Identifying protein complexes and functional modules—from static PPI networks to dynamic PPI networks

Bolin Chen, Weiwei Fan, Juan Liu and Fang-Xiang Wu

Submitted: 17th February 2013; Received (in revised form): 1st May 2013

Abstract
Cellular processes are typically carried out by protein complexes and functional modules. Identifying them plays an important role for our attempt to reveal principles of cellular organizations and functions. In this article, we review computational algorithms for identifying protein complexes and/or functional modules from protein–protein interaction (PPI) networks. We first describe issues and pitfalls when interpreting PPI networks. Then based on types of data used and main ideas involved, we briefly describe protein complex and/or functional module identification algorithms in four categories: (i) those based on topological structures of unweighted PPI networks; (ii) those based on characters of weighted PPI networks; (iii) those based on multiple data integrations; and (iv) those based on dynamic PPI networks. The PPI networks are modelled increasingly precise when integrating more types of data, and the study of protein complexes would benefit by shifting from static to dynamic PPI networks.

Keywords: protein complex; functional module; protein–protein interaction; static network; dynamic network

INTRODUCTION
Cellular processes are typically not carried out by individual proteins, but rather by groups of proteins that interact with each other [1–3]. Understanding those groups of proteins is a critical step towards unravelling the intricate molecular relationships within cells [4–6]. Given a set of protein interaction data, a protein–protein interaction (PPI) network can be constructed by taking individual proteins as vertices and pair-wise interactions between them as edges. Large-scale PPI data have provided maps of molecular networks for several organisms [7–10]. Although most of them are incomplete and inaccurate, they reveal important principles of protein organizations within cells.

Generally, two types of protein organizations are commonly studied: protein complexes and functional modules. A protein complex is a group of proteins that interact with each other at the same time and place, forming a single multi-molecular machine [11, 12], while a functional module consists of a group of proteins participating in a specific cellular process, but proteins may interact with each other at a different time and place [4, 11–13]. They have both close relationship and different biological meanings. On the one hand, functional modules often contain one or multiple protein complexes in specific time and space. Therefore, they often exhibit similar characters in PPI networks [13, 14]. On the other hand, they are grouped according to different criteria. Protein complexes are specific molecular entities whose proteins tend to be co-localized and co-expressed [13, 14], whereas functional modules are grouped according to individual cellular processes whose proteins carry out different biological functions within those processes [13].
It is important to distinguish between protein complexes and functional modules, as they are different protein organizations [12–14]. However, owing to the lack of temporal and spatial information for pair-wise protein interaction data, it is not easy to make this distinction. Most computational algorithms can detect sets of proteins grouped as either protein complexes or functional modules. However, they hardly distinguish between them unless other kinds of (typically dynamic) data are further incorporated [12, 14].

In this article, we review computational algorithms for identifying protein complexes and/or functional modules from static PPI networks to dynamic PPI networks. According to types of data used and main ideas involved, all algorithms are organized in four categories. To start with, those based on topological structures of unweighted PPI networks are regarded as the first category. They are often designed to detect sub-graphs with specific topological structures in a PPI network, such as cliques [12, 15–17], dense sub-graphs [18–24], core-attachment structures [25–29] and star-like structures [30]. Although the predictive accuracy is limited, algorithms in this category play fundamental roles in the identification of protein complexes and/or functional modules. The second category consists of algorithms that are based on characters of weighted PPI networks. Numerous topological indices, such as the local neighbourhood density [31], the number of common neighbours [32–34], the edge-betweenness [35–37], the edge-clustering coefficient [38] and the shortest path [39], are used to assign weights to vertices or edges. Various hierarchical clustering approaches are also designed to partition a network into sub-graphs in this category [35–41]. Next, those involving ideas of multiple data integrations contribute to the third category. They use other sources of information, such as the experimental conditions [42, 43], gene expression profiles [44–48] and gene ontology (GO) [49, 50] to assign weights to edges of a PPI network, rather than using those topological indices. Generally, more biological meaningful results can be obtained by integrating more types of data. Finally, algorithms based on dynamic PPI networks are regarded as the fourth category. They also need to integrate multiple types of data, but they model PPI networks as dynamic systems [51–53], which are more reasonable for cellular systems. It is also possible to distinguish between protein complexes and functional modules in this situation.

A recent review article is also proposed by Srihari and Leong [54]. They provide an up-to-date survey, classification and evaluation of most key protein complex identification methods till 2012. The algorithms are organized as a chronology-based ‘bin-and-stack’ and a methodology-based ‘tree’ classification. Open challenges are also discussed in [54] for reconstructing accurate protein complexes. Another recent survey article is proposed by Li et al. [55]. They list majority protein complex-detecting algorithms that have been developed till 2009. The surveyed algorithms are based on static network models, from pair-wise unweighted PPI networks to multiple data integrating. Both of them provide valuable insights for researches that have been done in this area. Differently, we focus on reviewing the related work according to different types of PPI networks in this article. Computational algorithms are organized according to the types of PPI networks being modelled and the main ideas involved in. By this way, one only needs to focus on what kinds of data are being used and accordingly select or design an appropriate algorithm. The organization of the article and the relationship of those computational algorithms are illustrated in Figure 1.

Before giving detail reviews of algorithms for identifying protein complexes and/or functional modules from PPI networks, it is important to make clear about characters of protein interactions and PPI networks, such as what protein interactions are, how PPI networks are established and what kinds of pitfalls should be aware when interpreting PPI networks.

PROTEIN INTERACTIONS AND PPI NETWORKS

Protein interactions occur when two or more proteins bind together in a cell invivo [2]. With advances in high-throughput proteomics technologies, such as yeast two-hybrid (Y2H) assay and affinity purification followed by mass spectrometry (AP/MS), numerous PPI datasets have been produced for many organisms. The availability of those large-scale PPI data has led to the recent popularity of the study in PPI networks [56], especially of those investigating principles of cellular organizations and functions. However, it should be careful when interpreting PPI data, especially to draw biologically relevant conclusions from reported PPI datasets.

