

Research Article

MOfinder: A Novel Algorithm for Detecting Overlapping Modules from Protein-Protein Interaction Network

Qi Yu,^{1,2} Gong-Hua Li,¹ and Jing-Fei Huang^{1,3}

¹ State Key Laboratory of Genetic Resources and Evolution, Kunming Institute of Zoology, Chinese Academy of Sciences, Kunming 650223, China

² Graduate School of the Chinese Academy of Sciences, Kunming 650223, China

³ Kunming Institute of Zoology-Chinese University of Hong Kong Joint Research Center for Bio-Resources and Human Disease Mechanisms, Kunming 650223, China

Correspondence should be addressed to Jing-Fei Huang, huangjf@mail.kiz.ac.cn

Received 7 September 2011; Revised 19 October 2011; Accepted 21 October 2011

Academic Editor: T. Akutsu

Copyright © 2012 Qi Yu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Since organism development and many critical cell biology processes are organized in modular patterns, many algorithms have been proposed to detect modules. In this study, a new method, MOfinder, was developed to detect overlapping modules in a protein-protein interaction (PPI) network. We demonstrate that our method is more accurate than other 5 methods. Then, we applied MOfinder to yeast and human PPI network and explored the overlapping information. Using the overlapping modules of human PPI network, we constructed the module-module communication network. Functional annotation showed that the immune-related and cancer-related proteins were always together and present in the same modules, which offer some clues for immune therapy for cancer. Our study around overlapping modules suggests a new perspective on the analysis of PPI network and improves our understanding of disease.

1. Introduction

PPI networks have been widely used to understand biology at the system level [1–3]. However, PPI data sets suffer from high false positive and false negative rates [4]. Network module, a group of proteins that are connected with each other to carry out a function [5], will be more accurate because a loss or gain of interaction will not break down the module structure. Modules have been applied to predict protein function [6] and disease genes [7] and trace the evolutionary history of networks [8–10].

To perform complex biochemical or developmental functions, modules have to work together. Thus several proteins are used to pass information from one module to another. For example, three modules in *S. cerevisiae*—the Set3C complex, protein phosphatase type 2A (PP2A) complex, and cell polarity budding—share a protein: Zds1 [11]. Zds1 can bind PP2A to control mitotic progression [12], and it also participates in Set3C complex during budding processes and repress meiotic process [13], so Zds1 may serve as a bridge between

mitosis and meiosis. Here we define these three modules as overlapping modules and define the shared protein as the overlapping nodes. The overlapping modules can form a module-module communication network. Construction of such network can be helpful for understanding the coordinated relationship between different biological processes.

The problem of identifying modules has been studied by bioinformatics, applied mathematics, and physics [14]. Many methods have been developed to identify modules within a network, and they have been reviewed and evaluated [15–19]. We thought these approaches can be classified into two types. (1) Local seed-based methods which start from a node or clique (fully connected subgraph) and follow by an expanding search strategy. MCODE [20] is the first method for module detecting, and it expands highly scoring seed nodes by a local search procedure. But this method only detects a few modules. CFinder [11] is the first algorithm for overlapping communities detection, and it develops a Clique Percolation Method (CPM) where k -cliques are explored by rotating about its component $(k-1)$ -cliques. CFinder is too

slow when applied to dense PPI networks, and particularly it cannot detect spoken-like module (noncliques). To overcome this problem, Zhang et al. [21] combines the Line Graph Transformation (LGT) and CPM to detect overlapping network modules and builds the overlapping modules network. Wu et al. [22] proposes COACH (core-attachment based method) to predict complexes by detecting protein-complex core and then adding attachments. The Local Protein Community Finder [23], LPCF for short, uses two local clustering algorithms to find a community close to a queried protein. (2) Global cluster methods. (NeMo) [14] combine a neighbour-sharing score with hierarchical agglomerative clustering to identify both dense network and dense bipartite network structures in a single approach. Reichardt and Bornholdt [24] propose a method to detect overlapping (fuzzy) communities that maps the graph onto a zero-temperature q -Potts model with nearest-neighbor interactions. Zhang et al. [25] combine the idea of modularity function Q , spectral relaxation, and fuzzy c -means clustering method for detecting overlapping community structure. Wang et al. [26] propose a BCD (Betweenness-Commonality Decomposition) algorithm which uses edge commonality and edge-betweenness. Other methods such as nonnegative matrix factorization (NMF) technique were also used for uncovering overlapping (fuzzy) communities [27, 28]. Besides these topological-based methods, Chen and Yuan [29] integrate 265 microarray datasets to detect functional modules in yeast protein-protein interaction network.

Here we describe MOfinder, an alternative method we have developed that can effectively identify functional modules, especially overlapping modules, from a PPI network. MOfinder allows flexibility and user customization with adjustable parameters. We compared the performance of MOfinder with other available methods. We explored the overlapping information of modules in yeast and human PPI network. We used all the overlapping modules detected from human PPI network to generate a graph of module-module communication, and we analyzed the functional properties of the overlapping modules.

2. Materials and Methods

2.1. Data Sources. The human PPI data sets were downloaded from HPRD (release 8) [30]. The yeast PPI data sets were collected from DIP [31]. Cancer Genes [32] (“Tumor Suppressor” and “Oncogene”) and Immunome [33, 34] were used to annotate cancer- and immune-related proteins.

2.2. Definition of Clustering Coefficient. Clustering coefficient of node n is defined as $CC(n) = 2S_n/K_n(K_n - 1)$, where K_n is the degree of n and S_n is the number of connected links between all neighbors of n .

2.3. Definition of Functional Module. Given a predicted module, the P -value of it with respect to a GO term is computed by the hypergeometric distribution in (1) and corrected by Bonferroni correction. The functional module is defined as

a module enriched in at least one GO term (Bonferroni P -value < 0.01):

$$P\text{-value} = \frac{\binom{m}{k} \binom{N-m}{n-k}}{\binom{N}{n}}, \quad (1)$$

where a predicted complex with size m , k proteins share a GO term, and in a total of N proteins, n of them have the same GO term.

