Finally, the conclusions are presented in Section 5.

2. Related Works

2.1. Protein–Protein Interaction Datasets

In the following, we introduce the most widely used and complete databases:

BIND contains the known interactions among biological molecules, not only among proteins but also between proteins and DNA, RNA, small molecules, lipids, and carbohydrate substances. BIND is updated daily and has an extensive coverage, including the PPIs of people, fruit flies, yeast, nematodes, and other species.

DIP specializes in storing the binary PPIs from the literature that are confirmed by experiments, as well as the protein complexes from the Protein Data Bank (PDB). DIP is created to establish a simple, easy-to-use, and highly credible PPI public database.

IntAct mainly records the binary interactions and their experiment methods, experimental conditions, and interaction domain structures, including those in people, yeast, fruit flies, *Escherichia coli*, and other species. IntAct query is divided into basic and advanced queries; the latter is more accurate.

HPRD contains protein annotations, PPIs, posttranslational modifications, subcellular localizations, and other comprehensive information. HPRD PPI data are classified based on two methods, namely, on interaction topology and on experiment type.

MINT mainly stores the physical interaction of proteins, particularly the PPIs of mammals. This database also contains the PPIs of yeast, fruit flies, and viruses. MINT is similar to DIP database, in which a homologous interaction can be searched according to the BLAST sequence.

Considering the deviation of definition and the term promiscuity in different databases, Gene Ontology (GO), which is developed and maintained by the GO Consortium, should be imported for the sharing and interoperability of the bioinformatics data. Therefore, the retrieval results between different databases are more unified.

2.2. Existing Works

In the past 10 years, the existing works in the detection of protein function modules are divided into three categories:

Graph theory-based approach. Similar to other networks, the graph theory is imported into PPIN for the detection of protein function module and is mainly conducted in three approaches: hierarchical algorithm, partitioning algorithm, and density algorithm [14]. The core idea of hierarchical algorithm is based on the similarity of the connections between each node, such as the modularity division-based method [15]. For the partitioning algorithm, the most represented method is based on the restricted neighborhood search clustering (RNSC) [16]. Although both hierarchical and partitioning algorithms are easy to understand and implement, the clustering number should be determined beforehand and the function modules cannot overlap.

For the density algorithm, if the density of a region is larger than a given threshold value, then this region will be added to the cluster. Typical algorithms include the clique percolation method [3] and molecular complex detection (MCODE) [17], both of which compute the protein complexes based on the connection value among the proteins. Although these two methods obtain a certain degree of protein function detection, some clusters are too thin because of the considerable weights between the nodes that are loosely connected.

Machine Learning-based approaches. One typical solution is based on the Markov model. Considering that the original MCL algorithm exhibits the disadvantages of poor scalability and low clustering quality, Lei et al. (2015) proposed the improved MCL clustering algorithm for PPIN [18] by importing two parameters: punishment and mutation factors. This approach improved the convergence speed but included substantial computation complexity.

Additionally, a group of algorithms combined with other computational mechanisms emerged in the field of PPIN recognition. The first attempt was based on the hybrid SVM-RVM algorithm [19]. In literature [20], Deddy proposed a rapid prediction algorithm for PPINs based on ELM algorithm, which has the speed advantage of the ELM algorithm and achieves better protein prediction result. Additionally, deep learning technology is considered for protein function detection. This model is composed of two parts: the deep feature memory (DME) and the associative memory (AME), as illustrated in Figure 1. The hidden structures of each network are learned through the DME phase, whereas the common features of the hidden structures are derived using the AME model.

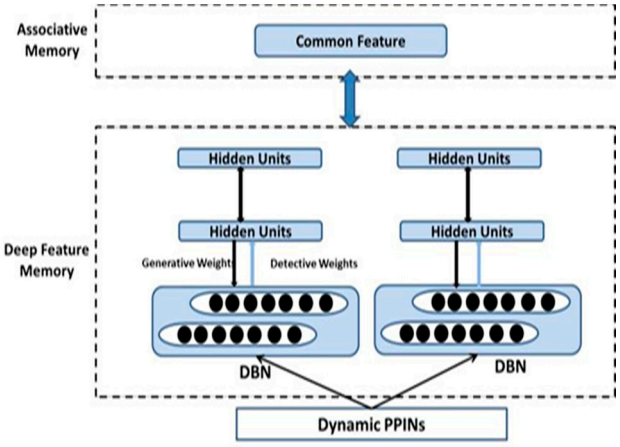

Figure 1. Deep Learning based Protein Function Detection

Intelligence algorithm-based approaches. Swarm intelligence algorithms are also implemented for PPIN function module detection. The swarm intelligence algorithm imitates the cooperative behavior of group biology (such as bees, ants, and birds) to abstract modeling, forming the ideas to solve the problem of all kinds of combinatorial optimization. Examples of these algorithms include ant colony optimization (ACO) [21], particle swarm optimization (PSO) [22], and artificial bee colony (ABC) [23].

