The Author(s) *BMC Bioinformatics* 2016, **17**(Suppl 12):372
DOI 10.1186/s12859-016-1191-6

BMC Bioinformatics

RESEARCH Open Access

Protein complex prediction for large protein protein interaction networks with the Core&Peel method

Marco Pellegrini*, Miriam Baglioni and Filippo Geraci

From Twelfth Annual Meeting of the Italian Society of Bioinformatics (BITS)
Milan, Italy. 3-5 June 2015

Abstract

Background: Biological networks play an increasingly important role in the exploration of functional modularity and cellular organization at a systemic level. Quite often the first tools used to analyze these networks are *clustering algorithms*. We concentrate here on the specific task of predicting protein complexes (PC) in large protein-protein interaction networks (PPIN). Currently, many state-of-the-art algorithms work well for networks of small or moderate size. However, their performance on much larger networks, which are becoming increasingly common in modern proteome-wise studies, needs to be re-assessed.

Results and discussion: We present a new fast algorithm for clustering large sparse networks: *Core&Peel*, which runs essentially in time and storage $O(a(G)m + n)$ for a network G of n nodes and m arcs, where $a(G)$ is the *arboricity* of G (which is roughly proportional to the maximum average degree of any induced subgraph in G). We evaluated *Core&Peel* on five PPI networks of large size and one of medium size from both yeast and homo sapiens, comparing its performance against those of ten state-of-the-art methods. We demonstrate that *Core&Peel* consistently outperforms the ten competitors in its ability to identify known protein complexes and in the functional coherence of its predictions. Our method is remarkably robust, being quite insensible to the injection of random interactions. *Core&Peel* is also empirically efficient attaining the second best running time over large networks among the tested algorithms.

Conclusions: Our algorithm *Core&Peel* pushes forward the state-of the-art in PPIN clustering providing an algorithmic solution with polynomial running time that attains experimentally demonstrable good output quality and speed on challenging large real networks.

Keywords: Large PPI networks, Protein complex prediction, Efficient algorithm

Abbreviations: AS, Aggregated score; BP, Biological process; C, Core number; CC, Core count; CD, Core decomposition; FDR, False discovery rate; FN, False negative; FP, False positive; GO, Gene ontology; PC, Protein complex; PDC, Partial dense cover; PPI, Protein protein interaction; PPIN, Protein protein interaction network; TAP-MS, Tandem affinity purification coupled with mass spectrometry; UBC, Ubiquitin; Y2H, Yeast two-hybrid system

*Correspondence: m.pellegrini@iit.cnr.it
Laboratory for Integrative Systems Medicine - Istituto di Informatica e Telematica and Istituto di Fisiologia Clinica del CNR, via Moruzzi 1, 56124 Pisa, Italy

Background

Due to recent advances in high-throughput proteomic techniques, such as yeast two-hybrid system (Y2H) and Tandem Affinity Purification coupled with Mass Spectrometry (TAP-MS), it is now possible to compile large maps of protein interactions, which are usually denoted as *protein-protein interaction networks* (PPIN). However, extracting useful knowledge from such networks is not straightforward. Therefore sophisticated PPI network analysis algorithms have been devised in the last decade for several goals such as: the prediction of protein-complexes ([1]), the prediction of higher level functional modules ([2–4]), the prediction of unknown interactions ([5, 6]), the prediction of single protein functions ([7]), the elucidation of the molecular basis of diseases ([8]), and the discovery of drug-disease associations ([9]), to name just a few. In this paper we concentrate on the issue of predicting protein-complexes (PC) in PPI networks. An incomplete list of complex prediction algorithms in chronological order is: MCODE [10], RNSC [11], Cfinder [12], MCL [13], COACH [14], CMC [15], HACO [3], CORE [16], CFA [17], SPICi [18], MCL-CAw [19], ClusterONE [20], Prorank [21], the Weak ties method [22], Overlapping Cluster Generator (OCG) [23], PLW [24], PPSampler2 [25], and Prorank+ [26]. Further references to existing methods can be found in recent surveys by [1, 27–31].

The graph representing a PPIN can also be augmented so to include additional biological knowledge, annotations and constraints. The conservation of protein complexes across species as an additional constraint is studied in [32]. Jung et al. [33] encode in PPIN the information on mutually exclusive interactions. Proteins in PPIN can also be marked with cellular localization annotations ([34]), and several types of quality scores. Though all these aspects are important, they are possible refinements applicable to the majority of the algorithms listed above, involving the modeling of additional knowledge in the PPIN framework (see [35]). In this paper we concentrate on the basic case of a PPIN modeled as an undirected and unweighted graph. The size of PPIN found in applications tend to grow over time because one can obtain with modern techniques from a single high-throughput experiment thousands of novel PPI, and also because one can collate groups of PPI from different experiments into a single larger network (ensemble PPIN) [36]. For example very large PPIN arise in multi-species PPI studies, ([37, 38]), in immunology studies ([39, 40]) and cancer data analysis ([41]). Large PPIN can be challenging for clustering algorithms as many of them have been designed and tested in the original publication with PPIN of small and medium size (with the possible exception of SPICi ([18]), that was designed intentionally for large PPIN). Greedy methods that optimize straightforward local conditions may be fast but speed may penalize quality. Thus, although more than a decade has passed since the first applications of clustering to PPIN, the issue of growing PPIN size poses new challenges and requires a fresh look at the problem.

We develop a new algorithm (*Core&Peel*) designed for clustering large PPIN and we apply it to the problem of predicting protein complexes in PPIN. The complexes we seek have just very basic properties, they should appear within the PPIN as ego-networks of high density and thus we can model them as maximal quasi-cliques. These features are not particularly new, but we show in Section 'Experiments' that they are sufficient to characterize a large fraction of PCs in a sample of five large PPIN for two species (yeast and human). Computational efficiency is attained by a systematic exploitation of the concept of *core decomposition* of a graph, which for each vertex (protein) in a graph provides a tight upper bound to the size of the largest quasi-clique that includes that vertex. We use this upper bound to trim locally the subgraphs of interest in order to isolate the sought quasi-clique, and proceed then to the final *peeling* out of loosely connected vertices. Our approach has some superficial similarity with that of CMC ([15]) which applies the enumeration algorithm of [42] to produce, as an intermediate step, a listing of all maximal cliques in a graph. We avoid this intermediate step that may cause an exponential running time in large PPIN and cannot be adapted easily to listing all maximal quasi-cliques, when density below 100 % is sought. Our approach is both more direct (no intermediate listing of potentially exponential size is produced) and more flexible (as we can tune freely and naturally the density parameter).

CFinder ([12]) lists all k-cliques, for a user defined value of k, and then merges together k-cliques sharing a (k-1)-clique. CFinder might produce too many low density clusters if the user choosees k too small, or miss interesting complexes if k is too large. *Core&Peel* avoids both pitfalls since we have a more adaptive control over cluster overlaps. Our algorithm is empirically very fast: all instances in this paper run in less than 2 minutes on common hardware. The asymptotic analysis (see Additional file 1: Section 8) indicates a running time very close to linear for sparse graphs. More precisely, with some additional mild sparsity assumptions, the algorithm runs in time $O(a(G)m + n)$ for a graph G of n nodes and m arcs, where $a(G)$ is the *arboricity* of G (which is roughly proportional to the maximum average degree of any induced subgraph in G). The output quality is assessed by comparative measures of the ability to predict known complexes and of the ability to produce biologically homogeneous clusters, against 10 state-of-the art methods. In both quality assessments *Core&Peel* leads or ties in most tests vs all other methods, often by a

large margin (See Section 'Comparative evaluation'). The robustness of our method is remarkably high, since practically no output variation is measured even when adding up to 25 % random edges in the input graph. Finally, we show several high quality predicted clusters that involve a known complex with additional proteins, which correspond to biologically relevant mechanisms described in literature.

Paper organization

In Section Methods we start by reviewing the issue of false positive/negative PPI in large PPIN with hindsight from the work in [5] indicating quasi-cliques as good models for protein complexes in our settings (Section 'On false positive and false negative PPI in dense and large PPIN). Next, in Section Preliminaries we recall the basic graph-theoretic definitions of subgraph density, quasi-cliques, and core-decompositions, that are central to our algorithmic design. In Section 'Partial dense cover of a graph' we introduce the notion of a *partial dense cover* as a formalization of our problem, showing its similarities with well known NP-hard problems of *minimum clique cover* and *maximum clique* [43]. In Section 'Algorithm *Core&Peel* in highlight' we give a high level description of our proposed polynomial time heuristic. For ease of description it is split in four phases, though in optimized code some of the phases may be interleaved. The rationale behind certain design choices is explained in further detail in Section 'Algorithm description: details'. The asymptotic analysis of the proposed algorithm can be found in (Additional file 1: Section 8). The experimental set up is described in Section 'Results and discussion', including the sources of raw data, the initial data cleaning (Section 'Used data and preprocessing') and the quality score functions (Sections 'Evaluation measures for protein complex prediction' and Evaluation measure for Gene Ontology coherence). Further data statistics and details of the comparative evaluations are in Section 'Experiments' and 'Comparative evaluation'. In particular we report on the ability to capture known complexes in Section 'Performance of protein complex prediction', to produce functionally coherent clusters (Section 'Coherence with Gene Ontology annotation'), on robustness in presence of random noise (Section 'Robustness against noise in the PPIN graph'), and on computation timings (Section 'Running times').

In Section 'Some predictions with support in the literature' we list ten interesting predictions in which a known complex interacts with an additional protein. These findings have an independent support in the literature. Finally in Section 'Conclusions' we comment on the potential applications and extensions of the proposed method, as well as on its limitations.

Methods

On false positive and false negative PPI in dense and large PPIN

The estimation of the number of erroneous PPI calls (false positive/false negative) in PPI networks is highly dependent on the technology and the experimental protocols used. Yu et al. [5] report an experiment on 56 proteins of Saccharomyces cerevisiae (yeast) for which PPI were detected using both error-prone high throughput technologies and more precise low throughput technologies. In 563 cases (pairs of proteins) for which the two methods differ, the vast majority (92.5 %) were false negatives (FN), and just 7.5 % false positive (FP). A similar ratio among FP/FN rates is reported in [36] for PPI obtained through Y2H and high confidence AP-MS techniques. While each technology has its own systematic biases, it is observed in [36] that such biases tend to compensate each other when data from several sources is used to compile ensemble PPIN. The implication is that, over time, as the evidence on reliable PPI accumulates, the number of undetected real PPI (FN) will steadily decrease, while the number of spurious PPI (FP) should increase quite slowly. In graph terms the subgraphs representing complexes in the PPI will become denser (i.e. closer to a clique), while the noisy interactions will still remain within a controllable level (assuming that only high quality interaction data is encoded in the PPI networks). Expanding on these finding Yu et al. [5] demonstrate that *quasi-cliques* (cliques with a few missing edges) are good predictors of the presence of a protein complex, provided the PPIN is large. Our own measurents on one medium size graphs ($\approx 20K$ PPI) and four large graphs ($\approx 130K/220K$ PPI) in Section 'Experiments' confirm this tendency of protein complex density increase in larger PPIN. Besides the increase in density, a second notable phenomenon, is that protein complexes often resemble ego-networks, that is, the protein complex is mostly contained in the 1-neighborhood of some protein (see Section 'Experiments').