2.4. Functional Similarity of Modules. Assuming $GO_1 = \{go_{11}, go_{12}, \dots, go_{1m}\}$ and $GO_2 = \{go_{21}, go_{22}, \dots, go_{2n}\}$ are two sets of GO terms that annotate modules A and B , respectively, the following Jaccard index was used to calculate the functional similarity between modules A and B :

$$\text{Sim}(A, B) = \frac{|GO_1 \cap GO_2|}{|GO_1 \cup GO_2|}. \quad (2)$$

3. Results

3.1. MOfinder Algorithm. MOfinder is based on an AMD (Approximate Minimum Degree Ordering) algorithm [35, 36] which has been used for network clustering from electrical engineering [37]. AMD algorithm is usually used in ordering a sparse matrix prior to Cholesky factorization (or for LU factorization with diagonal pivoting), and it can transform the sparse matrix to make the nonzero elements close to the diagonal. The approach used by MOfinder is summarized in Figure 1. MOfinder first converts the PPI file into a sparse matrix, where a nonzero element represents a protein-protein interaction. It then performs a global AMD of the sparse matrix in which the densely connected elements (module) will be clustered along the diagonal. Besides the global AMD, which produces the global ordering, a local AMD is performed to give the approximate minimum degree ordering. MOfinder uses a sliding window along the diagonal to fetch the local sparse matrix and make the local AMD. The clustering coefficient (CC) [38] value of the submatrix in the sliding window is calculated; if the CC value is not less than the cut-off, MOfinder will save the submatrix as a module. Then the sliding window moves one step along the diagonal to find new modules, and the iteration process is repeated until the sliding window reaches the end. Lastly, MOfinder removes redundant modules (if module A belongs to module B , A is removed) and saves results. The pseudocode of MOfinder algorithm is (see Algorithm 1).

3.2. MOfinder Is a Flexible Method. MOfinder contains two adjustable parameters: the CC cut-off value and the size of sliding window. Different parameters will vary the results. To optimize the parameters, the performance was assessed in term of accuracy of identified modules with respect to annotated function. MOfinder was tested over a broad range of parameters for CC cut-off value (0.2–1) and sliding window (20–450) using PPI data from yeast and human.

First, the percentage of functional modules was plotted against a range of CC cut-off values, and for each CC cut-off value, all sizes of sliding window (20–450, step = 10) were

```

Input: The PPI network  $G = (V, E)$ ;
Cluster coefficient threshold  $CC$ ;
Sliding window  $s$ .
Output: modules.
(1)  $M = \text{zeros}(V, V)$ ; // initialization of network matrix
(2) for each  $e \in E$  do
(3)    $M(e1, e2) = 1$ ; // build sparse matrix,  $e1 \in V, e2 \in V$ 
(4) end for
(5)  $Mg = \text{global AMD}(M)$ ;
(6) While  $a + s \leq V$  do //  $a = 1$ , as start
(7)    $mg = Mg(a : a + s, a : a + s)$ ;
(8)    $ml = \text{local AMD}(mg)$ ;
(9)   if  $cc(ml) \geq CC$  do
(10)     insert  $ml$  to modules;
(11)   end if
(12)    $a = a + 1$ ;
(13) end while
(14) for each module  $A$  in modules do
(15)   if module  $A \in$  module  $B$  do // module  $B$  in modules and module  $A \neq$  module  $B$ 
(16)     discard module  $A$  from modules; // delete redundancy
(17)   end if
(18) end for

```

ALGORITHM 1

TABLE 1: Statistics for the major module size and other features when applied six methods to the yeast PPI data.

Methods	Major module Size	Predicted modules	Covered proteins	Functional percentage
MCODE	3	21	64	85.7%
CFinder	4	53	184	81.1%
COACH	3	382	861	46.6%
NeMo	4	121	485	38.0%
LPCF	10	1601	4549	49.5%
MOfinder	5	125	335	90.4%

tested and the resulting percentages of functional modules were plotted as a group of points. As shown in Figure 2, the percentage of functional modules increases with the increase of CC cut-off value, and it is observed to have 4 distinct and stable ranges for values of CC cut-off, $[0.2, 0.5)$, $[0.5, 0.67)$, $[0.67, 0.84)$, and $[0.84, 1]$, respectively. Although the highest percentage of functional modules is achieved in the last range (CC cut-off value $\in [0.84, 1]$), using CC cut-off value of this range will identify densely connected complex and ignore other modules. Additionally in this range, MOfinder only generates a small number of modules (e.g., it predicts, on average, 36 modules from human PPI network when CC cut-off = 0.84). Since the purpose is to detect modules instead of complex, we recommended that the suitable setting of threshold would be in the third range (CC cut-off value $\in [0.67, 0.84)$). The best choice for CC cut-off value is 0.67 because the number of predicted modules decrease with CC cut-off value (data not shown).

Second, we investigate how the variation of sliding window affects the performance. Figure 3 shows the number of functional modules matched for the 0.67 cut-off value over all tried sizes of sliding window (20–450, step = 10).

The curve of the resulting number of functional modules first increases and then decreases. So the sliding window should be set to 350 which maximized the number of functional modules. To achieve best performance, we recommended that the parameter set was CC cut-off value = 0.67 and size of sliding window = 350.

3.3. Performance Evaluation. MOfinder was tested using PPI data from yeast and human and compared with the performance of other five software available algorithms: MCODE (default parameters), CFinder ($k = 4$, as suggested), COACH (default parameters) NeMo (default parameters), and LPCF (community size was set to 3–11 which was comparable to MOfinder). The percentage of functional modules was used to indicate accuracy, and MOfinder was the top performing algorithm with respect to accuracy in yeast (93.9%) (Figure 4(a)) and human (81.5%) (Figure 4(b)). Also, we compared the major module size of six methods in yeast (Table 1) and in human (See supplementary Table 1 in Supplementary Material available online at doi:10.1155/2011/103702). Most of the modules detected by MCODE are of size 3, size 4 for CFinder, size 3 for COACH,