Sallim first applied the ACO algorithm to the PPIN complex clustering problem, and further used the optimization ACO method in the protein interaction networks (ACOPIN) [24]. In 2012, Ji introduced a new heuristic function, namely, the NACO–FDM [25] algorithm, to improve the process of ACO in searching for the optimal path. However, this algorithm easily falls into the local optimum. In literature [26], a new ACO-MAE mechanism that combines ACO with the idea of multi-agent evolution (MAE) was developed to achieve better prediction accuracy.

2.3. Motivation

Although many studies were conducted, a few disadvantages were observed for these approaches. Graph-based approaches are far from being precise (with highest precision rate of 46% [16]) because some clusters are too thin due to the considerable weights between the loosely connected nodes. Most existing machine/deep learning-based approaches need huge amount of denoted sample for training, which is difficult to implement in PPIN research. On the contrary, although the Intelligence algorithms (mainly ACO-related approaches) implemented in PPI data showed better precision rate and efficiency than the graph-based solution, more intelligent algorithms should be considered and implemented.

The detection of protein interaction/function is a NP hard problem [27]. The PSO-related solution is efficient and was implemented in different kinds of NP hard problems. Therefore, we propose an improved and lightweight PSO algorithm, Simplified Swarm Optimization (SSO), to implement in the detection of protein function module. Theoretically, SSO solution achieves better precision than PSO algorithm and reduces the computing complexity.

3. Simplified Swarm Optimization-based Detection

In this section, we describe the proposed solution based on SSO algorithm in the following three steps: system model, feature extraction, and model description.

3.1. Interaction Model

Figure 2 (a–e) illustrates the process of protein function module detection. First, the PPI network is abstracted into the format of protein distance matrix. The network model is built by the measure of the distance between the proteins. Afterwards, the SSO algorithm is imported to search the shortest path among each node. Finally, the cutting and filtering strategies are defined and implemented to generate clustering results.

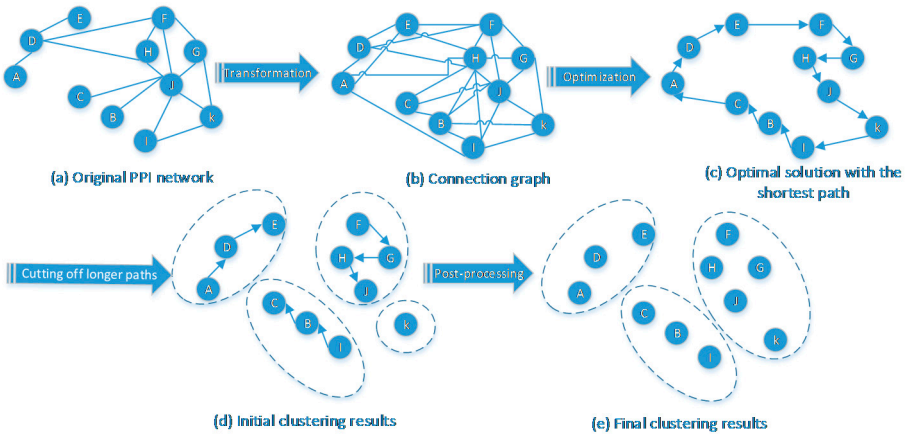

Figure 2. Interaction Model

3.2. Feature Selection and Extraction

After acquiring the original data from DIP and Go databases, the feature extraction, an important operation, is illustrated as follows:

1. Noise Filter. Noise data refers to the existence of errors in the crawled data, redundant data, or abnormal data. For example, in XML-based crawled data, the tag field "DIP:nnE" may be empty or not found. Therefore, eliminating noise and redundant data is the first step before the experiment.
2. Feature Selection. Feature selection is performed through the manual respection of protein xml data. For example, in the main part of the XML file, the tag name interactorList and

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 February 2017 doi:10.20944/preprints201702.0052.v1

Peer-reviewed version available at Appl. Sci. 2017, 7, , 412; doi:10.3390/app7040412

interactionList indicate the interaction relationship among protein nodes. Therefore, feature data are selected through the manual respection of protein data.