Preliminaries

An early incarnation of the *Core&Peel* algorithm targeting communities in social graphs is described in [44]. In order to make this paper self-contained we are describing in this section a version of *Core&Peel* that includes all the modifications needed to target potentially overlapping protein complexes in PPI network. Let $G = (V, E \subseteq V \times V)$ be a simple (undirected) graph (no self-loops, no multiple edges). A subset $Q \subset V$ induces a subgraph $H_Q = (Q, E_Q)$, where $E_Q = \{(a,b) \in E | a \in Q \wedge b \in Q\}$. For a graph G its *average degree* is:

$$av(G) = \frac{2|E|}{|V|}.$$

The *density* of a graph $D(G)$ is the following ratio:

$$D(G) = \frac{|E|}{\binom{|V|}{2}} = \frac{2|E|}{|V|(|V|-1)},$$

which gives the ratio of the number of edges in G to the maximum possible number of edges in a complete graph with the same number of nodes. We restrict ourselves to *local density definitions*, such as the two listed above, that are those for which the density of a subgraph induced by a subset $Q \subseteq V$ is a function depending only on Q and on the induced edges set E_Q. A nice survey of concepts and algorithms related to local density of subgraphs is in [45]. Cliques are subgraphs of density 1, and finding a maximum induced clique in a graph G is an NP-complete problem [46]. Several relaxations of the notion of clique have been proposed (see [47] for a survey), most of which also lead to NP-complete decision problems. Given a parameter $\gamma \in [0..1]$, a γ-quasi clique is a graph $G = (V, E)$ such that:

$$\forall v \in V \ |N_G(v)| \geq \gamma(|V|-1),$$

where $N_G(v) = \{u \in V | (v, u) \in E\}$ is the set of immediate neighbors of v in G. Note that a γ-quasi clique has density $D(G) \geq \gamma$. In general, however, for a dense graph with density $D(G)$ we cannot infer a bound on the value of γ for which there exists a quasi-clique in G (except for the value $D(G) = 1$ that implies $\gamma = 1$, and those cases covered by Turán's theorem ([48])). If we impose that the number of vertices in a subgraph is exactly k, then the average degree and the density depend only on the number of edges, and thus they attain their maximum values for the same subgraphs. Without this constraint, finding the subgraph of maximum average degree or the subgraph of maximum density are quite different problems: the former admits a polynomial time solution, the latter is NP-complete. In this paper we aim at detecting dense-subgraphs with a lower bound on the size of each sub-graph and on its density, thus still an NP-complete problem. A *k-core* of a graph G is a maximal connected subgraph of G in which all vertices have degree at least k. A vertex u has *core number* k if it belongs to a k-core but not to any (k+1)-core. A *core decomposition* of a graph is the partition of the vertices of a graph induced by their core numbers ([49]).

Partial dense cover of a graph

In this section we formalize our problem as that of computing a *partial dense cover* of a graph. We aim at collecting efficiently only high quality candidate dense sets that cover the dense regions of the input graph. A Partial Dense Cover $PDC(G, r, \delta, q)$ is defined as the range of the function $f : V \to 2^V$ that associates to any vertex $v \in V$ a subset of V with these properties:

(a) if $f(v) \neq \emptyset$ then $v \in f(v)$, (the set $f(v)$ contains the seed v or it is empty).
(b) $f(v) \subseteq N^r(v) \cup \{v\}$, (the set $f(v)$ is a subset of the r-neighborhood of v, i.e. all its vertices are at distance at most r from v. (In this study, we set $r = 1$ throughout)
(c) $f(v)$ is the largest set having size at least q, density at least δ, satisfying (a) and (b), or otherwise it is the empty set.

Note that there may be more than one set $f(v)$ that, for a given v, satisfies (a), (b) and (c). If this is the case, we pick arbitrarily one such set as the value of $f(v)$. We drop G and r from the notation when they are clear from the context. Since the $PDC(\delta, q)$ is the range of the function f, by definition, it contains no duplicate sets, though its elements can be highly overlapping. One way to imagine this structure is as a relaxation of a *minimum clique cover* of a graph that is the problem of determining the minimum value k such that the vertices of a graph can be partitioned into k cliques. We relax this problem by (1) relaxing the disjointness condition (we allow sets to overlap) (2) allowing also a covering with graphs of density smaller than 1.0 (cliques correspond to density value $\delta = 1.0$). Computing a clique cover of minimum size k is a well known NP-complete problem [50], and it is hard to approximate [51]. Even in this weaker form it remains NP-complete, by an easy reduction to the *maximum clique problem*. The cover we seek is *partial* since we do not insist that every vertex must be included in some set. We exclude sets that are too small (below a size threshold q) or too sparse (below a density threshold δ). The size parameter q and density parameter δ ensure that we can focus the computational effort towards those part of the graph that are more interesting (i.e. of large size and high density) with the goal of attaining computational efficiency while collecting high quality dense candidate sets. Note that for $\delta = 1.0$ the $PDC(1.0, q)$ is a subset of the set of all maximal cliques. While the set of all maximal cliques can be much larger than $|V|$, actually a worst case exponential number [43, 52], the $PDC(\delta, q)$ has always at most $|V|$ elements (and in practical cases quite fewer than that).

Algorithm *Core&Peel* in highlight

As noted above, computing a partial dense cover of a graph is a NP-complete problem. In this section we describe an efficient heuristic algorithm which is based on combining in a novel way several algorithmic ideas and procedures already presented separately in the literature. For each step we give intuitive arguments about its role

and an intuitive reason for its contribution to solving the problem efficiently and effectively. We first give a concise description of the four main phases of the *Core&Peel* algorithm. Subsequently we describe each phase in more detail.

Algorithm Overview. Phase I. Initially we compute the *Core Decomposition* of G (denoted with $CD(G)$) using the linear time algorithm in [53], giving us the core number $C(v)$ for each node $v \in V$. Moreover we compute for each vertex v in G the *Core Count* of v, denoted with $CC(v)$, defined as the number of neighbors of v having core number at least as large as $C(v)$. Next, we sort the vertices of V in decreasing lexicographic order of their core values $C(v)$ and core count value $CC(v)$.

Phase II. In Phase II we consider each node v in turn, in the order given by Phase I. For each v we construct the set $N_{C(v)}(v)$ of neighbors of v in G having core number greater than or equal to $C(v)$. We apply some filters based on simple node/edge counts in order to decide whether v should be processed in Phase III. If $|N_{C(v)}(v)| < q$ we do not process this node any more, being too small a set to start with. Otherwise we apply one of the following filters. We compute the density $\delta(v)$ of the induced subgraph $G[N_{C(v)}(v)]$. If this density is too small (i.e. $\delta(v) \leq \delta_{low}$) for a threshold δ_{low}, which we specify later, we do not process this node any more (filter (f=0)). In the second filter (f=1) we check if there are at least q nodes with degree at least $(q-1)\delta$. The third filter (f=2) is a combination of the previous two filters. Nodes that pass the chosen filter are processed in Phase III.

Phase III. In this phase we take v and the induced subgraph $G[N_{C(v)}(v)]$ and we apply a variant of the peeling procedure described in [54] that iteratively removes nodes of minimum degree in the graph. The peeling procedure stops (and reports failure) when the number of nodes drops below the threshold q. The peeling procedure stops (and reports success) when the density of the resulting subgraph is above or equal to the user defined threshold δ. The set of nodes returned by the successful peeling procedure is added to the output cover set.

Phase IV. Here we eliminate duplicates and sets completely enclosed in other sets, among those passing the Phase III. We also test the Jaccard coefficient of similarity between pairs of predicted complexes, removing one of the two predictions if they are too similar according to a user-defined threshold.

Algorithm description: details

Many of our choices rely in part on provable properties of the core number and of the peeling procedure shown in [54], and in part on the hypothesis that the peeling procedure will converge to the same dense subgraph for both notions of density, when the initial superset of nodes is sufficiently close to the final subset. However the connections between these properties, the approximation to a partial dense cover computed by the algorithm, and the properties of validated protein complexes in a PPIN network can be only conjectured. The final justification of individual choices is mainly based on the good outcome of the experimental evaluation phase.

Details on Phase I. The core decomposition of a graph $G = (V, E)$ associates to any vertex v a number $C(v)$ which is the largest number such that v has at least $C(v)$ neighbors having core number at least $C(v)$. Consider now a clique K_x of size x, for each node $v \in K_x$ its core number is $x - 1$. If K_x is an induced subgraph of G, then its core number is at least $x - 1$, thus $C(v)$ is an upper bound to the size of the largest induced clique incident to v. Consider a γ-quasi-clique $K_{(x,\gamma)}$ of x nodes, for each node v in $K_{(x,\gamma)}$ its core number is at least $\gamma(x-1)$. If $K_{(x,\gamma)}$ is an induced subgraph of G, then its core number can only be larger, thus $C(v)$ is an upper bound to the size of the largest (in terms of average degree) quasi-clique incident to v. Thus if the upper bound provided by the core number is tight, examining the nodes in (decreasing) order of their core number allows us to detect first the largest cliques (or quasi-cliques), and subsequently the smaller ones.

In a clique K_x each node is a *leader* for the clique, meaning that it is at distance 1 to any other node in the clique. Thus the first node of K_x encountered in the order computed in Phase I is always a leader. In the case of quasi-cliques of radius 1 we have by definition the existence of at least one leader node. For an isolated quasi-clique the leader node will have the maximum possible core count value, thus by sorting (in the descending lexicographic order) on the core count value we force the leader node to be discovered first in the order (assuming all nodes in the quasi-clique have the same core number). For an induced quasi-clique the influence of other nodes may increase the value of the core count for any node, but, assuming that the relative order between the leader and the other nodes does not change, we still obtain the effect of encountering the leader before the other nodes of the quasi-clique.

The core number of a node v gives us an estimate of the largest (in terms of average degree) quasi-clique (or clique) incident to v, thus it provides a very powerful filter. We employ the very simple and very efficient algorithm in [53] that computes the core decomposition of a graph in time and storage $O(|V| + |E|)$.

Details on Phase II. In Phase II we aim at computing simple conditions and we decide whether node v should be processed in the next (more expensive) phase III. The first condition to test is $|N_{C(v)}(v)| < q$, i.e. whether the number of nodes is below the user defined lower bound for the size (this is applied always). We apply then one of the following filter policies. We define the filter policy $f = 0$, by checking a sufficient condition for the existence of a clique in a dense graph based on the classical results of Turán

([48]) that guarantees the existence of a clique (or a clique with a few edges missing) in graphs with sufficiently many edges (approximately above $n^2/4$ for a graph of n nodes). This corresponds to setting $\delta_{low} = 1/2$, which indeed did perform well in our experiments with radius 1. We define the filter policy $f = 1$, by checking the necessary condition for the existence of a δ-quasi clique of at least q nodes (this condition is that $G[N_{C(v)}(v)]$ must contain at least q nodes of degree at least $(q-1)\delta$). Finally, we define the filter policy $f = 2$, that is the union of the previous two filters.

Details on Phase III. The peeling procedure we use is similar to the one described in [54]. It consists in an iterative procedure that removes a node of minimum degree and all its incident edges, and iterates on the residual graph. In [54] the graph of highest average degree constructed in this process is returned as output. We modify this procedure by returning the first subgraph generated that satisfies the density and size constraints. It is shown in [54] that this procedure is (1/2)-approximate for the *maximum average degree*, i.e. it returns a subgraph whose average degree is within a factor 1/2 of that of the subgraph of highest average degree. Empirically, we rely on the intuition that the input to the peeling procedure produced after Phase II is a superset of the target dense subgraph and that it is sufficiently tight and dense so that the peeling procedure converges quickly and the target dense subgraph is isolated effectively. We also use a novel heuristic to solve cases of ties within the peeling algorithm in [54]. When two or more vertices are of minimum degree the original peeling procedure picks one arbitrarily. In our variant we compute the sum of degrees of the adjacent nodes $S(v) = \sum_{w \in N(v)} |N(w)|$ and we select the vertex among those of minimum degree minimizing $S(.)$. This secondary selection criterion is inspired by observations in [55], where the objective is to select an independent set by iteratively removing small degree nodes, which is a dual of the problem of detecting cliques.

Details on Phase IV. In order to eliminate duplicate sets, we collect all the sets passing phase III, we split them in equal length classes and we represent them as lists of node identifiers in sorted order. Next we do a lexicographic order of each class, thus lists that are equal to each other end up as neighbors in the final sorted order and they can be easily detected and removed. In order to further exploit the sparsity of the output of phase III, we represent the collection of sets $\{\Gamma_i\}$ produced in phase III, with duplicates removed, as a graph whose nodes are the sets and elements of $\{\Gamma_i\}$. The edges represent the inclusion relation. In this graph the number of 2-paths joining nodes Γ_i and Γ_j is exactly $|\Gamma_i \cap \Gamma_j|$. If $|\Gamma_i \cap \Gamma_j| = |\Gamma_j|$, we know $\Gamma_j \subset \Gamma_i$ and we can remove Γ_j. We can count efficiently such number of 2-paths by doing a Breadth First Search at depth 2 starting from each set-node in the bipartite graph in increasing order of size, and by removing each starting node after its use. This operation allows us to compute if a set is a subset of another set, and also the Jaccard coefficient of similarity of any two non-disjoint sets.

Results and discussion

Used data and preprocessing

We used the following freely accessible data sets to test our method.