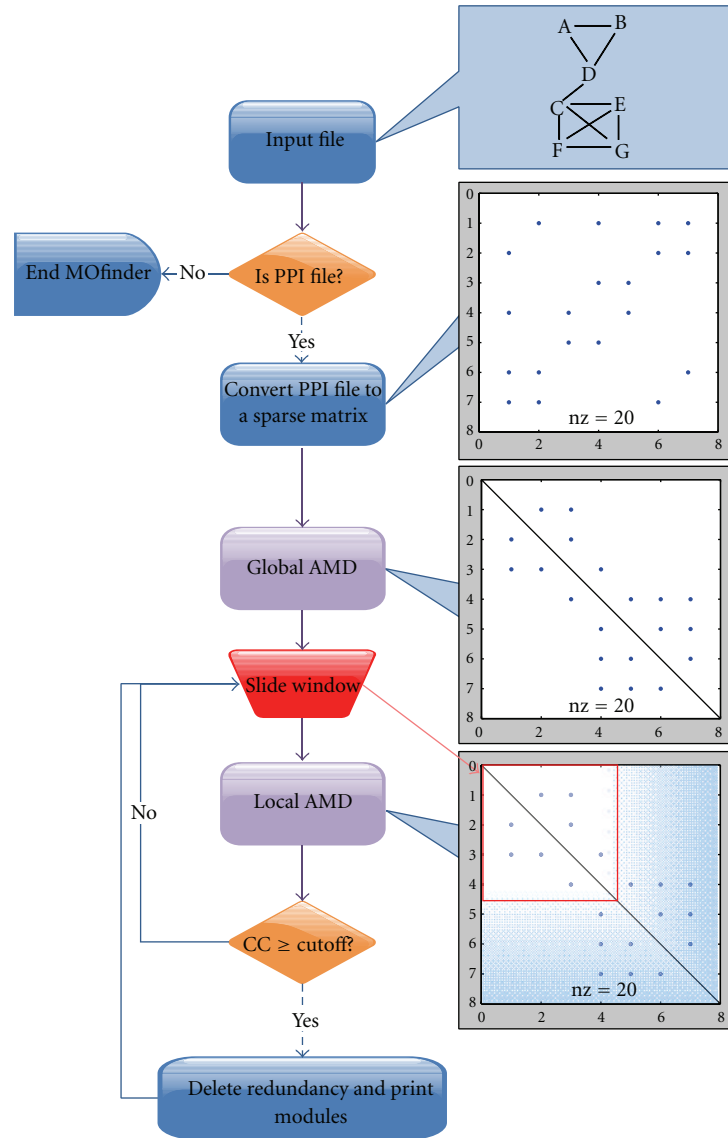


FIGURE 1: The flowchart of the MOfinder. MOfinder first converts the PPI file into a sparse matrix and then performs the global AMD. Next, the MOfinder fetches the local submatrix using a sliding window and also performs the local AMD. The CC value of the submatrix is calculated. At last, submatrixes with CC value $<$ cut-off are filtered and redundant modules are removed; others are saved as modules.

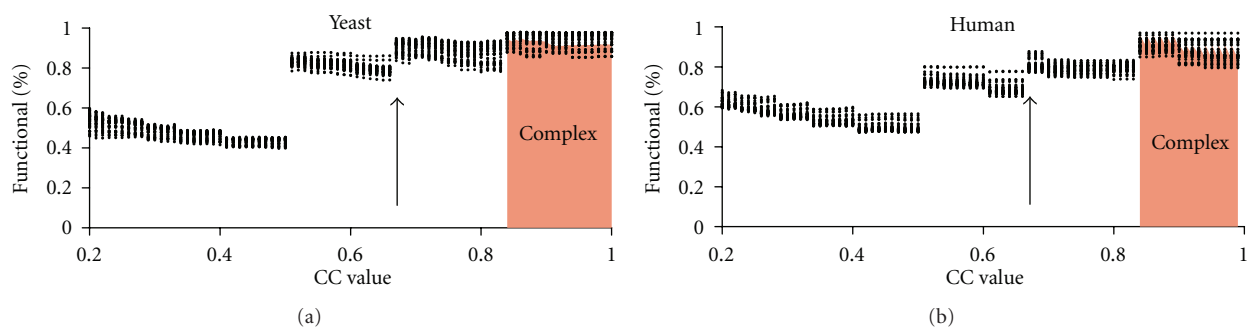


FIGURE 2: Performance of MOfinder in terms of various cut-off values for yeast and human network. Each point represents the functional percentage of modules observed for a cut-off value and a slide window. (a) Yeast PPI network. (b) Human PPI network.

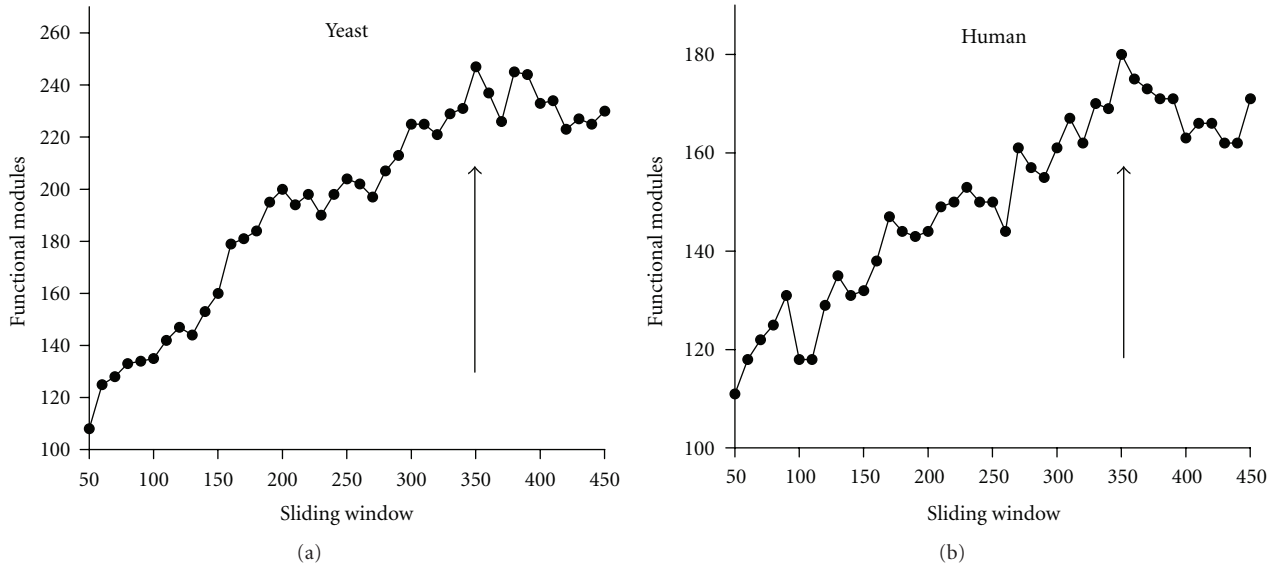


FIGURE 3: Performance of MOfinder in terms of various sliding window sizes for yeast and human network. The y-axis represents the number of functional modules. The threshold of 350 gives the best performance which can detect the maximum functional modules. (a) Yeast PPI network. (b) Human PPI network.