3. Feature Extraction and Reformat. After the feature selection, related data (e.g., protein id and interactor id) are extracted, reformatted, and stored in the structured database.

3.3. Model Description

3.3.1. Establishment of the PPIN Model

The SSO algorithm is described as follows. The initial particle swarm size n, the problem space dimensions m and the position are randomly assumed; the location $X_i^t = (x_{i1}^t, x_{i2}^t, ...x_{im}^t)$, where i = 1,2,3,…n, x_{im}^t is the value of the i-th particle with respect to the m-th dimension of the feature space at time t. The particles in the search process reach the optimal location and are marked as p^t_i. The optimal location for the group is g^t_i. Therefore, the location of particle i in j dimension at time t is described using the following formulas:

$$x_{ij}^{t+1} = \begin{cases} X_{ij}^{t}, & if \quad random \in [0, C_w); \\ p_{ij}^{t}, & if \quad random \in [C_w, C_p); \\ g_{ij}^{t}, & if \quad random \in [C_p, C_g \\ X, & if \quad random \in [C_g, 1). \end{cases} \qquad (1)$$

In Formula (1), X represents the new value of the particle in every dimension that are randomly generated from the random function; the random number is between *(0, 1)*. C_w, C_p, and C_g are the three predetermined positive constants with $C_w < C_p < C_g$.

In this study, we use the topological structure of a PPIN[28] as the basis, with the individual protein as a node and the interactions between proteins as an edge, to construct a PPIN model. The protein interaction network is shown in Figure 3, whereas the adjacency matrix is presented in Table 1. The interaction that occurs between the proteins is denoted as *1*, whereas no relationship found between proteins is denoted as *0*.

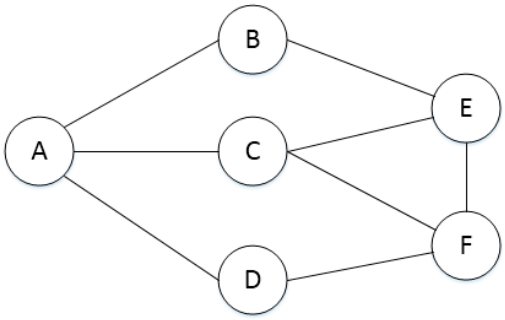

Figure 3. Protein interaction network

Table 1. Adjacency matrix corresponds to figure 2

	A	B	C	D	E	F
A	0	1	1	1	0	0
B	1	0	0	0	1	0
C	1	0	0	0	1	1
D	1	0	0	0	0	1
E	0	1	1	0	0	1
F	0	0	1	1	1	0

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 February 2017 doi:10.20944/preprints201702.0052.v1

Peer-reviewed version available at Appl. Sci. 2017, 7, , 412; doi:10.3390/app7040412

Therefore, the distance between proteins, d_{ij}, is calculated according to the adjacency matrix using Formula (2):

$$d_{ij} = \frac{|Int(i) \cup Int(j)| - |Int(i) \cap Int(j)|}{|Int(i) \cup Int(j)| + |Int(i) \cap Int(j)|} \quad (2)$$

where *i* and *j* express the two proteins, *Int(i)* and *Int(j)*, respectively on behalf of protein *i* and *j* with the sum of the number of adjacent protein. The molecular denotes the symmetrical difference of these proteins. The value of *dij* is between *0~1* including *1*. Normally, the value of *dij* is greater than *0*; in some special cases, when protein *i* and *j* are in a completely different set of proteins, the *dij* achieves the highest value (i.e., 1).

3.3.2. Parameter Setting

Parameter setting always plays a key important role in detecting function modules among the solutions. In SSO algorithm, the initial location of a particle swarm is random and not close to the actual problem. In addition, C_w, C_p, and C_g are usually set as 0.25, 0.5, and 0.75, respectively. This parameter easily falls into the local optimum. The values of C_w, C_p and C_g in this paper are set as 0.1, 0.55, and 0.8 to expand the search area to a global search at the beginning of the iteration in PPI network. Table 2 shows the parameter setting of the SSO and PSO algorithm used in the experiments. The letter *t* expresses the t-th Max_GEN.

Table 2. Parameter setting for PSO and SSO algorithm

Parameter Setting	PSO	SSO
MAX_GEN	500	500
Number of Particle	100	100
Maximum Fitness	1.0	1.0
C_w, C_p, C_g	-	0.1, 0.55, 0.8
Weight	0.9-0.4*t/MAX_GEN	-
c_1,c_2	2.0, t/MAX_GEN	-

3.4. Model Optimization

Function module optimization is divided into two main parts: module planning based on function information and on topology.