Protein protein interaction networks

Biogrid ([56]): we downloaded both Biogrid homo sapiens (BIOGRID-ORGANISM-Homo_sapiens-3.2.104.tab2.txt) and Biogrid yeast (BIOGRID-ORGANISM-Saccharomyces_cerevisiae-3.2.104.tab2.txt). String ([38]): we downloaded the general String file (protein.links.v9.05.txt.zip) and then we extracted the two subsets of interest: the homo sapiens one (related to the 9606 NCBI taxonomy id) and the yeast one (related to the 4932 NCBI taxonomy id). DIP ([57]): we downloaded the yeast db (file Scere20141001.txt).

Protein databases

From the NCBI web site we downloaded the two files for homo sapiens (Homo_sapines_gene_info.txt on 14/10/2013) and yeast (saccharomyces_cerevisiae_gene_info on 20/09/2013), the Uniprot db (uniprot_sprot.dat on 26/03/2013), and the Ensembl mapping for the associations of *ensemblproteinid* with *entrez id* for homo sapiens.

Protein complexes

We downloaded CYC2008 ([58]) and CORUM ([59]) data on 26/03/2013.

Gene Ontology (GO)

We downloaded the files for homo sapiens (gene_association. goa_human.gz) on 10/09/2014, and for yeast (gene_association.sgd.gz) on 10/09/2014

Preprocessing

Files from different sources of PPI are heterogeneous in many aspects. DIP exploits the *Uniprot accession id* (or other db entries as aliases) to represent the proteins involved in the interaction, Biogrid exploits the *NCBI entrez id* , and String uses *Ensembl proteins id* for homo sapiens and *gene locus* or *Uniprot accession* for yeast. The first operation was to represent in a uniform way the proteins for both the PPI files and the gold standard files. We decided to represent each protein with their associated NCBI entrez-id. In the process we removed possible duplications, and proteins for which the mapping was not possible. For the String data we also removed PPI with a quality score below 700. For the GO file, we identified and separated the three principal categories of the Gene

Ontology, which are Cellular Component (CC), Biological Process (BP), and Molecular Functions (MF). Following the methodology in [20], these files are filtered to remove the annotation with IEA, ND and NAS evidence codes (corresponding to the "Inferred from electronic annotation", "No biological data available" and "Non-traceable author statement", respectively). Each protein associated to an annotated function is then mapped to its NCBI entrez id. Eventual repetitions of proteins for an annotation have been removed.

Evaluation measures for protein complex prediction

In order to better capture the nuisances of matching predicted clusters with actual complexes, we use four scalar measures (one from [28], and three from [60]) and we sum them to form a single scalar *Aggregated Score* (AS). Each of the four measures differs form the others in some key aspects: some use a step-function, while other use cluster-size as weights. All four, however, aim at balancing precision and recall effects. A similar aggregation of indices has been used in [20], although we use a different pool of indices.

F-measure

From [28] we adopted the following f-measure computation to estimate the degree of matching between the found cluster and the gold standard complex. Let P be the collection of discovered clusters and let B be the collection of the gold standard complexes. For a pair of sets $p \in P$ and $b \in B$, the *precision-recall product score* is defined as $PR(p,b) = \frac{|p \cap b|^2}{|p| \times |b|}$. Only the clusters and complexes that pass a $PR(p,b)$ threshold ω (step function) are then used to compute precision and recall measures. Namely we define the matching sets: $N_p = |\{p | p \in P, \exists b \in B, PR(p,b) \geq \omega\}|$, and $N_b = |\{b | b \in B, \exists p \in P, PR(p,b) \geq \omega\}|$. Afterwards: $Pecision = \frac{N_p}{|P|}$, $Recall = \frac{N_b}{|B|}$, and the *F-measure* is the harmonic mean of precision and recall. In line with [28] and other authors we use $\omega = 0.2$. Experiments in [61] indicate that the relative ranking of methods is robust against variations of the value of ω.

From [60] we adopted three measures to evaluate the overlap between complexes and predicted clusters: the *Jaccard measure*, the *precision-recall measure* and the *semantic similarity measure*.

Jaccard measure

Let the sets P and B be as above, for a pair of sets $p \in P$ and $b \in B$, their Jaccard coefficient is $Jac(p,b) = \frac{|p \cap b|}{|p \cup b|}$. For each cluster p it is defined $Jac(p) = max_{b \in B} Jac(p,b)$, and for each complex b it is defined $Jac(b) = max_{p \in P} Jac(p,b)$. Next, we compute the weighted average Jaccard measures using, respectively, the cluster and complex sizes: $Jaccard(P) = \frac{\sum_{p \in P} |p| Jac(p)}{\sum_{p \in P} |p|}$, and $Jaccard(B) = \frac{\sum_{b \in B} |b| Jac(b)}{\sum_{b \in B} |b|}$. Finally, the *Jaccard measure* is the harmonic mean of $Jaccard(P)$ and $Jaccard(B)$.

Precision recall product

This measure is computed using exactly the same work flow as *Jaccard*, except that we replace the Jaccard coefficient with the precision-recall product score used also in [28].

Semantic similarity measure

Let the sets P and B be as above, for a protein x, we define $P(x)$ as the set of predicted clusters that contain x: $P(x) = \{p \in P | x \in p\}$, and $B(x)$ as the set of golden complexes that contain x: $B(x) = \{b \in B | x \in b\}$. Denote with $I(.)$ the indicator function of a set that is 0 for the empty set and 1 for any other set. Let $Bin(.)$ denote the set of unordered pairs of distinct elements of a set. The semantic similarity of p in B is: $Den(p,B) = \frac{\sum_{(x,y) \in Bin(p)} I(B(x) \cap B(y))}{|Bin(p)|}$. Analogously the semantic similarity of b in P is: $Den(b,P) = \frac{\sum_{(x,y) \in Bin(b)} I(P(x) \cap P(y))}{|Bin(b)|}$. Next, we compute the weighted average semantic similarity weighted respectively by cluster and complex size: $Density(P) = \frac{\sum_{p \in P} |p| Den(p,B)}{\sum_{p \in P} |p|}$, and $Density(B) = \frac{\sum_{b \in B} |b| Den(b,P)}{\sum_{b \in B} |b|}$. Finally, the *Semantic Similarity Measure* is computed as the harmonic mean of $Density(P)$ and $Density(B)$.

Handling of small protein complexes

The presence or absence of small protein complexes in the golden standard and in the outcome of the algorithms complicates the evaluation, thus in Additional file 1: Section 4 we describe a fair method for placing all algorithms on a level field with respect to this issue.

Evaluation measure for Gene Ontology coherence

For a predicted cluster $p \in P$ we compute a **q-value** score trying to assess its biological coherence and relevance. Let G be a collection of Gene Ontology annotations, and g one GO class. Let M be the set of all proteins. For a predicted cluster p, we compute the *hypergeometric* p-value $H(M,p,g)$ of the association of p to g, when $g \cap p \neq \emptyset$:

$$H(M,p,g) = \sum_{i=|p \cap g|}^{min(|p|,|g|)} \frac{\binom{|M|-|g|}{|p|-i}\binom{|g|}{i}}{\binom{|M|}{|p|}},$$

which represents the probability that a subset of M of size $|p|$ chosen uniformly at random has with g an intersection of size larger than or equal to $|p \cap g|$. As, in general, p will have an hypergeometric score for each Gene Ontology class it intersects, following [20] and [62], we associate to each p the intersecting Gene Ontology class of lower p-value. In order to correct for multiple comparisons we correct the vector of p-values using the q-value method of [63] which is a regularized version of

Table 1 Columns give: PPI name, Species (Sp.)(hs=homo sapiens, y=yeast), reference, number of proteins $|V|$, number of interactions $|E|$, average degree $\bar{d}$, and whether a quality filter (Fil.) has been applied

PPI	Sp.	Ref.	$\|V\|$	$\|E\|$	$\bar{d}$	Fil.
DIP	y	[57]	4637	21,107	9.1	No
Biogrid	y	[56]	6686	220,499	65.9	No
String	y	[38]	5590	133,082	47.6	Yes
Biogrid	hs	[56]	18,170	137,775	15.1	No
String	hs	[38]	12,717	193,105	30.3	Yes

the Benjamini Hochberg FDR estimation method. The q-values for the vector of p-values are computed via the R package provided at http://genomine.org/qvalue/.

Experiments

Basic direct measures

Basic measures on the PPINs and protein complexes data sets are reported in Table 1 and in Table 2, respectively. When we map the known curated complexes onto the PPI-networks we obtain 5 different data sets in which the number and density of the embedded complexes is specific to the involved PPIN (see Table 3). The resulting embedded complexes have variable density. We report in Table 3 the 90 % and the 50 % density percentiles. One of the assumptions we have used in our algorithm is that for each embedded complex there is one vertex that is linked to (almost) all the other nodes in the embedded complex (egocentricity). This is an important property that measures on the actual data support (see Table 4). In Table 5 we report on the degree of overlap among complexes by counting the number of proteins belonging to one, two, three or more than three complexes. This is an important feature of the prediction problem since algorithms need to handle properly overlapping clusters. Human complexes have higher overlap rates than yeast complexes. In (Additional file 1: Section 6) we report the distributions of basic measures relative to the graph (degree, core number, clustering coefficients), and to the embedded PC (size, average degree, density).

Table 2 Columns give: name of the data set, Species (Sp.)(hs=homo sapiens, y=yeast), reference, total number of complexes, number of complexes of size 3 or larger, number of complexes of size up to 2, total number of proteins covered by the complexes

Name	Sp.	Ref.	# complexs	# compl. size >2	# compl. size ≤ 2	# proteins
CYC2008	y	[58]	408	236	172	1627
CORUM	hs	[59]	1750	1257	493	2506

Table 3 Columns give: Name of the PPI and complex data set, number of complexes of size ≥ 3, min size, max size, average size, number of complexes with density δ greater than 0.9 and 0.5

Name	# CX	Min	Max	Mean	$\delta > 0.9$	$\delta > 0.5$
DIP-CYC2008	226	3	40	6.02	60 (25 %)	131 (55 %)
BioGrid-CYC2008	236	3	81	6.67	173 (73 %)	223 (94 %)
String-CYC2008	236	3	81	6.67	220 (93 %)	235 (99 %)
BioGrid-CORUM	1257	3	143	6.12	516 (41 %)	943 (75 %)
String-CORUM	1188	3	133	6.07	621 (52 %)	981 (82 %)

Quality testing

We report the comparative evaluation of our algorithm vs several other algorithms, among those considered state-of-the-art. We used for these experiments an Intel core i7 processor (4 cores) at 2.6 GHz, with 16 Gb RAM memory, and with Mac OS X 10.8.5.

We have selected 10 algorithms, namely: MCL, Coach, MCODE, CMC, MCL-CAW, ProRank+, SPICi, ClusterOne, RNSC, and Cfinder among those in literature. A brief description of each is in Additional file 1: Section 1. In the selection we applied these criteria: (a) we selected algorithms that appeared in several surveys and comparative evaluations, and well cited in the literature; (b) we included both old classical algorithms and more recent ones; (c) we have included algorithms using definitions of density similar to the one we adopt; (d) we included algorithms with available implementation in the public domain or obtainable from the authors upon request; (e) we preferred implementations based on widely available (i.e. non-proprietary) platforms; (f) we avoided algorithms that make use of additional biological annotations (e.g. gene expression data); (g) we preferred methods with a clear and unique underlying algorithm (e.g. "ensemble" methods are not included); (h) we preferred methods that aim at "protein complex detection" vs. those that aim at "functional module discovery", since the evaluation methodologies for these two classes are quite different, although many methods could be construed as dual-use.