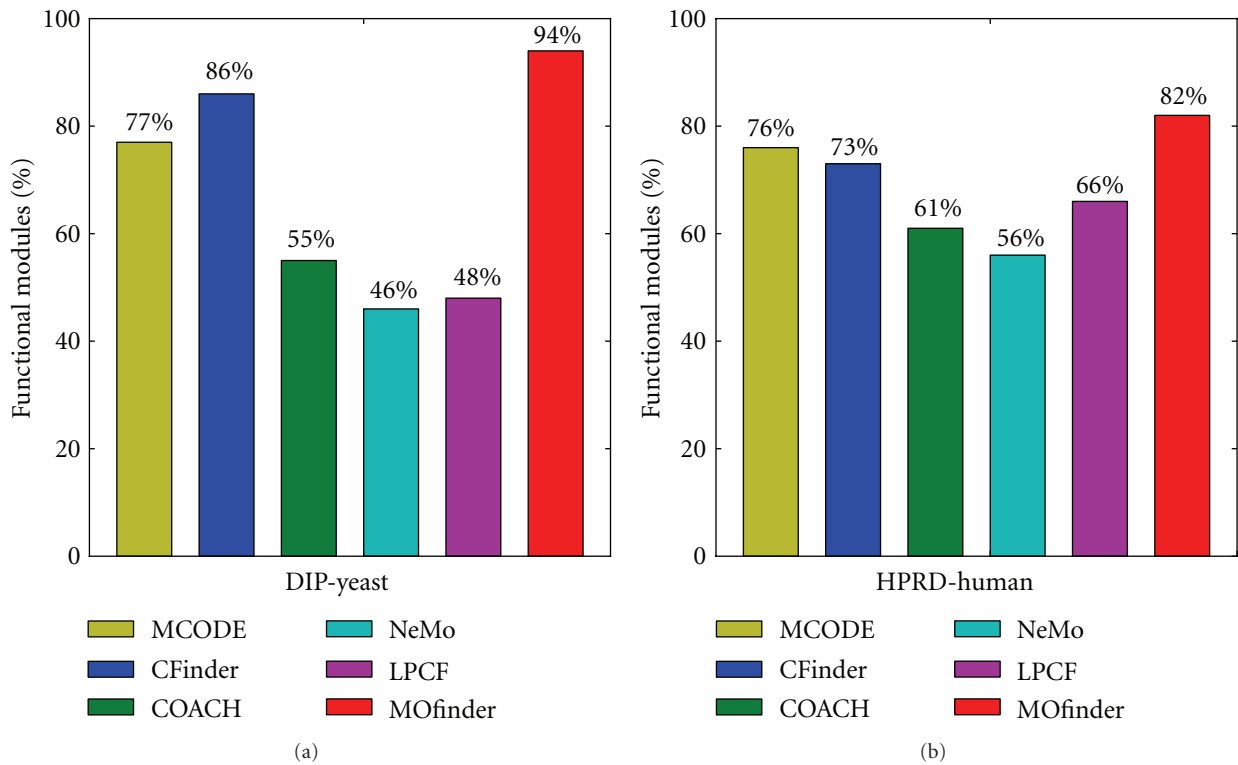


FIGURE 4: Comparative performances of MOfinder and the other five methods. The y-axis represents the proportion of functional modules of detected modules, according to GO annotation (Bonferroni P -value < 0.01).

size 4 for NeMo, size 10 for LPCF, and size 5 for MOfinder. Although the number of modules and the number of proteins assigned to modules were smaller for MOfinder than some of these methods, the percentage of functional modules was highest for MOfinder.

3.4. Overall Overlapping Properties in Yeast and Human. We applied MOfinder to the yeast and human PPI network with default parameters (CC cut-off = 0.67, sliding window = 350). Then we explored the distribution of overlapping size. As shown in Figure 5, the overlapping size distribution is

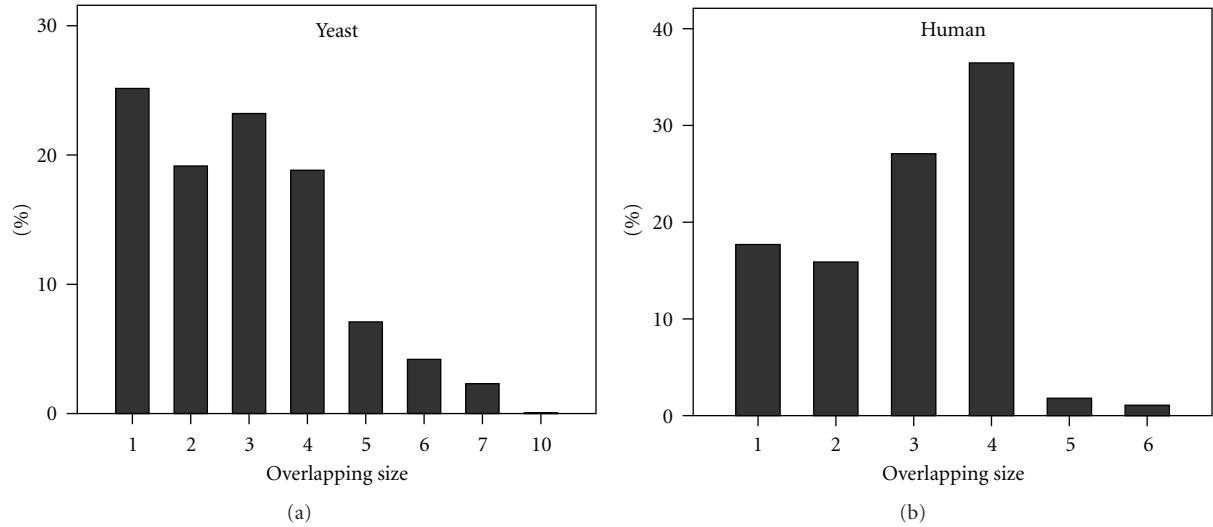


FIGURE 5: Overlapping size distribution of yeast and human PPI network. The x -axis represents the overlapping size. The y -axis is the percentage of each size. (a) Yeast PPI network. (b) Human PPI network.