The first issue is that most PPI datasets are not complete [56–60]. Various PPI databases are
established by collecting reported PPIs from literature and experiments. However, experimental data reported in literature are only a small fraction of all biologically relevant PPIs. In addition, many databases have to manually collect PPIs from literature, which can obtain an even smaller fraction of the entire PPI space [3]. For example, for yeast PPIs, which are extensively studied, only about 50% of them are reported [56, 57]. For human PPIs, which cover much less, only about 10% of them are reported [56, 57].

The second issue is that reported PPI data are not reliable. This problem arises from both the original experimental methods for identifying those interactions and the subsequent models for interpreting data generated using those experimental methods [61]. Data from both Y2H assay and AP/MS are subjected to high error rates if experiments are not performed under appropriate controls [62]. It is estimated that the reliability of Y2H assay, which reports pair-wise PPI data, does not exceed 50% [11]. The measurements are made under non-physiologic conditions, such that the observed interactions may not be present in the wild-type cells if two proteins are over expressed [62]. Although AP/MS detects PPIs within a native environment, it cannot distinguish whether binding of a prey to the bait is direct or indirect, due to the fact that AP/MS reports co-complex information of PPIs [2, 62]. Hence, an additional algorithm or model is needed to interpret co-complex observations into pair-wise interactions, which may introduce more noise, including both false-positive and false-negative PPIs [63].

The third issue is that reported PPI datasets contain biases as a consequence of differences in the original detected interactions and the following processing methods [56]. On the one hand, some PPIs are more widely to be studied than others. Hence, the reported PPIs are biased towards proteins from particular cellular environments and towards proteins of more ancient, conserved and highly expressed ones [56]. On the other hand, the attempt of circumventing the problem of inaccurate data often makes the issue of biases more serious. Many algorithms [42–50] select only interactions that satisfy

---

**Figure 1:** The organization of computational algorithms for identifying protein complex and/or functional modules.
specific criterions by multiple validation and data integration. However, they introduce new biases into PPI networks because the validated datasets are further subsets of known PPIs [56]. The issue of biases is at least the same problematic as issues of data quality. It can alter the underlying structure of networks in unpredictable ways. As a result, the PPI networks we obtained may drastically be different from the real and complete networks.

Finally, there are several pitfalls associated with the form of pair-wise PPI networks. Firstly, such a PPI network is an integrated network. It is not the same as the real cellular interaction system. Within cells, one kind of protein should have numerous copies, each as a specific molecular entity. Some copies may interact with one group of proteins, while some others interact with other groups. When it comes to a pair-wise PPI network, a vertex of the network represents a collection of all that kind of protein, rather than those individual protein copies [64]. This is the reason why a hub vertex can bind hundreds of ‘proteins’ in a PPI network, while it is impossible in biological cells. Secondly, a PPI network is actually an integration of many sub-networks, including both local protein organizations and global PPI networks, measured under various experimental conditions and cell cycle phases. Those sub-networks are also collected from different experiments that done in various laboratories. As a consequence, the integrated PPI network contains interactions happening in various times and spaces, no matter whether they happen simultaneously or not, or whether they are exclusive or not. Thirdly, it is easier to include data into a database than to clean them out. Because there is currently no simple way to report ‘negative’ interactions [65], the previous inaccurate interactions will affect the quality of entire datasets for a long time. Figure 2 illustrates the general way about how various experimental PPIs are transformed into PPI networks in a database.

Although PPI data in various databases are problematic, biologically relevant conclusions can be drawn with an extra carefulness when interpreting PPI networks. Mackay et al. [61, 65] and Chatr-aryamontri et al. [42] have discussed thoroughly and clearly about PPI data in their TiBS letter and responding articles. Mackay et al. [61] first analyse the reliability of reported PPI data, and they conclude that many reported PPIs might not occur as presented. Chatr-aryamontri et al. [42] later argue that sensible and biologically relevant conclusions can be obtained by integrating various experimental data. Although Mackay et al. do agree with it, they still appeal researchers to pay more attention to pitfalls associated with PPI networks in [65]. However, with the improvement of various PPI databases [66–69], it is believed that results from interpreting PPI networks would become increasingly reliable and thus important in current research of protein science.

IDENTIFYING PROTEIN COMPLEXES BASED ON TOPOLOGICAL STRUCTURES OF UNWEIGHTED PPI NETWORKS

Protein complexes exhibit specific topological structures in PPI networks. Systematical cataloguing all those protein complexes and their interactions within living cells is one of the key topics in post-genomic biomedical research [70]. Although it is difficult to tell their identical structures, various attempts have been made based on topological structures, such as cliques [12, 15–17], dense sub-graphs [18–24], core-attachment structures [25–29] and star-like structures [30], to identify protein complexes from PPI networks. Actually, cliques are special cases of dense subgraphs, while star-like structures are special cases of core-attachment structures. Taking cliques as a particular kind of category is due to the fact that many computational algorithms use cliques as candidates or components of protein complexes. Taking star-like structures as a particular kind of category is due to the fact that many core-attachment based algorithms only identify dense subgraphs as cores, which may miss plenty of predictions that exhibit star-like structures.