Table 4 Columns give: name of the PPI and complex data set, number of complexes of size ≥ 3, number of complexes with at least one center at distance 1 ($r = 1$) for a fraction of at least 0.9 of its size and at least 0.5 of its size. Similar data for a center at distance 2, ($r = 2$)

Name	# CX	r1 >0.9	r1 >0.5	r2 >0.9	r2 >0.5
DIP-CYC2008	226	131 (55 %)	197 (83 %)	163	212
BioGrid-CYC2008	236	216 (91 %)	234 (99 %)	234	236
String-CYC2008	236	235 (99 %)	236 (100 %)	236	236
BioGrid-CORUM	1257	891 (70 %)	1162 (92 %)	1176	1246
String-CORUM	1188	923 (77 %)	1139 (95 %)	1085	1188

Table 5 Columns give: name of the PPI and complex data set, number of proteins covered by some complex, number of protein covered by one, two, three or more then three complexes

Name	# P	1 CX	2 CX	3 CX	> 3 CX
DIP-CYC2008	1175	1005 (86 %)	134 (11 %)	23 (2 %)	13 (1 %)
BioGrid-CYC2008	1342	1166 (87 %)	139 (10 %)	24 (2 %)	13 (1 %)
String-CYC2008	1341	1165 (87 %)	139 (10 %)	24 (2 %)	13 (1 %)
BioGrid-CORUM	2227	909 (41 %)	483 (21 %)	233 (11 %)	602 (27 %)
String-CORUM	2067	852 (42 %)	430 (20 %)	217 (10 %)	568 (28 %)

Each method has its own pool of parameters to be set. For the quality score shown in Section Evaluation measures for protein complex prediction we have considered for each method an extensive range of input parameter values (see File Additional file 1: Section 2 and 3) and we selected for each quality measure used in the Aggregated Score the best result obtained. Note that each best value for the four base quality measures may be obtained with slightly different values of the control parameters. Missing measures indicate that, for a specific algorithm and data set, the computation would not complete within a reasonable amount of time (without any sign of progress) or it generated fatal runtime errors.

Comparative evaluation

Performance of protein complex prediction

Figures 1, 2, 3, 4 and 5 report the F-measure, the Semantic Similarity, the J-measure, the PR-measure and the Aggregated Score (as defined in Section 'Evaluation measures for protein complex prediction') for three data sets relative to yeast PPIN (DIP, Biogrid and String). Out of 15 measurements, *Core&Peel* has the best value in 12 cases, CMC in 2 cases, and ClusterOne in 1 case. The Aggregated Score, which balances strong and weak points of the four basic measures, indicates that *Core&Peel*, CMC and ClusterOne have about the same performance for the medium-size PPI newtwork DIP. But for Biogrid data and even more for String data *Core&Peel* takes the lead, even with a wide margin.

Figures 6, 7, 8, 9 and 10 report the F-measure, the Semantic similarity, the J-measure, the PR-measure and the aggregated score for three data sets relative to homo sapiens PPI (Biogrid and String). During the evaluation of the predicted clusters for Biogrid data we realized that the Biogrid PPI network had one node of very high degree corresponding to the Ubiquitin (UBC) protein. This fact has a straightforward biological explanation. Since UBC is involved in the degradation process of other proteins, UBC is linked to many other proteins at a certain time in their life-cycle. Given this special role of UBC, when protein degradation is not the main focus of the intended investigation, it may be convenient to consider also the same PPI network with the UBC node and its incident edges removed (Rolland et al. in [64] also remove interactions involving UBC in their high quality human PPIN.) We labelled this graph *BG-hs-UBC*. We tested also the other PPI network used in our study and this is the only case in which removing a node of maximum degree changes significantly the outcome of the prediction. Out of 15 measures, *Core&Peel* has the best value in all 15 cases. Good performance is obtained on some measures by CMC and Spici.

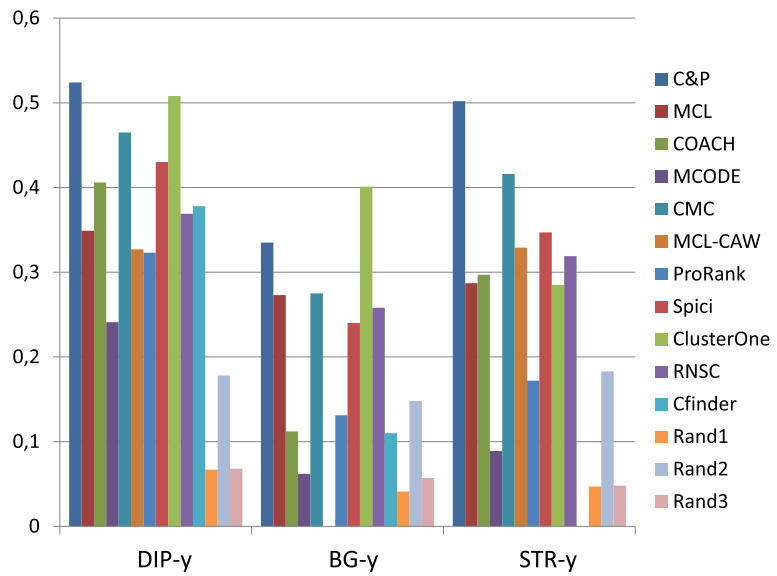

Fig. 1 F-measure score for 11 algorithms and 3 random baselines on yeast data. Runs optimizing the f-measure for each algorithm

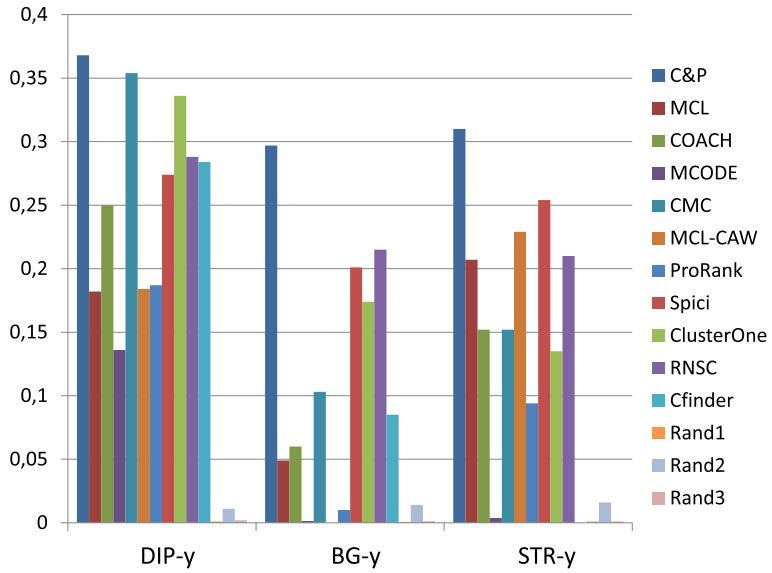

Fig. 2 Semantic similarity score for 11 algorithms and 3 random baselines on yeast data. Runs optimizing the ss-measure for each algorithm

It is interesting to notice how the algorithms perform differently on the BG-hs with and without UBC. On Biogrid data without UBC, *Core&Peel*, Spici and ClusterOne improve their AS value, while RNSC and COACH have a reduced AS value. The improvement in absence of UBC can be easily explained by the fact that UBC appears only in a few complexes of the golden standard, thus the evaluation phase is made more precise by its removal from the network and thus from the predicted clusters. The better results attained by RNSC and COACH on the graph with UBC may be a hint that, for these two approaches, the presence of UBC helps in homing in more quickly on the true complexes hidden in the graph.

We include as a sanity check also three random predictions (Rand1, Rand2, and Rand3). The purpose of this check is to assess how well the measure we are using are able to discriminate the predictions on real data sets from those generated randomly by generators allowed to access some partial knowledge about the structure of the golden standard.

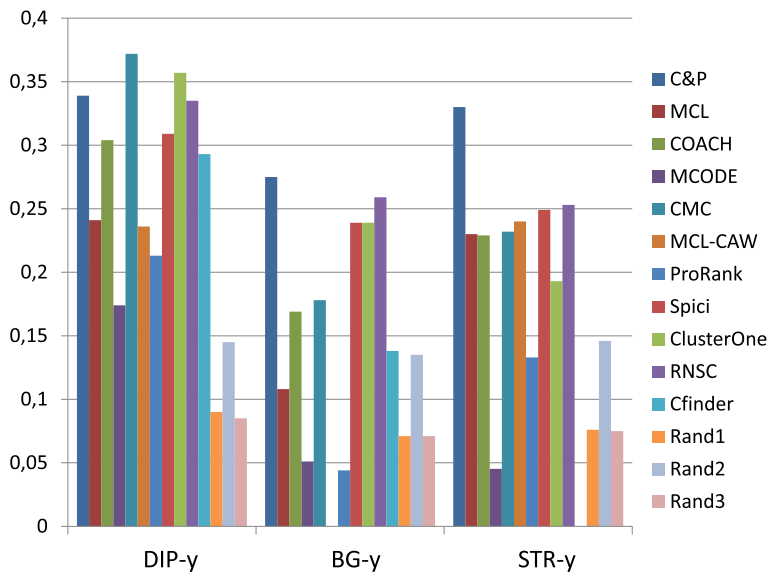

Fig. 3 J-measure score for 11 algorithms and 3 random baselines on yeast data. Runs optimizing the J-measure for each algorithm

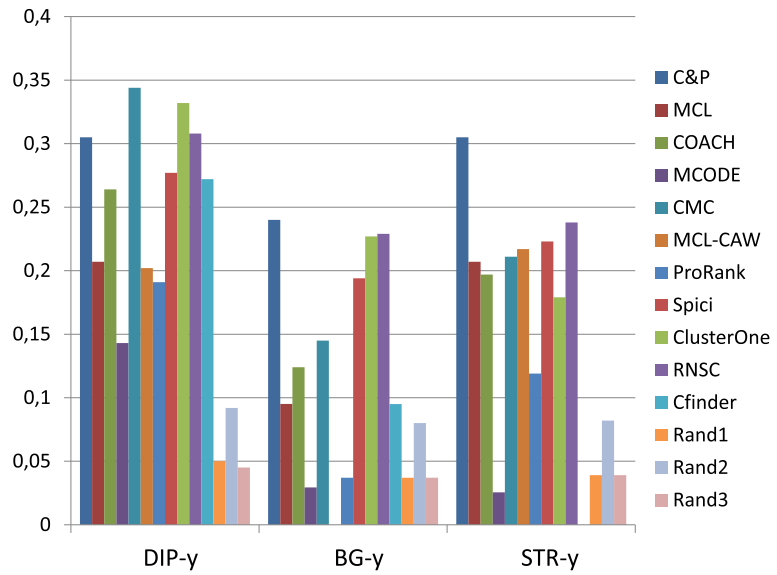

Fig. 4 PR-measure score for 11 algorithms and 3 random baselines on yeast data. Runs optimizing the PR-measure for each algorithm

The method Rand1 is given the size distribution of the sets in the golden standard and produce a random collection of sets out of the vertices of the PPI with the same size distribution. The method Rand2 is as Rand1 except that the random sets are generated starting from the subset of all vertices in the PPI that belong to some complex in the golden standard. The method Rand3 is obtained by taking the golden standard and applying to it a random permutation of the nodes of the PPI. Note that this approach besides preserving the size distribution preserves also the distribution of the size of the intersections of any number of sets of the golden standard.

In terms of performance, Rand1 behaves almost like Rand3, while Rand2 (having stronger hints) attains better results. The semantic similarity measure is the one that has better discrimination power vs all the three random test cases.