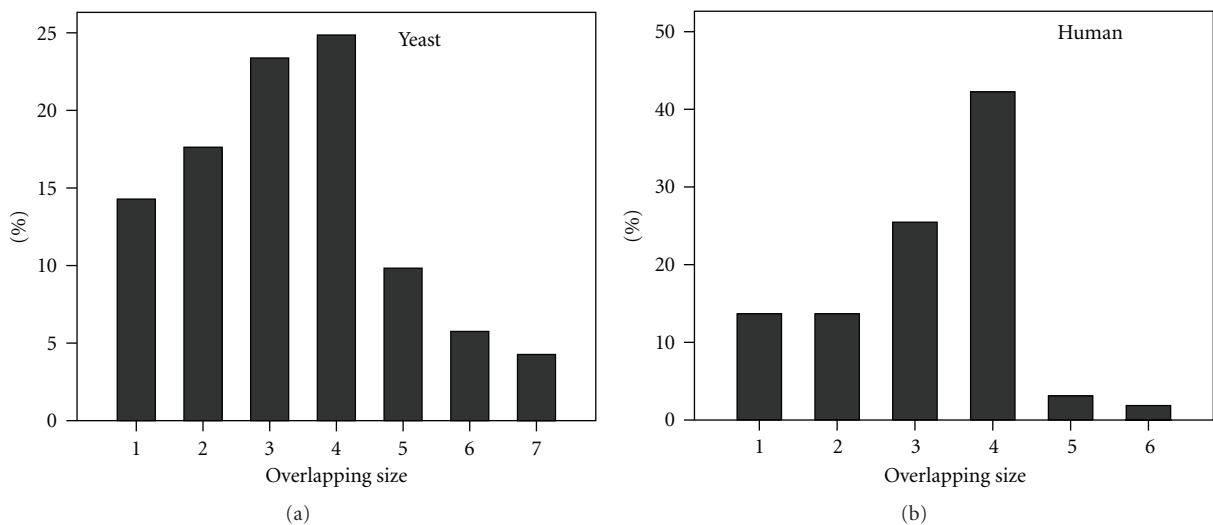


FIGURE 6: Overlapping size distribution of yeast and human PPI network after removing repeats. The x -axis represents the overlapping size. The y -axis is the percentage of each size. (a) Yeast PPI network. (b) Human PPI network.

different between yeast and human. Most of the modules in yeast PPI network share one protein (Figure 5(a)), but in human PPI network the most common overlapping size is 4 (Figure 5(b)). Some overlapping parts might be overcounted. For example, three modules (A , B , and C) share a protein D , so protein D is counted 3 times (A - B , A - C , B - C). To avoid the overcount problem, we deleted the repeats, so protein D is only counted once. Figure 6(a) shows that the resulting distribution of overlapping size in yeast is obviously changed, and the most common overlapping size changes into 4 which is similar to human (Figure 6(b)). These observations suggest that although modules in yeast tend to share less proteins than modules in human, the small overlapping parts (size 1 and size 2) are more repeatedly used in yeast than human,

and thus the distribution of overlapping size becomes similar in yeast and human after removing repeats.

Since proteins in one module work together to perform functions, a similar function is expected to appear if two modules are overlapping with each other. And the larger the overlapping size, the more likely the same function. To verify this, we used the GO annotation similarity to represent the functional similarity. Figure 7 shows that the average functional similarity is increased with the increase of overlapping size. Such a trend has been observed in both yeast (Figure 7(a)) and human (Figure 7(b)).

3.5. Overlapping Modules in the Human Interactome. MO-finder identified 221 modules, of which 152 were overlapped

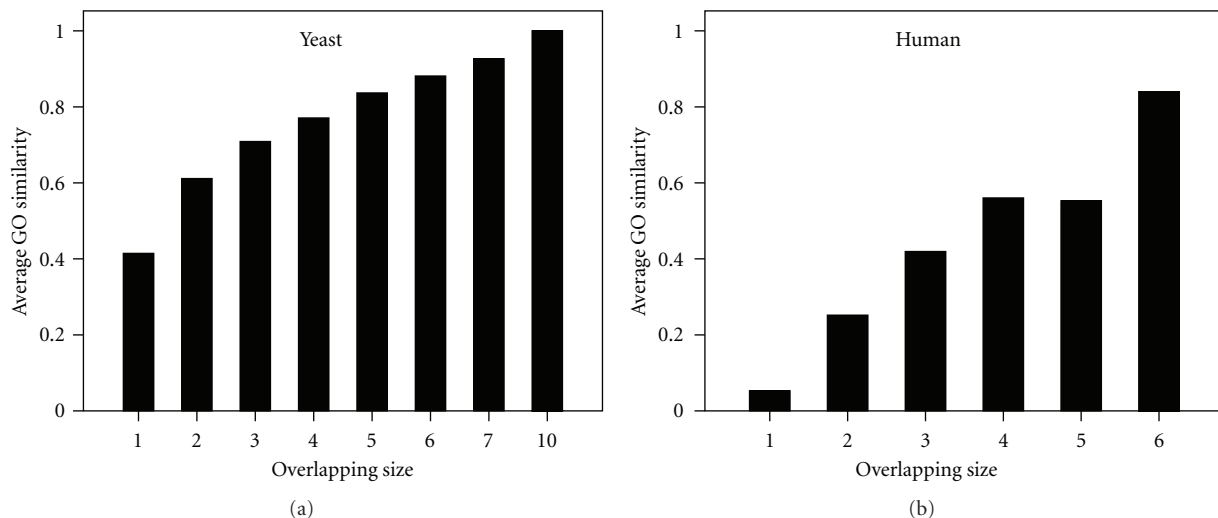


FIGURE 7: Modules with larger overlapping size confer more similar GO annotation. The x -axis represents the overlapping size. The y -axis is the mean value of GO similarity. (a) Yeast PPI network. (b) Human PPI network.

with at least one other module. These overlapped modules were used to construct a module-module communication network (Figure 8(a)). In the communication network, each node is a module, two modules being connected if they share at least one protein. To explore the functional of this network, we used DAVID 6.7 [40, 41] to search for enrichment of Gene Ontology (GO) terms and the KEGG pathways. We found that GO terms and pathways related to cancer and immune response were enriched in the network proteins, so we mapped the cancer and immune-related proteins to the modules. As shown in Figure 8(a), of the 47 modules containing immune-related proteins, 33 included cancer-related proteins, and the ratio (33/47) was greater than expected by chance (62 of 152 modules have cancer-related proteins, Binomial test, $P < 0.01$). Therefore, the modules containing immune-related proteins always included cancer-related proteins and vice versa (33/62 was greater than expected 47/152, Binomial test, $P < 0.01$).

To explore the communication between functional modules, we map the functional annotation to each module and evaluate the functional similarity between two overlapping modules. The functional similarity is shown as edge color in Figure 8: the values between 0 and 1 are painted with a pink/blue color gradient, and modules without GO annotation have gray edges. Figure 8(b) gives the functional annotation of modules from the largest cluster in Figure 8(a). Some overlapping modules have the same function, such as the three modules involve in the acetylation of peptidyl-lysine, while several overlapping modules have distinct function, for instance, a module involved in the change of mast cell is overlapping with another module which takes part in the reactions mediated by protein kinases. Figure 9 shows an example of two overlapping modules. One module function is in B-cell activation processes and it contains five proteins: Q15464, O75791, O43561, Q13094, and P08575. The other module (P08575, P20963, P06729, and P06127) involves in T-cell activation. These two Modules share a protein: P08575

(receptor-type tyrosine-protein phosphatase C, CD45), which plays a critical role in receptor-mediated signalling in both B and T-cells [42, 43]. The shared node between two modules suggests a pathway crosstalk between them. Consistent with this hypothesis, several studies have illustrated T-cell-dependent B-cell activation [44].