Firstly, cliques are often used as candidates or components of protein complexes. To start with, Spirin and Mirny [12] propose an iterative algorithm to enumerate all cliques in a network. Starting from cliques of size $n$, one can enumerate all cliques of size $n + 1$ by checking each adjacent vertex of previous cliques. If there is no vertex can be added to form a larger clique, one can also obtain a maximal clique simultaneously. It does not take too long time to enumerate all cliques in a PPI network because most of them are very sparse. However, simply using those cliques as candidates of protein complexes does not obtain a high accuracy. Alternatively, Li et al. [15] design the LCMA (Local
Clique Merging Algorithm) to detect protein complexes by using cliques. They first locate local cliques in the network for each vertex, and then merge overlapped cliques as predictions of protein complexes according to their affinity to form maximal dense subgraphs. Moreover, cliques can be used to construct a new graph for purposes of protein complex identification. One example based on this idea is CFinder [16, 17]. It first detects all $k$-cliques in a PPI network. Then, based on the definition that two $k$-cliques are accessible if they share $k-1$ vertices, a $k$-clique accessibility graph can be constructed by

Figure 2: Experimental protein interactions and a combined PPI network. (A) A protein-coding gene in DNA. (B) Some protein copies that coded from the same gene. (C) Real protein complexes that involve that protein copies. (D) Pair-wise PPIs that were obtained from experimental analysis of protein complexes. (E) Protein complexes in other experimental conditions or cell cycle phases. (F) PPI sub-networks that were obtained by various experiments or done by different laboratories. Each set of data represents a local structure of the PPI network. (G) PPI dataset in databases, which are established by collecting reported PPIs from literature and experiments.
taking individual $k$-cliques as vertices and the accessible relationships as edges. The connected components of the accessibility graph are then used to generate overlapping protein complexes, which are unions of all $k$-cliques that can be reached from each other through a series of adjacent $k$-cliques. Results of the CFinder are highly correlated to the value of the parameter $k$. Larger values of $k$ tend to reduce the number of adjacencies, and therefore may result in smaller protein complexes in the network.

Secondly, many algorithms [18–24] are designed to detect dense sub-graphs as candidates of protein complexes. This is due to the fact that proteins tend to exhibit strong interactions within a complex and weak interactions to proteins outside the complex [71]. However, there has not been a generally accepted quantitative definition for dense sub-graphs. They usually are described as sets of vertices within a network such that the connections between those vertices are denser than connections to the rest of the network [38]. Various cost-based methods [18–21], stochastic approaches [22, 23] and line graph-based algorithm [24] are developed to identify dense sub-graphs in PPI networks.

In the first instance, cost-based methods usually define specific cost functions to calculate the cost of a partition in PPI networks. Local dense sub-graphs are obtained by optimizing those costs. The RNSC (Restricted Neighbourhood Search Clustering) algorithm [18] is one of such methods. The cost function is calculated according to the number of invalid connections for each vertex. The algorithm starts from a random user-specified partition, and iteratively moves a vertex from one cluster to an adjacent cluster to decrease the total cost. It ends up with a partitioning of the network if some moves have been reached without decreasing the cost function. The output clusters are filtered according to criteria, such as the cluster size, the cluster density and the functional homogeneity. Cho et al. [19, 20] propose an entropy-based graph clustering algorithm that assigns a cost for each cluster. The vertex entropy is defined according to the connectivity of that vertex. The graph entropy is calculated by summing all vertex entropy in a graph. It is also a seed-growth style algorithm. Starting from a random seed vertex and its neighbours, the algorithm iteratively removes and adds vertices on the boundary of the cluster to minimize the graph entropy. The process of seed selection and optimal cluster generation is repeatedly performed for all candidate seeds. For such seed-growth style algorithms, Chen et al. [21] suggest using cliques as initial seeds, rather than individual vertices. The entropy-based algorithm is used as an example to show how clique seeds can be used to increase the predictive accuracy of protein complex identification [21].

In the second instance, stochastic approaches handle the problem of dense sub-graph identification from a statistic point of view. One of such algorithm is called MCL (Markov CLustering) [22, 23]. It works by simulating random flows in a graph. The process takes a stochastic matrix as input, which represents the transition probabilities between all pairs of nodes. The self-loop of each vertex is added initially, and the loop weight is assigned as the maximum weight of all edges connected to the vertex. It changes the values of the transition matrix at each step according to the previous one until a stochastic condition is satisfied. Two processes—expansion and inflation—are interactively involved during the simulation. The expansion takes the $e^{th}$ power of the stochastic matrix, while the inflation promotes the dense clusters and weakens the sparse clusters. Because greater path lengths are more common within clusters than between different clusters, the expected behaviour of random flows results in community structures of the original network. In practice, the MCL algorithm converges very fast, and it is highly scalable in terms of predicting protein complexes from PPI networks.

In the third instance, the line graph-based approach [24] gives another way to identify dense sub-graphs from PPI networks. It first transforms a PPI network into its line graph, and then applies the MCL algorithm on this new graph. The procedure of this transformation brings a number of advantages for graph clustering. First, it dose not sacrifice any information of the original graph. Second, it amplifies the higher-order local neighbourhood of connections. Third, it is more highly structured than the original graph. The algorithm can produce overlapping sub-graphs in PPI networks.

Thirdly, core-attachment structures are commonly used to identify protein complexes from PPI networks. Gavin et al. [72] have demonstrated that a protein complex should generally contain a core and attachments. A core in a protein complex is formed by a constant set of proteins, which are highly co-expressed and share high functional similarity [11, 25]. The attachments surrounding
the protein complex core assist in performing subordinate functions [25]. This property is also supported by other high-throughput protein data [73].

In terms of identification algorithms, Leung et al. [26] propose a method to identify cores and attachments of protein complexes separately. They use a p-score to evaluate how likely a potential core would be the core component of a complex, according to the number of interactions between the potential core and the rest of the networks. Then neighbours that have interactions with the majority of the core are added to form a protein complex. Wu et al. propose the COACH (core-attachment-based method) in [27]. They identify cores based on the neighbourhood graphs of vertices, and then adding attachments into these cores to form candidate clusters. Biologically meaningful clusters are then selected as final predictions. Other algorithms based on similar heuristic can be found in [25, 28, 29].

Last but not least, star-like structures are recently proposed to identify protein complexes from PPI networks. Chen et al. [30] investigate topological structures of known protein complexes in a Saccharomyces cerevisiae PPI network. They find that many protein complexes exhibit star-like structures. That is, proteins within individual complexes tend to have interactions with only one or a few hub-proteins, while most proteins do not interact with each other. A random-star algorithm is also proposed to identify star-like structures in PPI networks [30].