Core&Peel has better SS performance on all the 6 PPIN tested than the 10 competing methods. Semantic similarity is the only measure that explicitly places a premium

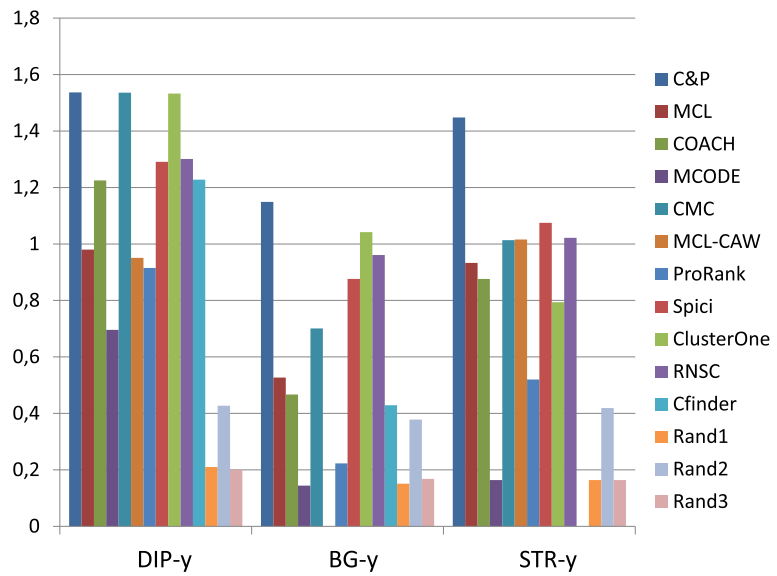

Fig. 5 Aggregated score for 11 algorithms and 3 random baselines on yeast data

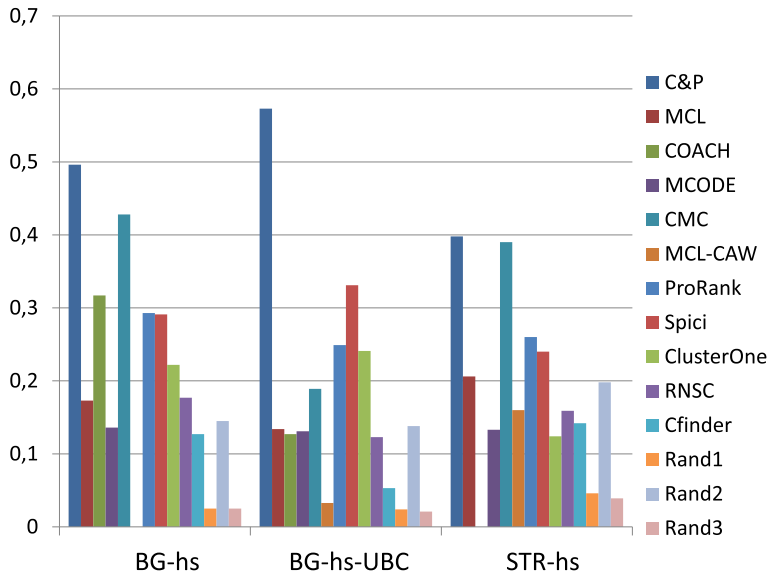

Fig. 6 F-measure score for 11 algorithms and 3 random baselines on homo sapiens data. Runs optimizing the f-measure for each algorithm

in correctly identifying the proteins that simultaneously belong to multiple complexes, thus we can infer that *Core&Peel* successfully uncovers the overlapping structure of the the known protein complexes.

Coherence with Gene Ontology annotation

The second index is the number of predicted clusters with an associated functional annotation (Biological Processes (BP) of Gene Onontology (GO)) below a given False Discovery Rate (FDR) threshold. Note that here we use a non-normalized measure (absolute count) since we want to favor algorithms with a rich high quality output. We are safeguarded against rewarding unduly methods that inflate their output since we operate each algorithm with the parameters that optimize the (balanced) F-measure. Moreover, even though none of the methods we use incorporates GO as part of its model, it is relatively safe to assume that, in most cases of interest, GO annotations are indeed available and may be used for a post-processing re-ranking or filtering of the predictions.

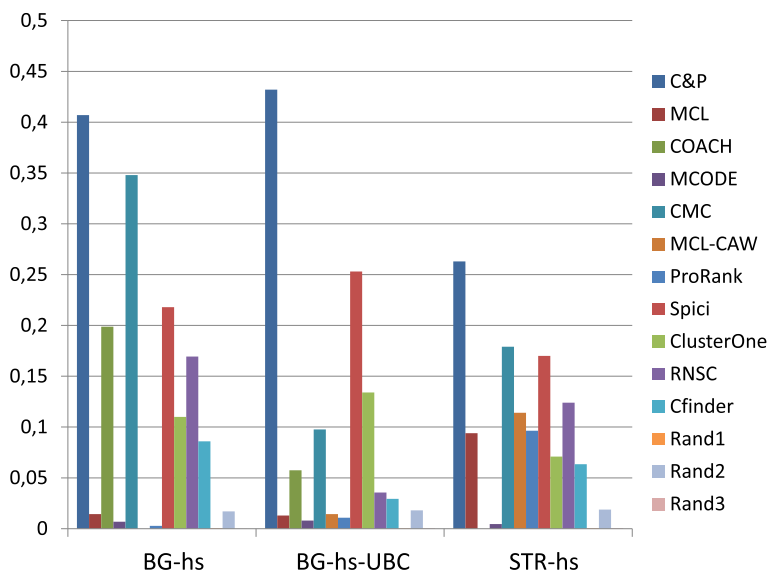

Fig. 7 Semantic similarity score for 11 algorithms and 3 random baselines on homo sapiens data. Runs optimizing the SS-measure for each algorithm

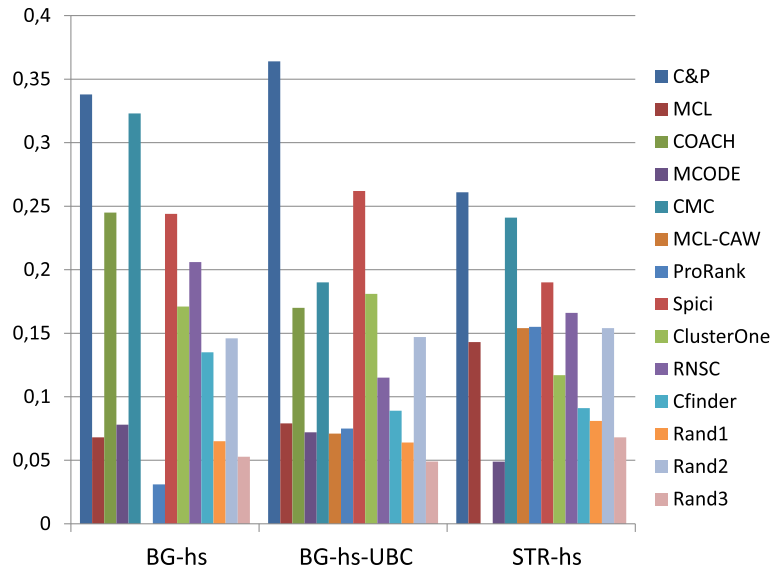

Fig. 8 J-measure score for 11 algorithms and 3 random baselines on homo sapiens data. Runs optimizing the J-measure for each algorithm

The biological function enrichment measure (using the BP annotation in GO) is shown in Figs. 11, 12, 13, 14, 15, and 16. We used in abscissa the FDR thresholds ranging from 10^{-2} to 10^{-7} on the q-value.

Core&Peel has a larger or equal absolute number of high quality predictions below q-value 10^{-3} than the competing methods on five data sets out of six. For the BG-hs-UBC dataset *Core&Peel* leads below q-value 10^{-4}. The overall trend is fairly consistent for all the six data sets tested.

Table 6 reports examples of predicted clusters with a notable low p-value, and the corresponding GO class. The top 10 predicted clusters we identify have p-value for their prevalent GO-annotation (all distinct) in the range $10^{-191} - 10^{-72}$. For a comparison, the top 10 functional modules detected by the recent method ADM (Adaptive Density Modularity) of Shen et al. ([65] Table 3) relative to the same GO BP annotation have p-values in the range $10^{-63} - 10^{-28}$.

Robustness against noise in the PPIN graph

We have tested our method for its robustness against injection of random noise in the input network. Starting with the the Biogrid HS network we have added randomly

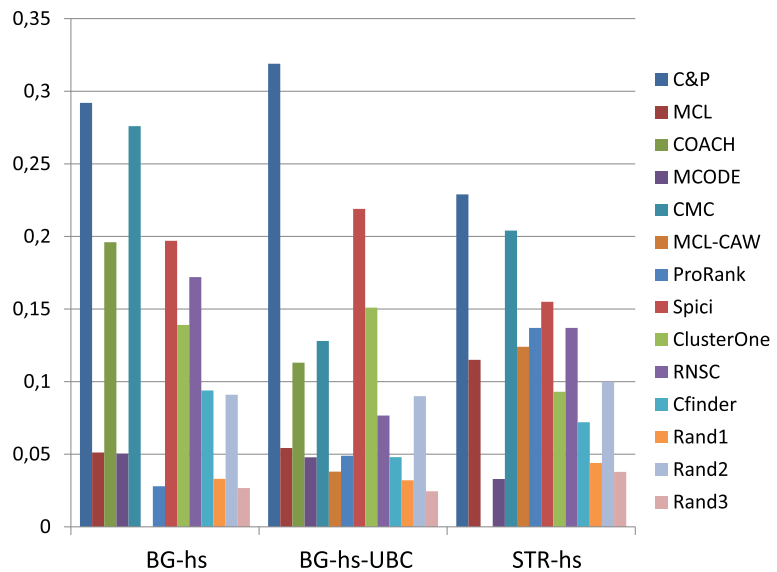

Fig. 9 PR-measure score for 11 algorithms and 3 random baselines on homo sapiens data. Runs optimizing the PR-measure for each algorithm

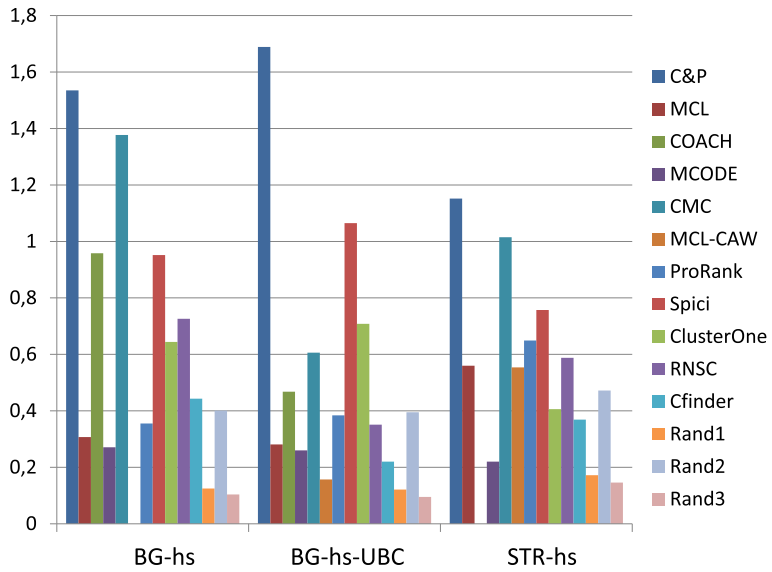

Fig. 10 Aggregated score for 11 algorithms and 3 random baselines on homo sapiens data

additional (noise) edges for a number of additions ranging from 5 to 25 % of the initial number of edges in steps of 5 %. We have generated 10 networks for each class and taken the mean value of the 4 basic quality indices of Section 'Evaluation measures for protein complex prediction'. The results are remarkably robust showing for three indices no variation up to the fourth decimal digit, and for the f-measure a variability of 0,001 across the range of noise values. Further tests with large random graphs are described in Additional file 1: Section 7, where we use the two stage multiple hypothesis test proposed in [66, 67] to bound the false discovery rate (FDR) associated with the identified complexes.

Running times

Figures 17, 18, 19, 20, 21, and 22 report in logarithmic scale the running times (seconds) for the 11 algorithms on the six data sets, with the parameters optimizing the f-measure. For MCL-caw we report the post-processing time in the graphic, thus a timing comparable with those

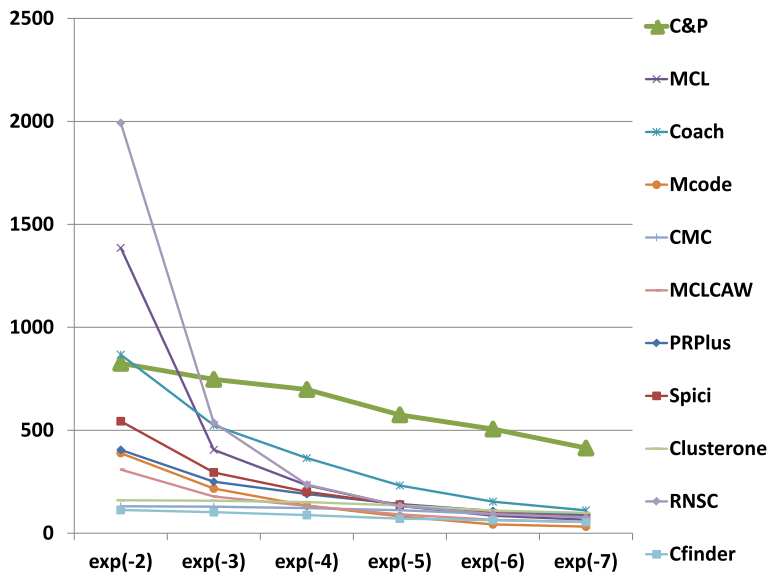

Fig. 11 Number of predicted clusters with GO enrichment q-value below threshold, as a function of the threshold, for DIP yeast data

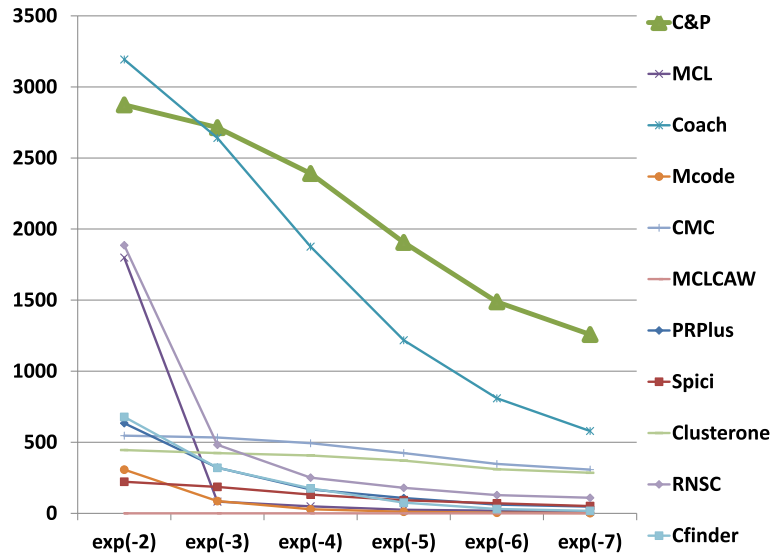

Fig. 12 Number of predicted clusters with GO enrichment q-value below threshold, as a function of the threshold, for Biogrid yeast data

of the other methods requires adding the MCL datum. Spici is the fastest method on all the data sets, often completing in less than a second. *Core&Peel* comes second in speed in all the data set (except for DIP where it is third).