The module-module communication network included 341 overlapping nodes (nodes belonging to two or more modules). Several studies showed that modular overlaps are potential drug targets because they are key determinants of cooperation between network modules [45]. So we investigated the potential druggability of overlapping nodes: 56 of them were established drug targets and another 43 proteins were from druggable family [46], which were 99 druggable proteins in all. The ratio of druggable proteins (99/341) was significantly higher than expected (2000–3000 druggable proteins in human [46], Binomial test, $P < 0.01$).

4. Discussion

For both yeast and human interactomes, MOfinder surpasses the other five methods in accuracy. Furthermore, MOfinder is fast in practice for large networks. For example, when applied to a yeast network including nearly 40,000 interactions (from I2D [47]), the running time of MOfinder was only 15 seconds. Since the size of biological networks continues to grow, MOfinder is likely to meet the needs of biological analysis. However, MOfinder has two possible limitations. One is that MOfinder specifically detects small-sized modules (less than 12), but the major module size (5) is close to the average size of MIPS complexes (6) [20]. MOfinder detects 125 modules from the yeast PPI network, which is less than COACH and LPCF. From the perspective of the covered proteins of predicted modules, MOfinder is rank 4. These observations suggest another limitation: MOfinder is too strict to detect loosely connected modules, partly because the CC cut-off value is set to 0.67. We suppose that setting the

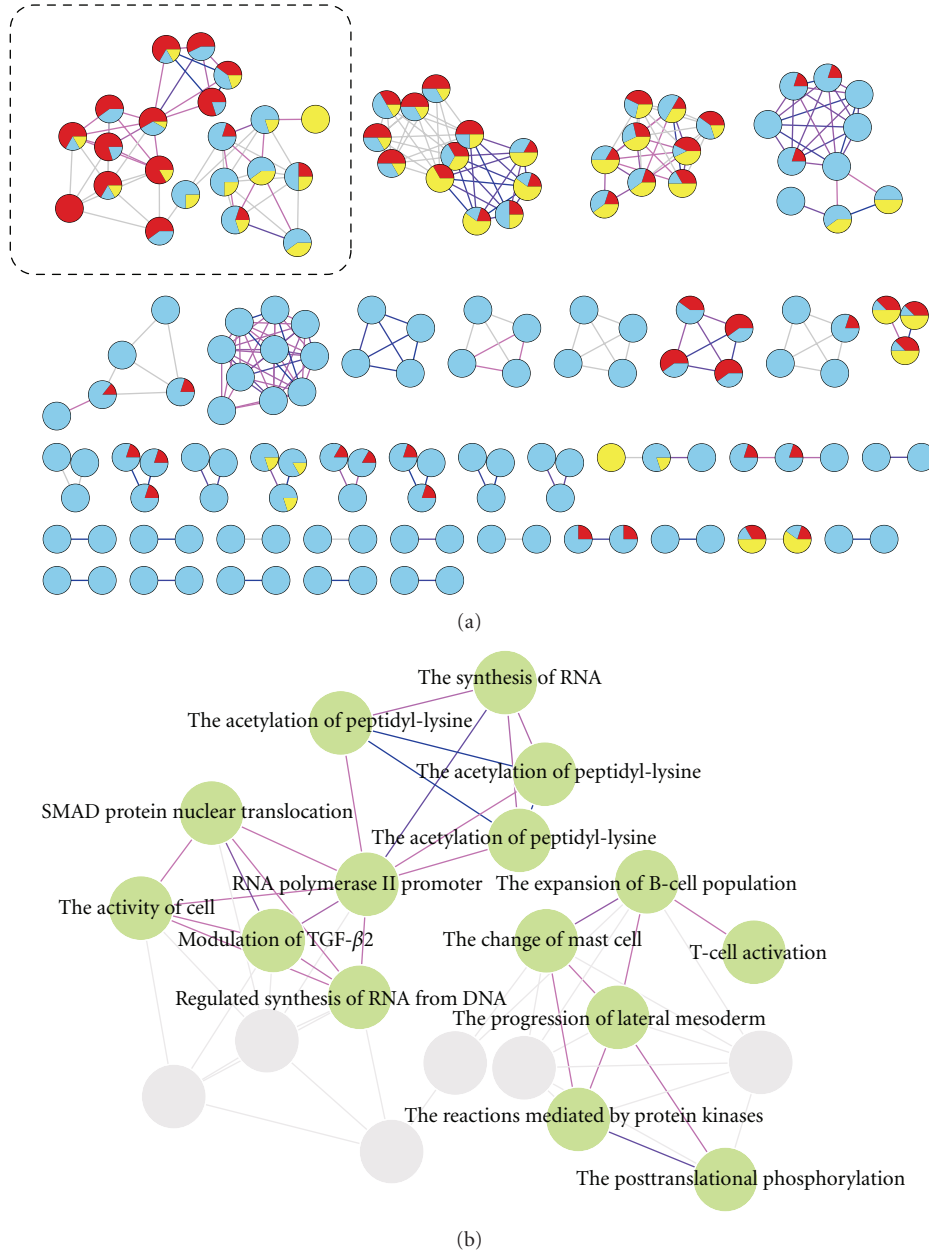


FIGURE 8: The module-module communication network and functional characterization. The network is built by GenePro 2.5.2b [39]. Each node represents a module, and the edge between two nodes means these two modules share at least one protein. Edge color represents the GO similarity: the values between 0 and 1 are painted with a pink/blue color gradient, and gray means lack of GO annotation. (a) The module module communication network. In the pie plot of each node, the color means function: red is cancer related, yellow is immune related, and blue is other function. The size of sector means the percentage of each type of functional proteins. (b) Functional characterization of each module in the largest cluster (rounded by rectangle of dotted line). Modules are painted green (if there is a GO annotation) or gray (lack of GO annotation).

CC cut-off value to a small value can increase the number of detected modules especially loosely connected modules (including pathways). But what is the biological significance of the different clustering coefficient thresholds is still an open question.