IDENTIFYING PROTEIN COMPLEXES BASED ON CHARACTERS OF WEIGHTED PPI NETWORKS

Interpreting large-scale PPI data is a challenging task because of the widespread of false positive (FP) [74]. To minimize the effect of those inaccurate data, various weighted strategies are used for identifying protein complexes in PPI networks. Although it is hard to assess the reliability of a single edge weight, Nepusz et al. [75] argue that taking into account network weights globally can greatly improve the detection of protein complexes. Therefore, weights should be used when available. Both weights of vertices and edges can be assigned to increase the reliability.

Bader and Hogue [31] propose the MCODE approach based on a strategy of weighted vertices. The algorithm is made up of three steps. At first, the weight of each vertex \( v \) is assigned based on the local neighbourhood density, which is defined from the density of the highest local \( k \)-core of \( v \) and the value of \( k \). Then, starting from the vertex with the highest weight, a cluster is obtained by recursively including neighbour vertices whose weights are above a given threshold. Finally, the algorithm iteratively removes one-degree vertices to form a cluster in the ‘haircut’ process and adds connected vertices to the cluster if the neighbourhood density exceeds a given threshold in the ‘fluff’ process.

Hwang et al. [76] develop a weight strategy for PPI networks in a different way. They propose an STM (signal transduction model) for PPI networks, and demonstrate the signal transduction behaviour of the perturbation by each vertex on a PPI network statistically. For each vertex \( v \), the signal between \( v \) and \( w \) is modelled by using the Erland distribution, where \( w \) is any vertex in the network except \( v \). Preliminary clusters in the network are formed by using this weighted relationship among all vertices, and the predicted clusters are generated by a merging process. It allows overlapping of output clusters and can identify clusters with a large size, arbitrary shape and low density. However, unexpected huge clusters may also be generated in the post-process of merging.

The number of common neighbours between two vertices is a kind of widely used information to assign weights to edges. Altaf-Ul-Amin et al. [32] design the DPCLus algorithm to assign weights to edges in this way. Then the weight of a vertex is assigned by summing weights of edges that are incident to it. A seed-growth style strategy is developed to generate clusters according to the edge weight and the vertex weight. Li et al. further modify the DPCLus algorithm and propose the IPCA in [33]. The rationale behind this algorithm is that most complexes have a very small diameter and a very small average vertex distance. They use the same process to assign weights to edges and vertices, but generate clusters based on a new criterion. The DPCLus identifies clusters that satisfy a density condition and certain cluster connectivity property, while the IPCA generates clusters that have a small diameter and satisfy a different cluster connectivity-density property. Once a cluster is identified the DPCLus removes the cluster and recalculates the vertex weights based on the new remaining graph while the IPCA computes the vertex weights based on the original graph only once. Kim and Tan [34] propose the miPALM
(module inference by Parametric Local Modularity) that combines the parametric local modularity measure and the greedy search strategy to identify communities in PPI networks. It first assigns weight to edges by using the number of common neighbours and vertex degrees. Each triangle of the weighted network is ranked according to the parametric local modularity and expanded to candidate complexes by a recursive greedy search. Additional parameters are used to control the background neighbourhood size around candidate complexes and to filter unreasonable results.

Statistic approaches are also involved by comparing the known PPI network with a random network of the same size. Samanta and Liang [74] rank the statistical significance of forming shared partnerships for all protein pairs in a PPI network and find that two proteins have close functional associations if they share a significantly larger number of common neighbours than random. They use p-value to rank all pairs of proteins in the PPI network and select only interactions with a p-value smaller than a threshold. Clusters are then generated in the weighted network. The algorithm is stable. Even adding 50% randomly generated interactions to the PPI dataset, it can still recover 89% of the original associations. Li and Liang [77] further used this heuristic by comparing a PPI network with truncated power-law preserving random networks and find that the likelihood of two proteins sharing a common or related biological function can be enhanced if they share significantly more neighbours than random. They adopt this idea to investigate the functional relationship among proteins of a human PPI network.

Because a weighted PPI network is often more accurate than the initial pair-wise one, the weighted PPI network itself can be used to improve the weight in an iterative manner. Liu et al. [78] propose an iterative scoring method to reassign weights for edges and develop the CMC (Clustering based on Maximal Cliques) method to identify protein complexes from PPI networks. The initial weight of an edge is calculated from the AdjustCD-distance by using the neighbourhood information and two penalty parameters. Although this iterative scoring method can effectively reduce the impact of random noise, more iterative steps do not necessarily produce better results. They suggest that two iterative steps are usually a safe choice. The CMC algorithm then generates all the maximal cliques from the weighted PPI network. Highly overlapped cliques are removed or merged to achieve the final predictions of complexes.

For any given weighted PPI network, Nepusz et al. [75] recently design the ClusterONE (Cluster with Overlapping Neighbourhood Expansion) to detect overlapped protein complexes from the network. They argue that a meaningful candidate cluster representing a protein complex should have two structural properties. First, it should contain many reliable interactions within the cluster. Second, it should be well separated from the rest of the network. Based on this heuristic, they define a cohesiveness score for a group of vertices, which considers the total weight of edges within those vertices and the total weight of edges between these vertices and the rest of the network. A penalty term is also included to model uncertainties of the undiscovered interaction data. Starting from each seed vertex, the algorithm iteratively generates high cohesiveness clusters by using a greedy procedure. After that, clusters are merged if the overlap score is above a specified threshold, and candidates that contain less than three proteins or whose density is below a given threshold are removed. The cohesiveness measures how likely a cluster is to form a protein complex, which provides an easy and efficient way to assess predictions for almost all algorithms. Wang et al. [79] propose the EPOF (Essential Protein and lOcal Fitness) by using essential proteins and the local vertex fitness. The fitness of a sub-graph is defined similar to the cohesiveness score used in [75], without the penalty term. Then the vertex fitness of v is defined for a sub-graph as the difference of the sub-graph fitness with and without the vertex v. A seed growth style algorithm is proposed in [79], where cliques that consist of only essential proteins and those do not contain any essential proteins are used as seeds, respectively.