Some predictions with support in the literature

In the long run the effectiveness of a protein-complex prediction method hinges upon its capability to uncover interesting and unexpected new phenomena of biological relevance. As an intermediate step we report on predictions made by *Core&Peel* that involve a known complex and one or two additional proteins, for which there is evidence of a biological function in the literature. We take the clusters detected by *Core&Peel* in the Biogrid and String homo sapiens network, we rank them by the highest value of the Jaccard correlation coefficient (jacc) and the semantic similarity (SS) with a matching known complex, we analyze the discrepancies (i.e. proteins non listed in the known complex but with a large number of PPI connections to it) and we highlight the literature supporting the functional relevance of the interaction. We chose here the parameter setting maximizing the f-measure.

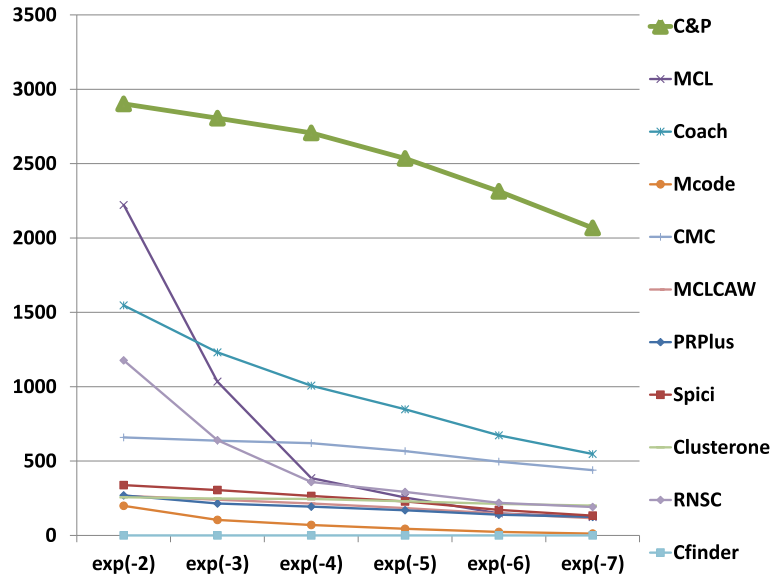

Fig. 13 Number of predicted clusters with GO enrichment q-value below threshold, as a function of the threshold, for String yeast data

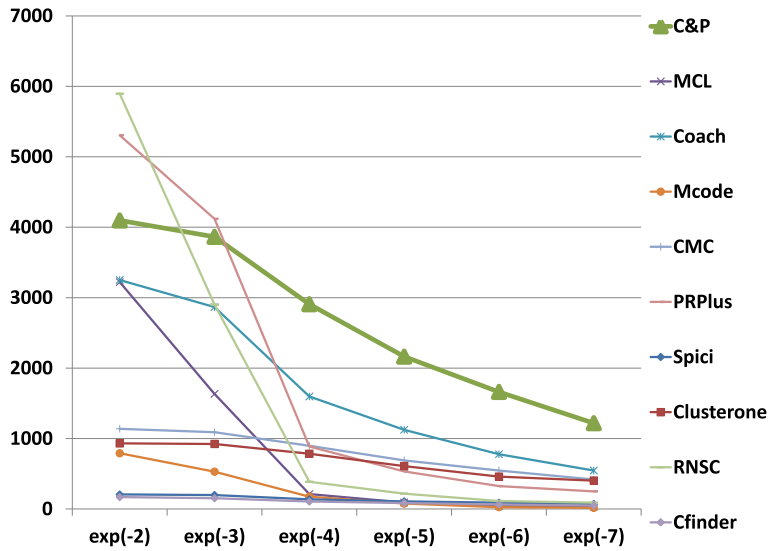

Fig. 14 Number of predicted clusters with GO enrichment q-value below threshold, as a function of the threshold, for Biogrid Homo sapiens data

Case 1 (rank=2 , jacc=0.875) This predicted cluster matches almost perfectly with the **20S proteasome** complex (Corum-id 191). Moreover the predicted cluster includes two additional proteins: **UBC** (Ubiquitin C) and **IQCB1** (IQ motif containing B1). The *hitpredict database* (http://hintdb.hgc.jp/htp/) also predicts high quality interactions between **IQCB1** and six **PSA** plus three **PSB** proteins. The functional connection of **UBC** and the **proteasome complex** within the protein degradation pathway is also well known (see e.g. [68]).

Case 2 (rank=5, jacc =0.875) This predicted cluster matches almost perfectly with complex **TFIIH** (transcription factor complex TFIIH)(Corum-id 5495). Moreover the predicted cluster includes an additional protein: **AR** (androgen receptor). Indeed the phosphorilation action of **TFIIH** upon **AR** is reported in [69].

Case 3 (rank=9, jacc=0.833) This predicted cluster matches almost perfectly with the **TFIIIC** containing complex (corum-id 1105), (and also with TFIIIC2,

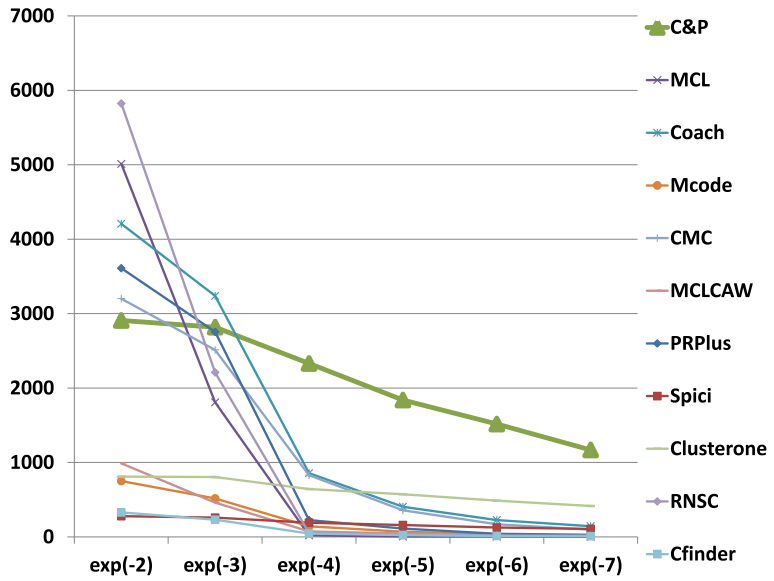

Fig. 15 Number of predicted clusters with GO enrichment q-value below threshold, as a function of the threshold, for Biogrid Homo sapiens data (UBC removed)

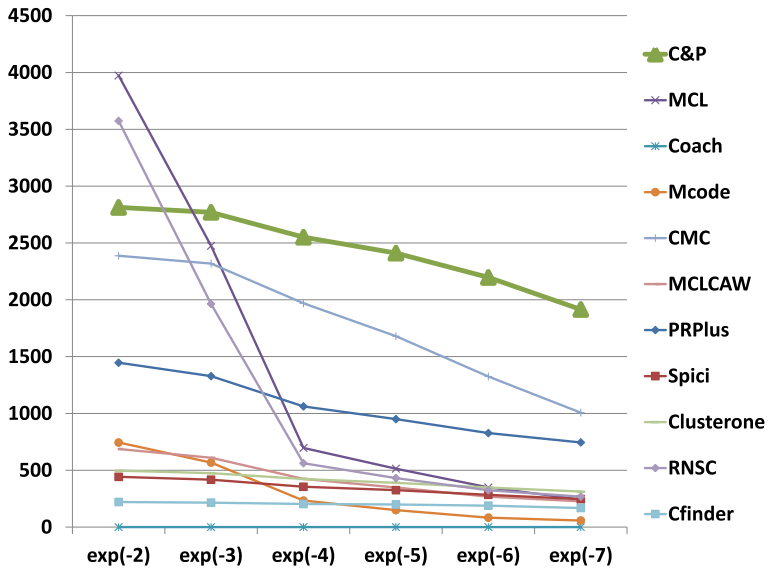

Fig. 16 Number of predicted clusters with GO enrichment q-value below threshold, as a function of the threshold, for String Homo sapiens data

a second TFIIIC containing complex, corum-id 1101). Moreover the predicted cluster includes an additional protein: **GTF3C6** (general transcription factor IIIC, polypeptide 6). Dumay et al. [70] identified a sixth human **TFIIIC subunit**, specifically **GTF3C6**, which corresponds to a previously uncharacterized 213-amino acid human protein (C6ORF51).

Case 4 (rank=14, jacc=0.823) This predicted cluster matches almost perfectly with the **20S proteasome** complex (Corum-id 191). Moreover the predicted cluster includes two additional proteins: PSMB8 (Proteasome subunit beta type-8) and **POMP** (proteasome maturation protein). The protein encoded by the **POMP** gene is a molecular chaperone that binds the **20S preproteasome** components and it is essential for **20S proteasome** formation. The **POMP** protein is degraded before the maturation of the **20S proteasome** is complete. A mutation in the 5' UTR of this gene has been associated with KLICK syndrome, a rare skin disorder ([71]).

Case 5 (rank 15, jacc = 0.818) This predicted cluster matches almost perfectly with PA700 complex (**26S protease/19S protease**) (corum-id 32). Moreover the predicted cluster includes an additional protein: **UCHL5**

Table 6 Data set STRING-CORE (homo sapiens). Functionally enriched clusters found with min size 8 and filtering policy 1

GO annotation	Type	GO id	Clust. size	Inters.	Class size	Hyp. *p*-value
mRNA splicing, via spliceosome	BP	GO:0000398	91	90	183	8.23E–191
G2/M transition of mitotic cell cycle	BP	GO:0000086	52	52	121	5.62E–120
Mitotic cell cycle	BP	GO:0000278	68	68	384	6.24E–118
G-protein coupled receptor signaling pathway	BP	GO:0007186	126	97	918	1.16E–101
Type I interferon signaling pathway	BP	GO:0060337	33	33	65	4.48E–86
O-glycan processing	BP	GO:0016266	30	30	55	7.44E–81
Platelet degranulation	BP	GO:0002576	32	32	82	4.14E–79
Interferon-gamma-mediated signaling pathway	BP	GO:0060333	30	30	70	1.33E–76
Extracellular matrix organization	BP	GO:0030198	40	40	272	2.26E–75
Transferrin transport	BP	GO:0033572	24	24	32	2.40E–72

We report the top ten clusters by hypergeometric *p*-value. Each row reports: the GO annotation class, GO class type (BP=Biological Process), the GO id, the size of the cluster, the size of the intersection, the size of the functional class, and the hypergeometric *p*-value

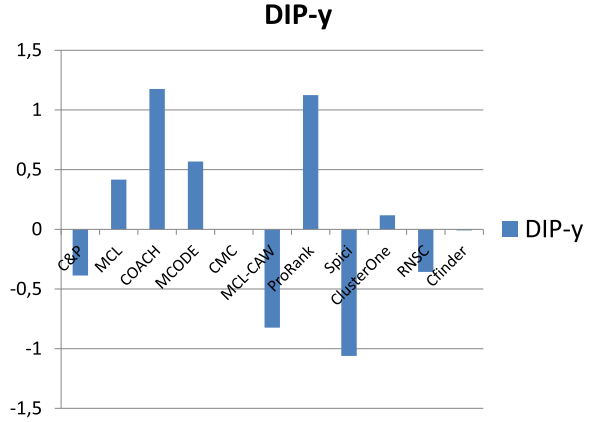

Fig. 17 Time (in seconds) in Log10 scale for DIP data. Runs optimizing the f-measure for each algorithm

(ubiquitin carboxyl-terminal hydrolase L5). Interestingly, Darcy et al. [72] report that a small molecule (b-AP15) inhibits the activity of two **19S** deubiquitinases regulatory particles: ubiquitin C-terminal hydrolase 5 (**UCHL5**) and ubiquitin-specific peptidase 14 (USP14), resulting in accumulation of polyubiquitin, which in turn induces tumor cell apoptosis. Thus Darcy et al. suggest that the deubiquitinating activity of this regulatory molecule may form the basis for a new anticancer drug.