Module Planning Based on Function Information

The objective of this step is to merge the similar initial PFMs. The basic idea is to use the similarity measure of the two modules, i.e., Formula (3).When the similarity is greater than a certain threshold, the two modules will merge. Formula (3) is presented as follows:

$$S(M_S, M_T) = \frac{\sum_{i \in M_S, j \in M_T} s(i, j)}{\min(|M_S|, |M_T|)} \quad (3)$$

where *M*S and M_T represent the sizes of the two modules (including the number of proteins), respectively, and *S(i, j)* is characterized by the following formula (4):

$$s(i, j) = \begin{cases} 1, & if \quad i = j \\ f_{ij}, & if \quad i \neq j \end{cases} \quad (4)$$

Among these paramaters, f_{ij} is the similarity function based on gene topology and is characterized by the following formula (5) [29]:

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 February 2017 doi:10.20944/preprints201702.0052.v1

Peer-reviewed version available at Appl. Sci. 2017, 7, , 412; doi:10.3390/app7040412

$$f_{ij} = \frac{|g^i \cap g^j|}{|g^i \cup g^j|} \quad (5)$$

In Formula (7), g^i and g^j represent the comments values of protein j and protein i in the Gene Ontology, respectively [30]. The greater value of f_{ij} indicates the higher similarity between the two proteins.

Module Planning Based on Topology

This step measures the density of the initial module and reduces the sparse protein module through filter setting. The density of the module is calculated according to this formula (6):

$$D_S = \frac{e}{n * (n - 1) / 2} \quad (6)$$

where n denotes the number of the current module and e represents the number of interaction in the module.

4. Experiments

4.1. Dataset description

Figures 4 and Figure 5 illustrate the XML format protein interaction data (species Scere from DIP database), in which the key content is marked by a red circle. Figure 4 shows the interaction relationship (with the identifier DIP-436E), as well as the proteins involved in the interaction that are marked by 780 and 18, respectively. Therefore, the final result is expressed as: DIP-436E, DIP-780N, DIP-18N. Figure 5 shows the data format of interactor that indicates the number of associated proteins in this species. The final result sets are DIP-6847N, DIP-6848N, and DIP-6849N.

```
<interaction id="436">
<xref>
<primaryRef db="dip" dbAc="MI:0465" refType="identity" refTypeAc="MI:0356" id="DIP-436E"
</xref>
<experimentList>
 ......
<participantList>
<participant id="1">
<names>
<shortLabel>Sip3p</shortLabel>
<fullName>SIP3 protein</fullName>
</names>
<interactorRef>780</interactorRef>
</participant>
<participant id="2">
<names>
<shortLabel>SNF1</shortLabel>
<fullName>Carbon catabolite-derepressing protein kinase</fullName>
</names>
<interactorRef>18</interactorRef>
</participant>
</participantList>
```

Figure 4. Xml Format of Protein Interaction

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 February 2017 doi:10.20944/preprints201702.0052.v1

Peer-reviewed version available at Appl. Sci. 2017, 7, , 412; doi:10.3390/app7040412


```
<interactor id="6847">
<interactor id="6848">
<interactor id="6849">
<names>
<shortLabel>metK</shortLabel>
<fullName>S-adenosylmethionine synthetase (Methionine adenosyltransferase) (AdoMet synthetase) (MAT)</fullName>
</names>
<xref>
<primaryRef db="dip" dbAc="MI:0465" id="DIP-6849N" refType="identity" refTypeAc="MI:0356"/>
<secondaryRef db="refseq" dbAc="MI:0481" id="NP_289514" refType="identity" refTypeAc="MI:0356"/>
<secondaryRef db="uniprot knowledge base" dbAc="MI:0486" id="P04384" refType="identity" refTypeAc="MI:0356"/>
<secondaryRef db="entrez gene/locuslink" dbAc="MI:0477" id="945389" refType="gene product" refTypeAc="MI:0251"/>
</xref>
<interactorType>
<names>
<fullName>protein</fullName>
</names>
<xref>
<primaryRef refTypeAc="MI:0356" refType="identity" id="MI:0326" dbAc="MI:0488" db="psi-mi"/>
</xref>
</interactorType>
<organism ncbiTaxId="83333">
<names>
<shortLabel>Escherichia coli K12</shortLabel>
</names>
</organism>
</interactor>
<interactor id="6851">
```

Figure 5. Xml Format of Protein Interactor

As previously mentioned, the existing format of the PPI and GO raw data is not the format we used in our experiments. The data must be pre-processed before the experiment. For the raw data of PPI that were downloaded from the DIP library, we need to extract the protein node and PPI data and then conduct the format conversion. The process flow diagram is shown in Figure 6, including the removal of redundant data, deletion of incomplete data, and coordination of DIP with GO data.