Yeast is a simple single-celled eukaryote, so the overlapping modules in yeast generally use one protein for communication. On the contrary, human, a multicellular organism,

employs more complex system, and thus the overlapping size of human is larger than that of yeast. We also found the overall distribution of overlapping size is similar between yeast and human after removing repeats. And in Figure 2 the functional steps occur at similar places between yeast and human. These observations reflect the evolutionary conservation across eukaryotes. Although overlaps may lead to redundant modules which overlap with each other heavily, excluding

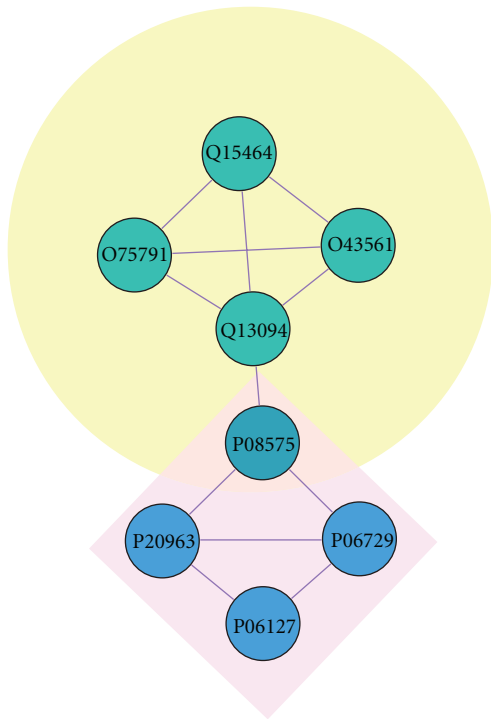


FIGURE 9: Two overlapping modules shared one protein P08575. The yellow module works on T cell activation and the pink module take part in B cell activation.

the overlapping size 4 (a heavily overlap because the major module size is 5) from Figures 5, 6, and 7 does not change the overall pattern of results.

Overlapping modules will work together to carry out several complicated jobs, such as signal transduction. So constructing a module-module communication network to explore how these modules communicate with each other can help to understand biological complexity. Although we just built such a network in human, similar approach can be applied to other species. We found that the immune- and cancer-related proteins are always in the same modules. The association between immune cells and cancer has been discussed [48], and several clinical studies and experiment have proven that the immune system is a new weapon against cancer [49]. Antitumor adaptive immune responses can suppress tumor growth [50], and several immunotherapy drugs could cure cancers [51]. We provided the evidence for their close relationship on the system level.

5. Conclusions

In this paper, we describe a novel algorithm for the identification of overlapping modules in PPI networks. MOfinder performs competitively with other methods and uses two adjustable parameters that enable it to identify modules flexibly. MOfinder is a cross-platform package which is implemented as a C/C++ script, and it can be downloaded and installed free of charge (<http://bsb.kiz.ac.cn/mofinder/>). The application of MOfinder to human PPI gives clues for

fighting against cancer using immune system. And the overlapping nodes, which are in charge of intermodule cross-talk, could help to identify potential drug targets.

Acknowledgment

This work was supported by the National Basic Research Program of China (Grant no. 2009CB941300) for J.-F. Huang. Q. Yu and G. H. Li contributed equally to this work.

References

- [1] A. C. Gavin, M. Bösch, R. Krause et al., "Functional organization of the yeast proteome by systematic analysis of protein complexes," *Nature*, vol. 415, no. 6868, pp. 141–147, 2002.
- [2] J. F. Rual, K. Venkatesan, T. Hao et al., "Towards a proteome-scale map of the human protein-protein interaction network," *Nature*, vol. 437, no. 7062, pp. 1173–1178, 2005.
- [3] H. Kitano, "Systems biology: a brief overview," *Science*, vol. 295, no. 5560, pp. 1662–1664, 2002.
- [4] P. Bork, L. J. Jensen, C. Von Mering, A. K. Ramani, I. Lee, and E. M. Marcotte, "Protein interaction networks from yeast to human," *Current Opinion in Structural Biology*, vol. 14, no. 3, pp. 292–299, 2004.
- [5] G. P. Wagner, M. Pavlicev, and J. M. Cheverud, "The road to modularity," *Nature Reviews Genetics*, vol. 8, no. 12, pp. 921–931, 2007.
- [6] R. Sharan, I. Ulitsky, and R. Shamir, "Network-based prediction of protein function," *Molecular Systems Biology*, vol. 3, p. 88, 2007.
- [7] X. Wu, R. Jiang, M. Q. Zhang, and S. Li, "Network-based global inference of human disease genes," *Molecular Systems Biology*, vol. 4, article 189, 2008.
- [8] Y. Chen and N. V. Dokholyan, "The coordinated evolution of yeast proteins is constrained by functional modularity," *Trends in Genetics*, vol. 22, no. 8, pp. 416–419, 2006.
- [9] M. Campillos, C. Von Mering, L. J. Jensen, and P. Bork, "Identification and analysis of evolutionarily cohesive functional modules in protein networks," *Genome Research*, vol. 16, no. 3, pp. 374–382, 2006.
- [10] L. H. Hartwell, J. J. Hopfield, S. Leibler, and A. W. Murray, "From molecular to modular cell biology," *Nature*, vol. 402, no. 6761, pp. C47–C52, 1999.
- [11] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," *Nature*, vol. 435, no. 7043, pp. 814–818, 2005.
- [12] V. Rossio and S. Yoshida, "Spatial regulation of Cdc55-PP2A by Zds1/Zds2 controls mitotic entry and mitotic exit in budding yeast," *Journal of Cell Biology*, vol. 193, no. 3, pp. 445–454, 2011.
- [13] W. W. M. Pim Pijnappel, D. Schaft, A. Roguev et al., "The *S. cerevisiae* SET3 complex includes two histone deacetylases, Hos2 and Hst1, and is a meiotic-specific repressor of the sporulation gene program," *Genes and Development*, vol. 15, no. 22, pp. 2991–3004, 2001.
- [14] C. G. Rivera, R. Vakil, and J. S. Bader, "NeMo: network module identification in cytoscape," *BMC Bioinformatics*, vol. 11, no. 1, article S61, 2010.
- [15] X. Li, M. Wu, C. K. Kwok, and S. K. Ng, "Computational approaches for detecting protein complexes from protein