Case 6 (rank=16, jacc=0.818) This predicted cluster matches almost perfectly with BAF complex (Synonyms: **SWI/SNF** complex A) (corum-id 1251 and 1237). Moreover the predicted cluster includes two additional proteins: **BCL7B** (B cell CLL/lymphoma 7B) and ARID1B (AT rich interactive domain 1B SWI1-like). A connection of the **SWI/SNF** complex with the first protein (**BCL7B**) is described in [73] where it is reported a proteomic analysis of endogenous **mSWI/SNF** complexes, which identified several new dedicated stable subunits of **SWI/SNF** complexes, including, among others, **BCL7B**.

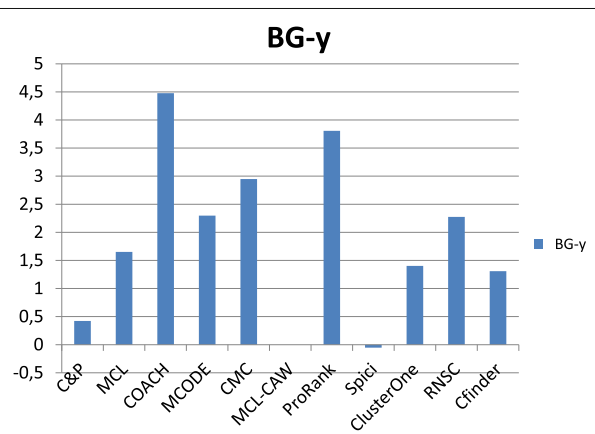

Fig. 18 Time (in seconds) in Log10 scale for Biogrid yeast data. Runs optimizing the f-measure for each algorithm

Case 7(rank=20, jacc=0.8) This predicted cluster matches almost perfectly with corum-id 1097, **eIF3** complex (which is made of 13 proteins). Moreover the predicted cluster includes an additional protein: **Gag-Pol** [Human immunodeficiency virus 1]. It is reported in [74] that "A conserved structure within the HIV **gag** open reading frame that controls translation initiation directly recruits the 40S subunit and **eIF3**".

Case 8 (rank 27, ss=0.952) This predicted cluster matches almost perfectly with **SAP complex** (Sin3-associated protein complex) (corum id 591). Moreover the predicted cluster includes an additional protein: **ING2** (inhibitor of growth family, member 2). It is reported in [75] that "Besides the paralogous proteins, including HDAC1/HDAC2, mSin3A/mSin3B, and the histone-interacting RbAp46/RbAp48 proteins, the mammalian Rpd3L/Sin3L complex comprises at least five other subunits, including **SAP30**, Sds3, SAP180/RBP1, SAP130, and ING1b/**ING2**, whose precise roles at the molecular level are poorly understood but most likely involve targeting the complex to specific genomic loci via one or more interaction surfaces".

Case 9 (rank 28, ss= 0.92) This predicted cluster matches almost perfectly with the **Ribosome** complex (corum-id 306). Moreover the predicted cluster includes an additional protein: SIRT7 (**Sirtuin 7**). Tsai et al. [76] investigate the role of **Sirtuin 7** in the **Ribosome** biogenesis and protein synthesis.

Case 10 (String-hs data, rank=10, jacc=0.916) This predicted cluster matches almost perfectly with the **Exosome** (11 prot) complex (corum-id 789). Moreover the predicted cluster includes an additional protein: **XRN1** (5'-3' exoribonuclease 1). Li et al. [77] describe the competing role played by **XRN1** and the **Exosome** complex in Hepatitis C-Virus RNA decay.

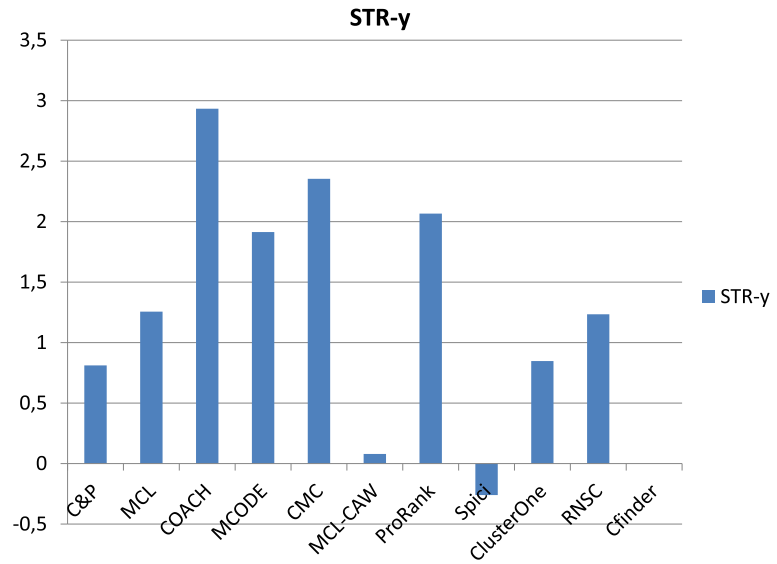

Fig. 19 Time (in seconds) in Log10 scale for String yeast data. Runs optimizing the f-measure for each algorithm

Conclusions

The experimental results reported in Section 'Experiments' show that *Core&Peel* is remarkably consistent in finding known complexes across 1 medium and 5 large data sets, ranking first in aggregated score against ten state-of-the-art methods in all 6 cases (CMC is second trice; SPICi twice, and Clusterone once).

Core&Peel also leads in the ability to produce cluster predictions that are highly consistent with GO-BP annotations. The specific complex-protein interaction predictions listed in Section 'Some predictions with support in the literature' have all a strong support in the literature. Although such predictions may not always correspond to actual complexes, they do indeed point at functionally relevant phenomena.

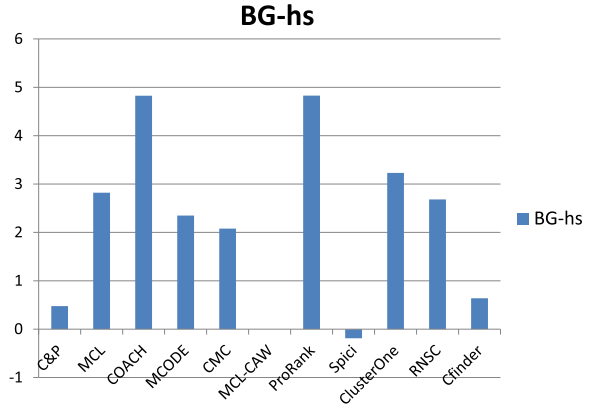

Fig. 20 Time (in seconds) in Log10 scale for Biogrid homo sapiens data. Runs optimizing the f-measure for each algorithm

The *Core&Peel* algorithm exploits properties of complexes embedded in PPINs (egocentricity, density) that are more evident the larger the PPINs become, and it does not suffer from phenomena of combinatorial explosion (as both the theoretical analysis and the empirical running time attest). Thus we believe that *Core&Peel* can become a method of choice when even larger PPINs are built and analyzed, such as those arising in multi-species PPIN studies (see [37]) and those arising in immunology studies (see [40]).

Core&Peel is fast and easy to use, requiring the setting of very few natural parameters relative to the minimum size, density and separation of the target complexes. Indeed, having a small sample of the type of complexes to be sought, these parameters can be extracted directly form the sample.

Core&Peel uses very little biological information except that embedded in the PPIN topology. Thus we believe further gains can be achieved by augmenting our scheme with the ability to handle PPINs endowed with edge weights modeling, for example, PPI quality, or other types of a priori knowledge), or by incorporating GO annotations-based filters within the basic algorithmic framework (See discussion in Additional file 1: Sec 5). Improvements and tests along these lines are left for future research.

In this paper our main focus is to compare our proposed algorithm versus 10 competing algorithms on a sufficiently diverse pool of test data (2 species, 3 repositories, 6 PPIN, 2 PC golden standard sets) so to gain

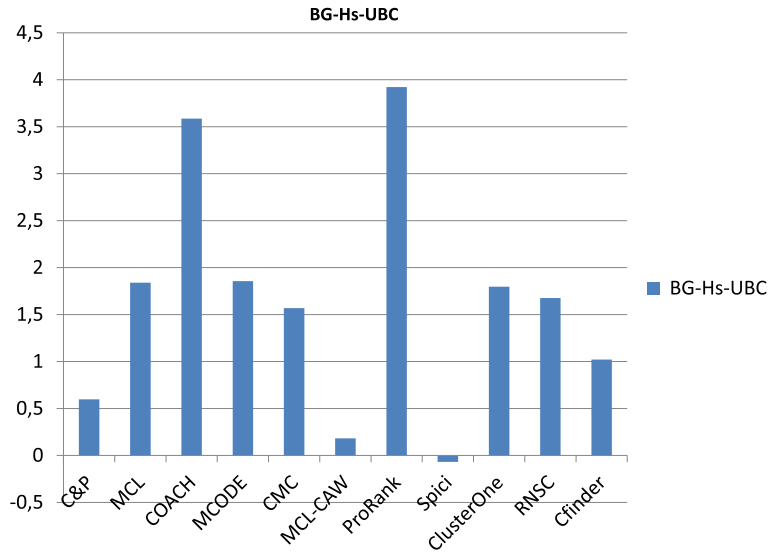

Fig. 21 Time (in seconds) in Log10 scale for Biogrid homo sapiens data without UBC. Runs optimizing the f-measure for each algorithm

confidence in the robustness of the main thesis (i.e. the suitability of *Core&Peel* for discovering PC in large PPIN). We do not aim at suggesting that a particular type of PPIN repository should be preferred over others, and we do not even aim at implying that one should always use large PPIN in place of smaller ones (see e.g. [78]). Both questions are worthy of attention but fall outside the scope of the present article. The choice of the PPI data to be used for a given study is a non-trivial choice since many hidden biases could be implicit in the data (due both to its experimental origin, and to subsequent filtering) [79], thus these issues should be considered carefully at the initial stage of any experimental design.

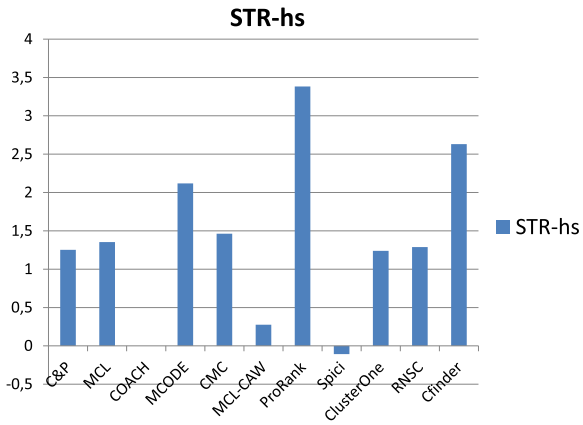

Fig. 22 Time (in seconds) in Log10 scale for String homo sapiens data. Runs optimizing the f-measure for each algorithm

Additional file

Additional file 1: Detailed Experimental settings. Protein complex prediction for large protein protein interaction networks with the *Core&Peel* Method- Supplementary Materials. Description of the parameters and settings used in testing the competing sw for PC detection in Large PPIN. Description of data sets features distributions. Analysis of Asymptotic Complexity for *Core&Peel*. (PDF 1284 kb)

Acknowledgments
The authors would like to thank the participants to the NII Shonan Meeting "Towards the ground truth Exact algorithms for bio-informatics research", January 17-20, 2014, for their support in the early stages of this research.