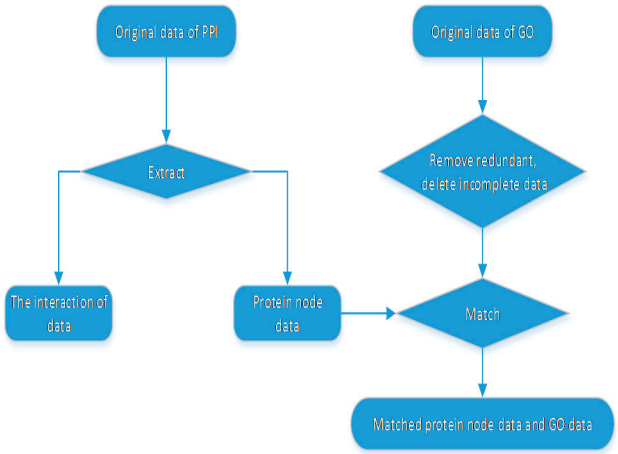

Figure 6. Data preprocessing flow

4.2. Results of Description

We use four different categories of species data: Fruitfly, Mouse, Scere, and Human. Among these categories, eight different thresholds were selected for each species in the experiment: 0.05, 0.055, 0.06, 0.065, 0.07, 0.075, 0.08, and 0.085. Additionally, we conduct a statistical analysis of the experimental results, and the results are shown in Figures 7–9.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 February 2017 doi:10.20944/preprints201702.0052.v1

Peer-reviewed version available at Appl. Sci. 2017, 7, , 412; doi:10.3390/app7040412

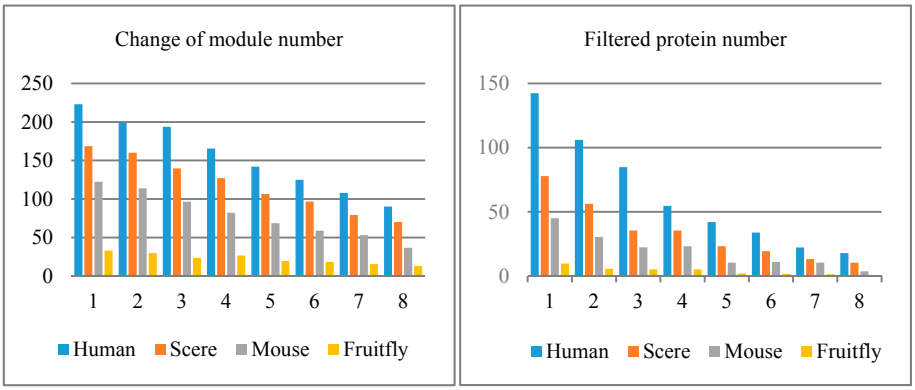

Figure 7. Integrate the experimental results of the four species together. Threshold change in response to 1(0.05), 2(0.055), 3(0.06), 4(0.065), 5(0.07), 6(0.075), 7(0.08), 8(0.085). (A) module number change during the course of the experiment. (B)filtered protein number in each threshold.

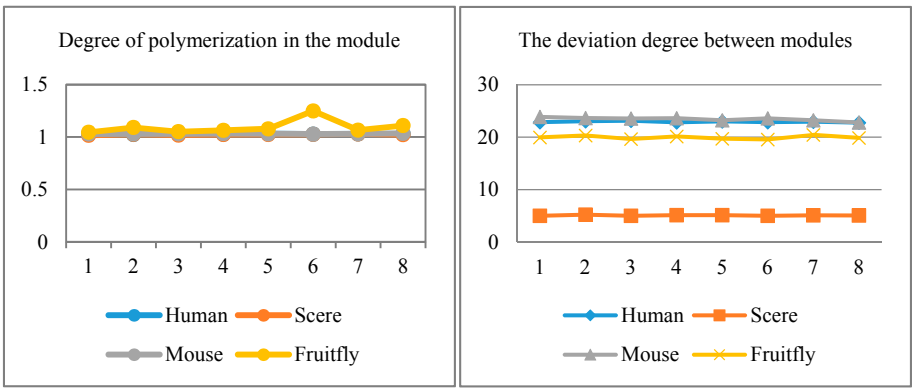

Figure 8. Integrate the experimental results of the four species together. Threshold change in response to 1(0.05), 2(0.055), 3(0.06), 4(0.065), 5(0.07), 6(0.075), 7(0.08), 8(0.085). (C) degree of polymerization in the module. (D) the deviation degree between modules.