Various hierarchical clustering algorithms also make contributions for identifying communities from different networks. Most of them can be used in PPI networks in terms of detecting protein complexes. Biological processes usually exhibit hierarchical structures in which proteins physically bind together as stable complexes [6]. The general procedure of hierarchical clustering algorithms is illustrated in Figure 3, where a weighted network is commonly transformed into a dendrogram [38]. The leaves of the dendrogram represent the vertices of the network while the branches from joint nodes.
indicate groups of clusters in the network. Therefore, identifying hierarchical structure of clusters equals to designing a way to generate such a dendrogram and assigning joint nodes for branches.

Typically, two methods of generating the dendrogram of a network are used: agglomerative and divisive. The agglomerative method starts from the state of all vertices in distinct clusters. The similarity of each pair of vertices in the network is calculated, which represents how closely the vertices are connected. Vertices and/or branches are iteratively organized into the hierarchical structure by merging the highest similar clusters step-after-step. In this method the dendrogram is built from leaves to the root, where all vertices of the network in one cluster. In contrast, the divisive method builds the dendrogram in a reverse order. It first starts from all vertices in one cluster and then subsequently splits the big cluster iteratively into smaller ones identified as clusters. In this manner, the dendrogram will down to the level of single vertices. In practice, additional information is needed to decide which branches of the dendrogram have real significance [38]. Figure 4 illustrates the basic idea behind the agglomerative and divisive clustering method.

Two issues are usually considered in a hierarchical clustering algorithm: assigning weights to edges for iteratively merging/splitting clusters and designing quantitative measures to evaluate output clusters.

For the first issue, Girvan and Newman [35] introduce a divisive algorithm, the G-N algorithm, based on the value of ‘edge-betweenness’. The betweenness of an edge is defined as the number of all shortest paths running through it [35]. The rationale behind this idea is that a highly organized network is filled with densely inter-community edges and sparsely intra-community edges. Therefore, all shortest paths between vertices of different clusters have to go through a few intra-community edges, thereby obtaining higher betweenness values. The algorithm iteratively removes the edges of the highest betweenness until a given network breaks into desired number of clusters. The G-N algorithm represents a major step in terms of identifying communities in networks, and is widely adopted to investigate functional associated communities in PPI networks in the past years. Dunn et al. [36] apply the G-N algorithm on a small set of human protein interactions to investigate biological functions involved in them. Newman [37] proposes a new agglomerative algorithm to improve the computational efficiency of the G-N algorithm. Radicchi et al. [38] develop a fast algorithm to address the similar issue. They alternatively introduce an edge-clustering coefficient by considering the number of triangles that build on edges. Edges connecting vertices in different communities are included in a few or no triangles and tend to have small values of edge-clustering coefficients. Rives and Galitski [39] use the all-pairs-shortest path matrix to define an association for each pair of vertices in a network. The association is calculated by $f/d^2$, where $d$ is the length of the
shortest path. Then they develop an agglomerative algorithm based on the average linkage to reveal the modular organizations in yeast signaling networks. Wang et al. [40] propose a fast agglomerative algorithm, called HC-PIN, by using the number of common neighbours to calculate the clustering value of individual edges in a weighted PPI network. Cho and Zhang [41] introduce another way to use the hierarchical idea to identify functional hubs and modules in a network. They propose an algorithm by exploring two intrinsic topological features of PPI networks: the high modularity and the hub-oriented structures. A weighted PPI network is taken as input and a path strength model is designed to measure the functional similarity between protein pairs. Then the network is converted into a hub-oriented hierarchical structure and communities are generated by using the score of hub confidence.

For the other issue of designing measures to evaluate clusters of hierarchical algorithms, various quantitative measures are proposed. Newman and Girvan [37, 80] introduce a measure called the modularity $Q$ by comparing the observed fraction of edges inside a cluster with the expected fraction of edges in the cluster. It is defined on the global sense. However, in many networks, sub-graphs are only locally connected. Based on this idea, Muff et al. [81] give a local version of the modularity measure, $LQ$, by considering only the immediate neighbours of a cluster, rather than the entire network. Kim and Tan [34] extend this idea by introducing a coarseness parameter. Li et al. [82] argue that the modularity $Q$ has been exposed to resolution limits. The size of a detected community by $Q$ depends on the size of the whole network, which may fail to identify modules smaller than a scale. Alternatively, they propose a modularity density, which they call $D$ value based on the concept of average modularity degree. Zhang et al. [83] further extend the $D$ value into a more general case, where a tuning parameter is introduced. They also adopt the simulated annealing algorithm to maximize the modularity density. Radicchi et al. [38] define the concepts of strong community and weak community, by considering the

Figure 4: Schematic of the (A) agglomerative and (B) divisive clustering methods. In agglomerative clustering, the distance between two clusters is calculated by using the single-linkage method. In the divisive clustering, edges of high distance within a cluster are removed until the cluster breaks into two separated clusters.
Identifying protein complexes and functional modules

connections within a cluster and those toward vertices in the rest of the network. It gives a general criterion for deciding which detected sub-graphs are meaningful. Chen and Yuan [44] extend the idea to a directed graph, and propose a quantitative measure in both strong and practical sense.