- interaction networks: a survey,” *BMC Genomics*, vol. 11, no. 1, article S3, 2010.
- [16] S. Brohée and J. van Helden, “Evaluation of clustering algorithms for protein-protein interaction networks,” *BMC Bioinformatics*, vol. 7, article 488, 2006.
- [17] M. E. Sardiú, L. Florens, and M. P. Washburn, “Evaluation of clustering algorithms for protein complex and protein interaction network assembly,” *Journal of Proteome Research*, vol. 8, no. 6, pp. 2944–2952, 2009.
- [18] A. Lancichinetti and S. Fortunato, “Community detection algorithms: a comparative analysis,” *Physical Review E*, vol. 80, no. 5, Article ID 056117, 2009.
- [19] S. Fortunato, “Community detection in graphs,” *Physics Reports*, vol. 486, no. 3–5, pp. 75–174, 2010.
- [20] G. D. Bader and C. W. V. Hogue, “An automated method for finding molecular complexes in large protein interaction networks,” *BMC Bioinformatics*, vol. 4, article 2, 2003.
- [21] S. Zhang, H. W. Liu, X. M. Ning, and X. S. Zhang, “A hybrid graph-theoretic method for mining overlapping functional modules in large sparse protein interaction networks,” *International Journal of Data Mining and Bioinformatics*, vol. 3, no. 1, pp. 68–84, 2009.
- [22] M. Wu, X. Li, C. K. Kwok, and S. K. Ng, “A core-attachment based method to detect protein complexes in PPI networks,” *BMC Bioinformatics*, vol. 10, article 169, 2009.
- [23] K. Voevodski, S. H. Teng, and Y. Xia, “Finding local communities in protein networks,” *BMC Bioinformatics*, vol. 10, article 1471, p. 297, 2009.
- [24] J. Reichardt and S. Bornholdt, “Detecting fuzzy community structures in complex networks with a potts model,” *Physical Review Letters*, vol. 93, no. 21, Article ID 218701, 2004.
- [25] S. Zhang, R. S. Wang, and X. S. Zhang, “Identification of overlapping community structure in complex networks using fuzzy c-means clustering,” *Physica A*, vol. 374, no. 1, pp. 483–490, 2007.
- [26] C. Wang, C. Ding, Q. Yang, and S. R. Holbrook, “Consistent dissection of the protein interaction network by combining global and local metrics,” *Genome Biology*, vol. 8, no. 12, article R271, 2007.
- [27] S. Zhang, R. S. Wang, and X. S. Zhang, “Uncovering fuzzy community structure in complex networks,” *Physical Review E*, vol. 76, no. 4, Article ID 046103, 2007.
- [28] M. Zarei, D. Izadi, and K. A. Samani, “Detecting overlapping community structure of networks based on vertex-vertex correlations,” *Journal of Statistical Mechanics-Theory and Experiment*, vol. 2009, article P11013, 2009.
- [29] J. Chen and B. Yuan, “Detecting functional modules in the yeast protein-protein interaction network,” *Bioinformatics*, vol. 22, no. 18, pp. 2283–2290, 2006.
- [30] T. S. Keshava Prasad, R. Goel, K. Kandasamy et al., “Human protein reference database—2009 update,” *Nucleic Acids Research*, vol. 37, no. 1, pp. D767–D772, 2009.
- [31] L. Salwinski, C. S. Miller, A. J. Smith, F. K. Pettit, J. U. Bowie, and D. Eisenberg, “The database of interacting proteins: 2004 update,” *Nucleic Acids Research*, vol. 32, pp. D449–D451, 2004.
- [32] M. E. Higgins, M. Claremont, J. E. Major, C. Sander, and A. E. Lash, “CancerGenes: a gene selection resource for cancer genome projects,” *Nucleic Acids Research*, vol. 35, no. 1, pp. D721–D726, 2007.
- [33] C. Ortutay, M. Siermala, and M. Vihinen, “Molecular characterization of the immune system: emergence of proteins, processes, and domains,” *Immunogenetics*, vol. 59, no. 5, pp. 333–348, 2007.
- [34] C. Ortutay and M. Vihinen, “Immunome: a reference set of genes and proteins for systems biology of the human immune system,” *Cellular Immunology*, vol. 244, no. 2, pp. 87–89, 2006.
- [35] P. R. Amestoy, T. A. Davis, and I. S. Duff, “An approximate minimum degree ordering algorithm,” *SIAM Journal on Matrix Analysis and Applications*, vol. 17, no. 4, pp. 886–905, 1996.
- [36] P. R. Amestoy, T. A. Davis, and I. S. Duff, “Algorithm 837: AMD, an approximate minimum degree ordering algorithm,” *ACM Transactions on Mathematical Software*, vol. 30, no. 3, pp. 381–388, 2004.
- [37] A. O. M. Saleh and M. A. Laughton, “Cluster-analysis of power system networks for array-processing solutions,” *IEE Proceedings C*, vol. 132, no. 4, pp. 172–178, 1985.
- [38] X. Zhu, M. Gerstein, and M. Snyder, “Getting connected: analysis and principles of biological networks,” *Genes and Development*, vol. 21, no. 9, pp. 1010–1024, 2007.
- [39] J. Vlasblom, S. Wu, S. Pu et al., “GenePro: a cytoscape plugin for advanced visualization and analysis of interaction networks,” *Bioinformatics*, vol. 22, no. 17, pp. 2178–2179, 2006.
- [40] D. W. Huang, B. T. Sherman, and R. A. Lempicki, “Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources,” *Nature Protocols*, vol. 4, no. 1, pp. 44–57, 2009.
- [41] G. Dennis, B. T. Sherman, D. A. Hosack et al., “DAVID: database for annotation, visualization, and integrated discovery,” *Genome Biology*, vol. 4, no. 5, p. P3, 2003.
- [42] L. A. Earl and L. G. Baum, “CD45 glycosylation controls T-cell life and death,” *Immunology and Cell Biology*, vol. 86, no. 7, pp. 608–615, 2008.
- [43] P. Shrivastava, T. Katagiri, M. Ogimoto, K. Mizuno, and H. Yakura, “Dynamic regulation of Src-family kinases by CD45 in B cells,” *Blood*, vol. 103, no. 4, pp. 1425–1432, 2004.
- [44] D. C. Parker, “T cell-dependent B cell activation,” *Annual Review of Immunology*, vol. 11, pp. 331–360, 1993.
- [45] I. J. Farkas, T. Korcsmáros, I. A. Kovács et al., “Network-based tools for the identification of novel drug targets,” *Science Signaling*, vol. 4, no. 173, 2011.
- [46] A. P. Russ and S. Lampel, “The druggable genome: an update,” *Drug Discovery Today*, vol. 10, no. 23–24, pp. 1607–1610, 2005.
- [47] K. R. Brown and I. Jurisica, “Online predicted human interaction database,” *Bioinformatics*, vol. 21, no. 9, pp. 2076–2082, 2005.
- [48] F. Balkwill and A. Mantovani, “Inflammation and cancer: back to virchow?” *The Lancet*, vol. 357, no. 9255, pp. 539–545, 2001.
- [49] O. J. Finn, “Tumor immunology top 10 list,” *Immunological Reviews*, vol. 222, no. 1, pp. 5–8, 2008.
- [50] K. E. De Visser, A. Eichten, and L. M. Coussens, “Paradoxical roles of the immune system during cancer development,” *Nature Reviews Cancer*, vol. 6, no. 1, pp. 24–37, 2006.
- [51] M. A. Mac Cheever, “Twelve immunotherapy drugs that could cure cancers,” *Immunological Reviews*, vol. 222, no. 1, pp. 357–368, 2008.