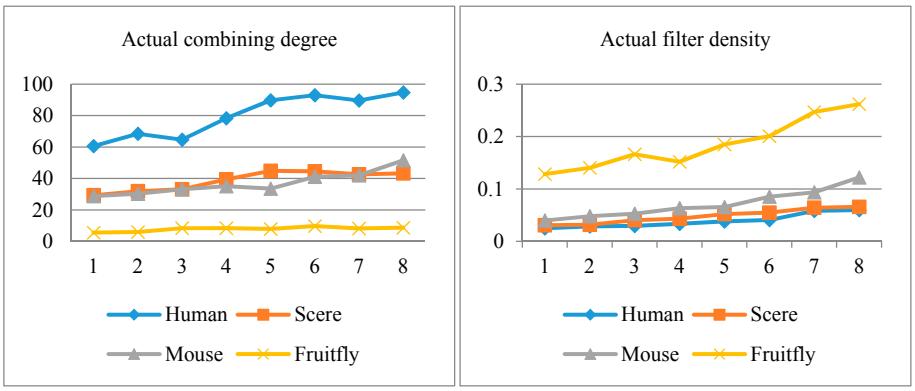

Figure 9. Integrate the experimental results of the four species together. Threshold change in response to 1(0.05), 2(0.055), 3(0.06), 4(0.065), 5(0.07), 6(0.075), 7(0.08), 8(0.085). (E) actual combining degree in each threshold. (F) actual filter density during the experiment.

Results show that the threshold increases to a certain extent. For the large scale species (here it refers to the number of protein), the unnecessary protein filtration is greatly reduced. The size of the module also increases. However, the other aspects of the effect are not significant and hence remained at the same level.

Meanwhile, we compare SSO and PSO of the experimental results from the six aspects. The result generated by SSO is better than that of PSO, which is mainly reflected in the following aspects shown in Tables 3–5. Table 3 shows the changes of module number for PSO and SSO algorithms. The

Peer-reviewed version available at Appl. Sci. 2017, 7, , 412; doi:10.3390/app7040412

result indicates that the value of SSO is basically lower but close than that of PSO in addition to the species of Fruitfly. Table 4 shows the filtered protein number for PSO and SSO algorithms. The results of SSO are significantly better than those of PSO, especially when the number of proteins in the species increased. Table 5 shows the degree of polymerization in the module for PSO and SSO algorithms. A higher value indicates higher similarity in the module. The result reveals that the two algorithms are relatively close; however, the SSO algorithm has better stability.

Table 3. Change of module number for PSO and SSO algorithm

Threshold change of module number category	SSO				PSO			
	Fruit fly	Mouse	Scere	Human	Fruit fly	Mouse	Scere	Human
0.05	33	122.4	168.6	223	25.8	94.8	174.2	223.4
0.055	30	114	160	198.8	27	92.2	163.8	202.8
0.06	23.6	96.6	139.8	193.6	22.6	82.6	139.8	187.4
0.065	26.8	82.2	127	165.4	18	72.4	129.6	161.2
0.07	19.8	68.8	106.6	142	16.2	66	110.8	147
0.075	18.4	59	96.8	125	14.4	60.8	101.8	126
0.08	15.6	53.2	79.4	108	15.4	46.4	79.2	109
0.085	13.2	36.8	70.2	90.4	11.6	41.8	74.6	96.8

Table 4. Filtered protein number for PSO and SSO algorithm

Threshold filtered protein number category	SSO				PSO			
	Fruit fly	Mouse	Scere	Human	Fruit fly	Mouse	Scere	Human
0.05	9.8	45	77.8	142.4	10	280.2	342.4	706
0.055	5.6	30.4	56.2	106	9	172.6	205.2	432.8
0.06	5.2	22.4	35.4	84.8	6	95	102	270
0.065	5.2	23.2	35.4	54.6	4.4	56	64	234.4
0.07	2	10.4	23.2	42	3.4	45.6	35.8	157
0.075	1.6	11	19.4	33.8	4.4	17.8	33.8	100.8
0.08	1.4	10.4	13.2	22.2	1.4	12	18	70.2
0.085	0.8	3.6	10.4	17.8	1.4	6.4	13.8	48.2