IDENTIFYING PROTEIN COMPLEXES AND/OR FUNCTIONAL MODULES BY MULTIPLE DATA INTEGRATIONS

It is believed that no single experimental approach can reach the sensitivity of 100% (i.e. no false negative) and the specificity of 100% (i.e. no false positive) [42]. The data emerging from individual ‘omic’ approaches should be viewed with caution [84]. Moreover, large-scale PPI data usually do not readily allow one to discriminate their various features [85] such as the interaction strength (affinity), the type of interactions (protein–protein interaction or protein–peptide interaction), and spatiotemporal existence (where and when the proteins are present and interact). However, it does not mean that we can do nothing to deal with these issues. The approach multiple data integrations can achieve this goal to some extent.

Various kinds of data contain the information of protein interactions. Besides the high-throughput technologies such as the Y2H assay and AP/MS, many other kinds of information such as the reliability of experiments, the gene expression profile (gene microarray, co-expression), the GO terms [86] and the subcellular localization annotations [87] can be used to assess the reliability of PPIs and their biological features. Of course, the additional cautious should still be emphasized here. As Hakes et al. [56] remind that keeping only those highly reliable data may introduce new biases about the PPI data. Reasonable ways of data integration needs to pay attention to the interpretation of PPI networks.

Firstly, the reliability of experimental technologies is commonly used to evaluate PPIs. It is clear that on the one hand, interactions observed at multiple times should be more likely to be true than those that have only been observed once, and on the other hand, the reliability of different experiment methods are not always the same. Therefore, one way to achieve the high reliability of PPIs is to assign different weights to interactions according to different times they are reported and different types of experiments they are derived from. Chatr-aryamontri et al. [42] have concluded that sensible and biologically relevant results can be obtained by integrating multiple interaction evidences. For instance, Tan et al. [43] first build interaction-specific networks independently from six groups of data. The integrated network is obtained by the weighted combination of individual networks. The MINT [66] is one of the databases that annotate various information such as detection methods, expression levels, protein tags, in vivo and in vitro conditions, the experimental role, post-translational modifications, and so on [42], which can be used to evaluate the reliability of PPIs from the experimental viewpoints.

Secondly, genomic associations are believed to reflect functional associations between their proteins [45]. It is acknowledged that the strength of genomic associations correlates with the strength of protein interactions. Various genomic contexts such as gene fusions, gene co-occurrences, gene expression profiles, phenotype data and transcription factor binding data have been used to predict functional associations [88]. Tanay et al. [46] propose a biclustering algorithm that integrates genomic data to partition the molecular network of yeast. They use a weighting scheme on a bipartite graph to identify groups of genes with statistically significantly correlated behaviour. Snel et al. [45] introduce a method that integrates genomic associations to identify functional modules. Among those kinds of genomic information, gene expression profiles are most commonly used for data integration. Genes with similar expression profiles tend to encode proteins that interact with each other [47]. Integrating PPIs and gene-expression data can generate a meaningful biological content in terms of identifying functional associations [48]. Chen and Yuan [44] use the abundant information of microarray expression profiles to assign weights to edges of PPI networks. The weight of an edge represents the dissimilarity between two associated expression profiles. They extend the idea of edge-betweenness to a ‘non-redundancy’ way. The shortest paths are not enumerated among all-against-all vertices, but rather the non-redundancy ones. An extended G-N algorithm is also proposed to find functional modules in weighted PPI networks in [44].

Thirdly, the GO annotations [86] contribute to another resource that can be used to assign weights to PPI networks. The weights of edges in PPI networks can be assigned by the semantic similarity of
the relative GO terms. It is an effective way to identify protein complexes than the unweighted ones. Lubovac et al. [49] use two measures, called weighted clustering coefficient and weighted average nearest-neighbours degree, to assign weights to protein interactions. They are calculated from Lin’s similarity [89] of GO terms. The SWEMODE (Semantic WEights for MODule Elucidation) algorithm is developed to identify communities containing functionally similar proteins. It first ranks vertices in the network according to their weighted clustering coefficient. Those with the high rank are iteratively selected as seeds to generate densely connected clusters with high functional similarity according to the chosen parameters. Xu et al. [50] propose the OIIP approach to identify protein complexes from a weighted PPI network. The weight of an edge is assigned according to the annotation size of GO terms while the weight of a vertex is assigned by summing weights of incident edges. A seed-growth style is applied to this network similarly to the way that the IPCA [33] algorithm used.

Finally, many algorithms try to integrate more kinds of information for PPI networks. Shi and Zhang [90] first use GO to build a weighted PPI network. Then a semi-supervised learning method is developed to learn features of protein complexes. There are 21 features of protein interactions used in their multi-layer neural network model, all of which are used to identify protein complexes in the weighted networks. Georgii et al. [91] develop the DME (Dense Module Enumeration) to detect all clusters that satisfy a user-defined minimum density threshold in a given weighted network. The weight can be determined by any additional information, such as gene expression, phenotype data, evolutionary conservation and subcellular localization. Luo et al. [92] propose a framework for discovering conditional co-regulated protein complexes by integrating transcription regulation data, gene expression data and PPI data. Jansen et al. propose a Bayesian approach to combine multiple types of data to reconstruct PPI networks in [93].

DISTINGUISHING BETWEEN PROTEIN COMPLEXES AND FUNCTIONAL MODULES VIA DYNAMIC PPI NETWORKS

Although PPIs imply physical contact between proteins, it does not mean that all possible interactions occur in any cell at any time. PPIs are not static but dynamic [71, 94, 95]. They vary with time and space that mediate protein complexes to assemble and disassemble as cellular processes [11, 85]. It is thus crucial to understand a PPI network in a sense of dynamic, such as how the cellular system responds to cues of environment and how it changes during the development or differentiation [95]. It is essential to shift the analysis of PPI networks from static to dynamic for further understanding of molecular systems [51].

The large-scale PPI datasets are unable to capture the dynamic properties of protein interactions [96]. The challenge now becomes how to grasp dynamic behaviour of PPI networks and how to figure out which interactions occur simultaneously. The commonly used way is by projecting additional information onto PPI networks [85, 97]. The temporal dimensionality of PPI networks can be enhanced by linking protein complexes to time series of gene expression data [11] while the spatial information can be partly handled from the subcellular localization annotations [98]. For instance, de Lichtenberg et al. [98] use both those data to investigate the dynamics of protein complexes during the yeast cell cycle. They find that almost all complexes contain both dynamic and static subunits. Most of them cannot be identified through the analysis of any single type of experimental data, but only through integrative analysis of several types of data. Moreover, condition-specific co-expression information also gives a way to achieve the dynamic features of the networks. Lin et al. [99] integrate PPIs with biological annotations and gene expression profiles to reveal dynamic functional modules under conditions of dilated cardiomyopathy. They show that hub proteins tend to differentially express in different biological states. However, Lu et al. [100] claim that hubs and superhubs tend to have similar gene expression profiles under conditions of experimental asthma, by comparing with peripheral vertices based on the GO classification. Moreover, Han et al. [101] investigate how hubs might contribute to robustness and other cellular properties in the yeast PPI network. They find two types of hubs: party hubs and date hubs. Party hubs interact with most of their partners simultaneously to function inside modules, whereas date hubs bind their partners at different times or locations to organize the proteome and connect biological processes.

A generalized framework to identify communities in a dynamic network is introduced by Mucha et al.
[52]. It can be used in time-dependent, multiscale and multiplex networks that contain arbitrary slices. Each slice represents a network at a specific time point. In terms of identifying protein complexes from PPI networks, the time-course microarray data can be used to reconstruct such dynamic behaviour. The composition of protein complexes and/or functional modules may change during a cell cycle. Even in the same time, a ‘protein’ may also be involved in several different processes (by different protein copies). A party hub can be identified from the network in each slice, while the date hub can be detected by considering multiple slices. It is also possible to distinguish between protein complexes and functional modules from such dynamic PPI networks by checking whether detected communities are in individual slices or not. Figure 5 gives a simple schematic for reconstructing dynamic PPI networks from a small static PPI network.

It is proposed that different protein modules can be found in the vertices of dynamic PPI networks [11]. Permanent interactions are strong and stable, which give rise to protein complexes while the transient interactions vary with cellular processes and form functional modules. Yu et al. [5] find the bottlenecks in protein interaction networks are key connectors that correspond to the dynamic properties. Komurov and White [102] conclude that static and dynamic modules in the eukaryotic protein interaction network have distinct properties. Static modules provide robustness to the cell against genetic perturbation or protein expression noises while dynamic modules are mainly responsible for condition-dependent regulations of cell behaviours. Taylor et al. [103] examine the dynamic structure of the human protein interaction network. They argue that similar to the date hub and party hub, inter-modular hubs co-express with their partners in a tissue-restricted manner while intra-modular hubs co-express with their partners in most tissues.

In the framework of dynamic networks, Mucha et al. [52] develop a way to detect communities from such multiple slice networks. Similarly to the situation in static networks, the way they propose to quantify communities is by comparing the number of intra-community edges to what one would expect at random. Three types of connections are considered: intra-slice connections, inter-slice connections between only neighbouring slices and inter-slice connections among all-to-all slices. A multiple adjacency matrix is also defined to handle the problem of community identification. Jin et al. [53] define a dynamic network module to be a set of proteins satisfying two conditions. First, it is connected in the static PPI networks. Second, the expression profiles of its vertices form certain structures in the temporal domain. Then they detect dynamic modules in temporal networks by a mining algorithm. Although most dynamic network modules are highly condition-specific, they further demonstrate that identifying frequent dynamic modules can significantly increase the signal to noise separation. Tang et al. [51] propose a time course PPI model to identify functional modules. Time series gene expression data are used to construct the network. Although the temporal parameters are not sufficiently accurate, they find that the functional modules from the time course network have much more significant biological meanings, comparing with the static PPI network and a pseudorandom network.

Figure 5: Static PPI network and dynamic PPI network. (A) A part hub, a data hub, a protein complex and a functional module in static PPI network. (B) The protein complex and functional module can be distinguished by checking their existence in individual slices. The party hub and date hub can also be identified by checking protein interactions in different time points.
EVALUATION METHODS

Evaluating a set of prediction algorithms is always a challenging problem, especially when there is no complete gold standard dataset available as a reference. The incompleteness of the datasets would introduce biases for any evaluation methods, which may mislead the comparison results. In addition, there is no widely accepted evaluation criterion. One algorithm may outperform the others in terms of one criterion, yet it may perform worse in terms of other criteria. Nevertheless, we believe that each algorithm has its own advantages and is helpful to spark novel ideas for the identification of protein complexes. Moreover, this article is to give a survey on how computational methods revolve with the available data. Hence, we summarize widely used criteria across this research area, without quantitatively comparing those algorithms.

Typically, known gold standard protein complexes, GO annotations or localization annotations are often involved in those criteria.

Giving a set of gold standard protein complexes as references, two levels of comparison are conducted to perform such evaluation: (i) the comparison between a predicted protein complex and a reference protein complex and (ii) the comparison between a group of predicted protein complexes and a group of reference protein complexes.

Four measures are commonly used to compare the difference between a predicted complex $X$ and a reference complex $P$, which are precision, recall, f-score and overlapping score. Suppose a predicted protein complex $X$ is compared with a reference protein complexes $P$, the precision and recall are defined as follows:

\[
\text{precision} = \frac{|X \cap P|}{|X|} \quad \text{and} \quad \text{recall} = \frac{|X \cap P|}{|P|}.
\]

The $f$-score is the harmonic mean of precision and recall, which is

\[
f-score = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} = \frac{2 \cdot |X \cap P|}{|X| + |P|}.
\]

while the overlapping score is defined as:

\[
\text{overlapping score} = \frac{|X \cap P|^2}{|X| \cdot |P|},
\]

which is the multiplying between precision and recall. The comparison of those metrics is shown in Figure 6.