Table 5. Degree of polymerization in the module for PSO and SSO algorithm

Threshold degree of polymerization in the module category	SSO				PSO			
	Fruit fly	Mouse	Scere	Human	Fruit fly	Mouse	Scere	Human
0.05	1.04 48	1.02 8	1.01 6	1.02 7	1.08 42	1.02 88	1.01 78	1.02 72
0.055	1.09 06	1.02 98	1.02 56	1.02 42	1.06 88	1.02 78	1.01 72	1.02 76
0.06	1.05 16	1.02 96	1.01 82	1.02 78	1.06 66	1.03 34	1.02 48	1.02 74
0.065	1.06 34	1.03 26	1.02 36	1.02 76	1.07 1	1.03 1	1.02 5	1.03 1
0.07	1.07 78	1.03 06	1.02 46	1.03 08	1.08 52	1.03 56	1.02 38	1.03 12
0.075	1.24 86	1.02 74	1.02 5	1.02 76	1.09 3	1.02 96	1.02 54	1.02 88
0.08	1.06 58	1.03	1.02 9	1.03 02	1.09 8	1.03 34	1.02 44	1.02 96
0.085	1.10 94	1.03 42	1.02 2	1.03 02	1.10 5	1.03 62	1.02 82	1.04 08

5. Conclusions

In this study, we introduce relevant research on protein interaction networks conducted in recent years, including the commonly used protein databases and traditional detection methods. We then describe our proposed SSO algorithm for the detection problem of PFM in PPIN. Simultaneously, biological gene ontology knowledge is combined to improve the prediction accuracy. The performance of SSO is compared with that of PSO algorithm through the analysis of the experimental results. Results show that the SSO algorithm has great advantage in the detection of protein function. For example, in the experiment, the clustering results show that both the degree of polymerization in the module and the deviation degree between modules are maintained in a relatively stable level and have values within the expected range. the SSO is better than PSO based on the accuracy, including the module number, filtered protein number, and more stable degree of polymerization in the module.

Acknowledgments: All sources of funding of the study should be disclosed. Please clearly indicate grants that you have received in support of your research work. Clearly state if you received funds for covering the costs to publish in open access.

Author Contributions: For research articles with several authors, a short paragraph specifying their individual contributions must be provided. The following statements should be used “X.X. and Y.Y. conceived and designed the experiments; X.X. performed the experiments; X.X. and Y.Y. analyzed the data; W.W. contributed reagents/materials/analysis tools; Y.Y. wrote the paper.” Authorship must be limited to those who have contributed substantially to the work reported.

Conflicts of Interest: Declare conflicts of interest or state “The authors declare no conflict of interest."