After calculating the $f$-score and/or overlapping score for each predicted and reference protein complex pairs, the set of true-positive predictions can be obtained by selecting predictions with $f$-score or overlapping score larger than a threshold. Most researchers also use the term precision and recall to represent the true-positive rate and the positive predicted value, respectively, when comparing a group of predicted complexes with a group of references. To make distinction from previous ones, we use $Pr$ and $Rc$ to represent precision and recall, respectively, in this situation. They are defined as:

\[
Pr = \frac{TP}{TP + FN} \quad \text{and} \quad Rc = \frac{TP}{TP + FP}.
\]

The F-measure combines the $Pr$ and $Rc$, which is defined as:

\[
f-measure = \frac{2 \cdot Pr \cdot Rc}{Pr + Rc}.
\]

Generally, the value of the average $f$-score or the $f$-measure can be used to measure the performance of an algorithm. However, because the number of predictions varies widely for different algorithms and the set of reference protein complexes are commonly incomplete, it is unfair to use the average $f$-score or the $f$-measure to compare different algorithms. Moreover, a reference protein complex often partially matches with more than one predicted complex and vice verse. To handle this problem, Nepusz et al. [75] introduce a maximum matching ratio (MMR) to evaluate different predictions. A weighted bipartite graph is built where two sets of vertices represent the predicted and reference complexes.
Identifying protein complexes and functional modules

protein complexes, respectively, and weights of edges represent the overlapping score between predicted and reference proteins complexes. The MMR is calculated by the total weight of the maximum matching edges, divided by the number of reference complexes.

The other commonly used measure is called accuracy (Acc), which is the geometric mean of the clustering-wise sensitivity (Sn) and the clustering-wise positive predictive value (PPV). Given m predicted and n reference protein complexes, a confusion matrix \( T = [t_{ij}] \) is constructed, where \( t_{ij} \) denote the number of common proteins in the \( i^{th} \) reference and the \( j^{th} \) predicted complex. The Sn and PPV are defined as:

\[
Sn = \frac{\sum_{i=1}^{n} \max_{j} t_{ij}}{\sum_{i=1}^{n} n_i}, \quad \text{and} \quad PPV = \frac{\sum_{i=1}^{n} \max_{j} t_{ij}}{\sum_{i=1}^{m} \sum_{j=1}^{m} t_{ij}},
\]

where \( n_i \) is the number of proteins in the \( i^{th} \) reference protein complex. The Acc is then defined as:

\[
Acc = \sqrt{Sn \cdot PPV}.
\]

Because gold standard protein complex datasets are commonly incomplete, a predicted protein complex that does not match with any known complexes may belong to valid but still uncharacterized complexes. Hence, it is also important to analyse those unmatched predictions by using GO annotations and/or localization annotations.

For GO annotations [86], they are usually accepted as ground-truth and used for comparison and validation purposes. A prediction can be statically evaluated using the \( p\text{-value} \) defined by the following hypergeometric distribution:

\[
p\text{-value} = 1 - \sum_{i=0}^{k-1} \binom{F}{i} \binom{N-F}{n-i} / \binom{N}{n},
\]

where \( F \) is the number of proteins in a GO term, \( n \) is the number of proteins in the predicted complex, \( k \) is the number of proteins they have in common and \( N \) is the total number of proteins in a PPI network. The smaller the \( p\text{-value} \) is, the more statistically significant the protein complex is enriched by GO annotations.

The other kind of data used is localization annotations. This is motivated by the fact that a protein complex can be formed only when its constituents are to be found in the same cellular compartment [87]. The co-localization score of a single complex is defined as the maximum fraction of proteins in the complex that are found at the same localization. The co-localization score of a set of complexes is the mean co-localization score of all complexes in the set, weighted by the size of the complexes.

CONCLUSIONS

In this review, we focus on computational algorithms for identifying protein complexes and functional modules from PPI networks in terms of what kinds of data are used and what kinds of detection ideas are based on. Four categories of algorithms for interpreting PPI networks are surveyed from static ones to dynamic ones. The first category focuses on algorithms that based on only topological structures of a single static PPI network. They treat vertices and edges equally in a PPI network and sub-graphs such as cliques, dense sub-graphs, core-attachment structures and star-like structures are mined as predictions of protein complexes. The next category consists of algorithms that based on characters of weighted PPI networks. They are also based on a single static PPI network, but they use various topological indices of a network to assign weights to vertices and/or edges. Many hierarchical clustering algorithms also contribute to this category. The third category addresses algorithms that involving multiple data integrations. They use other experimental dependent and/or independent datasets to assign weights to PPI networks, such as the experimental conditions, gene expression profiles and GO annotations. More biological meaningful results can be achieved by using such data integration. The fourth category is made up of algorithms that involving dynamic PPI networks. They are reviewed from the general framework of dynamic systems to the time-course PPI networks. It is hard to say whether an algorithm is better than the other, as there is no generally accepted criterion to perform such comparison. In this review, we have summarized some evaluation measures to compare algorithms, which are widely used across this research area. It is believed that the PPI networks are modelled increasingly precise when integrating more types of data, and the study of protein complexes would benefit by shifting from static to dynamic PPI networks.
Key points

- Issues and pitfalls of protein interactions and PPI networks for identifying protein complexes and/or functional modules are summarized in Section 2.
- Protein complex identification methods based on topological structures of unweighted PPI networks are reviewed in Section 3.
- Protein complex identification methods based on characters of weighted PPI networks are reviewed in Section 4.
- Multiple data integration methods for identifying protein complexes and/or functional modules are reviewed in Section 5.
- Algorithms that involve dynamic PPI networks are reviewed in Section 6.

FUNDING

Natural Sciences and Engineering Research Council of Canada (NSERC); National Science Foundation of China [61272274, 60970063]; Program for New Century Excellent Talents in Universities [NCET-10–0644]; Ph. D. Programs Foundation of Ministry of Education of China [20090141110026].

References

Identifying protein complexes and functional modules