References

1. Ji J Z, Jiao L, Yang C C, et al. MAE-FMD: Multi-agent evolutionary method for functional module detection in protein-protein interaction networks[J]. BMC Bioinformatics, 2014, 15(1):325.
2. Islam M F, Hoque M M, Banik R S, et al. Comparative analysis of differential network modularity in tissue specific normal and cancer protein interaction networks[J]. Journal of Clinical Bioinformatics, 2013, 3(1):19.
3. Yong-Yeol Ahn, James P. Bagrow, Sune Lehmann. Link communities reveal multiscale complexity in networks. Nature, 2010, 435(7307): 761-764.
4. Aashiq H. Kachroo1, Jon M. Laurent1, Christopher M. Yellman1, et al.Systematic humanization of yeast genes reveals conserved functions and genetic modularity. Science, 2015, 348(65237):921-925.
5. Tanay A, Sharan R, Kupiec M, Shamir R. Revealing modularity and organization in the yeast molecular network by integrated analysis of highly heterogeneous genomewide data. Proceedings of the National Academy of Sciences of the United States of America, 2004, 101(9): 2981-2986.
6. Damian Szklarczyk, Andrea Franceschini, Stefan Wyder, et al .Protein–protein interaction networks, integrated over the tree of life. Nucleic Acids Res, 2015, 39 (suppl 1): D561-D568.
7. Damian Szklarczyk1, Andrea Franceschini2, Michael Kuhn3, et al. The STRING database in 2011: functional interaction networks of proteins, globally integrated and scored. Nucleic Acids Res, 2011, 1093(973):D561-D568.
8. H Xu，X Li，Z Zhang，J Song.Identifying Coevolution Between Amino Acid Residues in Protein Families: Advances in the Improvement and Evaluation of Correlated Mutation Algorithms. Current Bioinformatics, 2013,8(2):148-160(13).
9. Hongchun Li, Yuan-Yu Chang, Lee-Wei Yang, Ivet Bahar. the Gaussian network model database for biomolecular structural dynamics. Nucleic Acids Res, 2016, 44 (D1): D415-D422.
10. Philipp Blohm, Goar Frishman, Pawel Smialowski. A database of non-interacting proteins derived by literature mining, manual annotation and protein structure analysis. Nucleic Acids Res, 2014, 42 (D1): D396-D400.
11. Sandra Orchard1, Mais Ammari2, Bruno Aranda, Lionel Breuza, et al. The MIntAct project—IntAct as a common curation platform for 11 molecular interaction databases. Nucleic Acids Res, 2014, 1093(D1): D358-D363.
12. Chie Motono, Junichi Nakata, Ryotaro Koike. A comprehensive database of predicted structures of all human proteins. Nucleic Acids Res, 2011, 39 (suppl 1): D487-D493.
13. Luana Licata, Leonardo Briganti, Daniele Peluso. MINT, the molecular interaction database: 2012 update. Nucleic Acids Res, 2012, 40 (D1): D857-D861.
14. Ester B M, Kriegel H P, Sander J. et al. A Density Based algorithm for discovering clusters in large spatial databases[C].Proceedings of International Conference on knowledge Discovery and Data Mining, AAAI. 2013.
15. Newman M E. Fast algorithm for detecting community structure in networks[J]. Physical Review E Statistical Nonlinear & Soft Matter Physics, 2004, 69(6):066133-066133.
16. Hartuv E, Shamir R. A clustering algorithm based on graph connectivity[J]. Information processing letters, 2000, 76(4): 175-181.
17. Bader G D, Hogue C W V. An automated method for finding molecular complexes in large protein interaction networks[J]. BMC bioinformatics, 2003, 4(1): 2.
18. Qing-sheng hu, Xiu-juan lei. PPI network improved markov clustering algorithm[J]. Journal of computer science, 2015 (7) : 108-113.
19. Ruan P, Hayashida M, Maruyama O, et al. Prediction of heterotrimeric protein complexes by two-phase learning using neighboring kernels[J]. BMC bioinformatics, 2014, 15(2): S6.
20. Lei X J. The Information Flow Clustering Model and Algorithm Based on the Artificial Bee Colony Mechanism of PPI Network[J]. Chinese Journal of Computers, 2012, 35(1):134-145.
21. Dorigo M, el al. Ant Colony Optimization. MIT Press/Bradford Books, Cambridge, MA 2004.
22. J Kennedy, R Eberhart. Particle swarm optimization[C].Neural Networks, 1995. Proceed-ings, IEEE International Conference on. IEEE, 1995:1942-1948 vol.4.
23. Karaboga D, Basturk B. On the performance of artificial bee colony(ABC)algorithm. Ap-plied Soft Computing, 2008, 8(1):687-697.
24. Sallim, Jamaludin, et al. ACOPIN: An ACO Algorithm with TSP Approach for Clustering Proteins from

Protein Interaction Network[C].Uksim European Symposium on Computer Modeling & Simulation. IEEE Computer Society, 2008:203-208.

25. Ji J Z, Liu Z J, et al. Improve ant colony optimization for detecting functional modules in protein-protein interaction networks. Proceedings of the 3rd International Conference on Information Computing and Applications. Berlin, Heidelberg: Springer, 2012. 404-413
26. Ji J Z, Liu Z J, et al Ant colony optimization with multi-agent evolution for detecting func-tional modules in protein-protein interaction networks. Proceedings of the 3rd Interna-tional Conference on Information Computing and Applications. Berlin, Heidelberg: Spring-er,2012. 445-453
27. Debby D W, Ran W and Hong Y. Fast prediction of protein-protein interaction sites based on Extreme Learning Machines[j]. Neurocomputing 128.128(2014):258-266.
28. Zhang Y, Cheng Y, Jia K. A generative model of identifying informative proteins from dy-namic PPI networks[J]. Science China-life Sciences, 2014, 57(11): 1080-1089.
29. Schlicker A, Albrecht M. FunSimMat: a comprehensive functional similarity database[J]. Nucleic Acids Research, 2008.
30. Ashburner M, Ball C J, Botstein D, et al. Gene ontology: tool for the unification of biology. The Gene Ontology Consortium.[J]. Nature Genetics, 2000, 25(1):25-29.