Declarations
This article has been published as part of BMC Bioinformatics Vol 17 Suppl 12 2016: Italian Society of Bioinformatics (BITS): Annual Meeting 2015. The full contents of the supplement are available online at http://bmcbioinformatics.biomedcentral.com/articles/supplements/volume-17-supplement-12.

Funding
Work and publication costs supported by Italian Ministry of Education, Universities and Research (MIUR) and by the National Research Council of Italy (CNR) within the Flagship Project *InterOmics* PB.P05. (CUP B91J12000270001).

Availability of data and materials

- Project name: Core&Peel
- Project home page: http://bioalgo.iit.cnr.it/index.php?pg=ppin
- Operating system(s): Platform independent
- Programming language: Python
- Other requirements: None
- License: Lesser General Public License (LGPL)
- Any restrictions to use by non-academics: None, Core&Peel is a web application free and open to all users.
- Data used for this study is available at http://bioalgo.iit.cnr.it/index.php?pg=ppin

Authors' contributions
MP devised the algorithm and its analysis, drafted the manuscript, and exercised general supervision. MB collected the data, wrote the evaluation

code and performed experiments. FG performed experiments and set up the web site interface. All authors read and approved the manuscript and contributed extensively to the work presented in this paper.

Competing interests
The authors declare that they have no competing interests.

Consent for publication
Not applicable.

Ethics approval and consent to participate
No ethical approval or consent was required for this study.

Published: 8 November 2016

References

1. Srihari S, Leong HW. A survey of computational methods for protein complex prediction from protein interaction networks. J Bioinform Comput Biol. 2013;11(2):1230002.
2. Dittrich MT, Klau GW, Rosenwald A, Dandekar T, Müller T. Identifying functional modules in protein protein interaction networks: an integrated exact approach. Bioinformatics. 2008;24(13):223–31. doi:10.1093/bioinformatics/btn161.
3. Wang H, Kakaradov B, Collins SR, Karotki L, Fiedler D, Shales M, Shokat KM, Walther TC, Krogan NJ, Koller D. A complex-based reconstruction of the saccharomyces cerevisiae interactome. Mol Cellular Proteomics. 2009;8(6):1361–81. doi:10.1074/mcp.M800490-MCP200. http://www.mcponline.org/content/8/6/1361.full.pdf+html.
4. Ji J, Zhang A, Liu C, Quan X, Liu Z. Survey: Functional module detection from protein-protein interaction networks. Knowl Data Eng IEEE Trans. 2014;26(2):261–77.
5. Yu H, Paccanaro A, Trifonov V, Gerstein M. Predicting interactions in protein networks by completing defective cliques. Bioinformatics. 2006;22(7):823–9. doi:10.1093/bioinformatics/btl014.
6. Jansen R, Yu H, Greenbaum D, Kluger Y, Krogan NJ, Chung S, Emili A, Snyder M, Greenblatt JF, Gerstein M. A bayesian networks approach for predicting protein-protein interactions from genomic data. Science. 2003;302(5644):449–53. doi:10.1126/science.1087361. http://www.sciencemag.org/content/302/5644/449.full.pdf.
7. Sharan R, Ulitsky I, Shamir R. Network-based prediction of protein function. Mol Syst Biol. 2007;3(1):. doi:10.1038/msb4100129. http://msb.embopress.org/content/3/1/88.full.pdf.
8. Ideker T, Sharan R. Protein networks in disease. Genome Res. 2008;18(4):644–52. doi:10.1101/gr.071852.107. http://genome.cshlp.org/content/18/4/644.full.pdf+html.
9. Yu L, Huang J, Ma Z, Zhang J, Zou Y, Gao L. Inferring drug-disease associations based on known protein complexes. BMC Med Genomics. 2015;8(Suppl 2):2.
10. Bader G, Hogue C. An automated method for finding molecular complexes in large protein interaction networks. BMC Bioinformatics. 2003;4(1):2. doi:10.1186/1471-2105-4-2.
11. King AD, Pržulj N, Jurisica I. Protein complex prediction via cost-based clustering. Bioinformatics. 2004;20(17):3013–20. doi:10.1093/bioinformatics/bth351.
12. Adamcsek B, Palla G, Farkas IJ, Derényi I, Vicsek T. Cfinder: locating cliques and overlapping modules in biological networks. Bioinformatics. 2006;22(8):1021–3. doi:10.1093/bioinformatics/btl039.
13. Van Dongen S. Graph clustering via a discrete uncoupling process. SIAM J Matrix Anal Appl. 2008;30(1):121–41. doi:10.1137/040608635.
14. Wu M, Li X, Kwoh CK, Ng SK. A core-attachment based method to detect protein complexes in ppi networks. BMC Bioinformatics. 2009;10:1–16.
15. Liu G, Wong L, Chua HN. Complex discovery from weighted ppi networks. Bioinformatics. 2009;25(15):1891–7. doi:10.1093/bioinformatics/btp311. http://bioinformatics.oxfordjournals.org/content/25/15/1891.full.pdf+html.
16. Leung HCM, Xiang Q, Yiu SM, Chin FYL. Predicting protein complexes from ppi data: a core-attachment approach. J Comput Biol. 2009;16(2):133–44.
17. Habibi M, Eslahchi C, Wong L. Protein complex prediction based on k-connected subgraphs in protein interaction network. BMC Syst Biol. 2010;4:129.
18. Jiang P, Singh M. Spici: a fast clustering algorithm for large biological networks. Bioinformatics. 2010;26(8):1105–11. doi:10.1093/bioinformatics/btq078. http://bioinformatics.oxfordjournals.org/content/26/8/1105.full.pdf+html.
19. Srihari S, Ning K, Leong H. Mcl-caw: a refinement of mcl for detecting yeast complexes from weighted ppi networks by incorporating core-attachment structure. BMC Bioinformatics. 2010;11(1):504. doi:10.1186/1471-2105-11-504.
20. Nepusz T, Yu H, Paccanaro A. Detecting overlapping protein complexes in protein protein interaction networks. Nat Methods. 2012;9:471–2.
21. Zaki N, Berengueres J, Efimov D. Detection of protein complexes using a protein ranking algorithm. Proteins Struct Function Bioinformatics. 2012;80(10):2459–68. doi:10.1002/prot.24130.
22. Ma X, Gao L. Discovering protein complexes in protein interaction networks via exploring the weak ties effect. BMC Syst Biol. 2012;6(S-1):6.
23. Becker E, Robisson B, Chapple CE, Guénoche A, Brun C. Multifunctional proteins revealed by overlapping clustering in protein interaction network. Bioinformatics. 2012;28(1):84–90.
24. Wong D, Li XL, Wu M, Zheng J, Ng SK. Plw: Probabilistic local walks for detecting protein complexes from protein interaction networks. BMC Genomics. 2013;14(Suppl 5):15. doi:10.1186/1471-2164-14-S5-S15.
25. Widita CK, Maruyama O. Ppsampler2: Predicting protein complexes more accurately and efficiently by sampling. BMC Syst Biol. 2013;7(Suppl 6):14.
26. Hanna E, Zaki N. Detecting protein complexes in protein interaction networks using a ranking algorithm with a refined merging procedure. BMC Bioinformatics. 2014;15(1):204. doi:10.1186/1471-2105-15-204.
27. Lin C, Cho YR, Hwang WC, Pei P, Zhang A. Clustering Methods in a Protein Protein Interaction Network. Hoboken, NJ: John Wiley & Sons, Inc.; 2007, pp. 319–55. doi:10.1002/9780470124642.ch16.
28. Li X, Wu M, Kwoh CK, Ng SK. Computational approaches for detecting protein complexes from protein interaction networks: a survey. BMC Genomics. 2010;11(Suppl 1):3. doi:10.1186/1471-2164-11-S1-S3.
29. Wang J, Li M, Deng Y, Pan Y. Recent advances in clustering methods for protein interaction networks. BMC Genomics. 2010;11(Suppl 3):10. doi:10.1186/1471-2164-11-S3-S10.
30. Chen B, Fan W, Liu J, Wu FX. Identifying protein complexes and functional modules: from static ppi networks to dynamic ppi networks. Brief Bioinform. 2014;15(2):177–94.
31. Srihari S, Yong CH, Patil A, Wong L. Methods for protein complex prediction and their contributions towards understanding the organisation, function and dynamics of complexes. FEBS Lett. 2015;589(19):2590–602.
32. Nguyen PV, Srihari S, Leong HW. Identifying conserved protein complexes between species by constructing interolog networks. BMC Bioinformatics. 2013;14(S-16):8.
33. Jung SH, Hyun B, Jang WH, Hur HY, Han DS. Protein complex prediction based on simultaneous protein interaction network. Bioinformatics. 2010;26(3):385–91. doi:10.1093/bioinformatics/btp668. http://bioinformatics.oxfordjournals.org/content/26/3/385.full.pdf+html.
34. Veres DV, Gyurkó DM, Thaler B, Szalay KZ, Fazekas D, Korcsmáros T, Csermely P. Comppi: a cellular compartment-specific database for protein–protein interaction network analysis. Nucleic Acids Res. 2015;43(Database issue (2015)):485–93.
35. Xu B, Lin H, Chen Y, Yang Z, Liu H. Protein complex identification by integrating protein-protein interaction evidence from multiple sources. PLoS ONE. 2013;8(12):83841. doi:10.1371/journal.pone.0083841.
36. Wodak SJ, Vlasblom J, Turinsky AL, Pu S. Protein protein interaction networks: the puzzling riches. Curr Opin Struct Biol. 2013;23(6):941–53. doi:10.1016/j.sbi.2013.08.002.
37. Park D, Singh R, Baym M, Liao CS, Berger B. Isobase: a database of functionally related proteins across ppi networks. Nucleic Acids Res. 2011;39(suppl 1):295–300. doi:10.1093/nar/gkq1234.
38. Franceschini A, Szklarczyk D, Frankild S, Kuhn M, Simonovic M, Roth A, Lin J, Minguez P, Bork P, von Mering C, Jensen LJ. String v9.1: protein-protein interaction networks, with increased coverage and integration. Nucleic Acids Res. 2013;41(D1):808–15. doi:10.1093/nar/gks1094. http://nar.oxfordjournals.org/content/41/D1/D808.full.pdf+html.

39. Zhang QC, Petrey D, Garzón JI, Deng L, Honig B. Preppi: a structure-informed database of protein–protein interactions. Nucleic Acids Res. 2012;1231:D828–33.
40. Li S, Rouphael N, Duraisingham S, et al. Molecular signatures of antibody responses derived from a systems biological study of 5 human vaccines. Nat Immunol. 2014;15(2):195–204. doi:10.1038/ni.2789.
41. Wu G, Feng X, Stein L. A human functional protein interaction network and its application to cancer data analysis. Genome Biol. 2010;11:53.
42. Tomita E, Tanaka A, Takahashi H. The worst-case time complexity for generating all maximal cliques and computational experiments. Theor Comput Sci. 2006;363(1):28–42.
43. Bomze IM, Budinich M, Pardalos PM, Pelillo M. The maximum clique problem. In: Handbook of Combinatorial Optimization. New York: Springer; 1999. p. 1–74.
44. Pellegrini M, Geraci F, Baglioni M. Detecting dense communities in large social and information networks with the core & peel algorithm. Technical Report arXiv:1210.3266, Cornell University Library ArXiv. 2012. http://arxiv.org/abs/1210.3266.
45. Kosub S. Local density. In: Brandes U, Erlebach T, editors. Network Analysis. Lecture Notes in Computer Science. New York: Springer; 2004. p. 112–42.
46. Garey MR, Johnson DS. Computers and Intractability: A Guide to the Theory of NP-Completeness. New York: W. H. Freeman & Co.; 1979.
47. Balasundaram B. Graph theoretic generalizations of clique: Optimization and extensions. PhD thesis, Texas A&M University. 2007.
48. Turán P. On an extremal problem in graph theory. Math Fiz Lapok. 1941;48:436–52.
49. Seidman SB. Network structure and minimum degree. Soc Netw. 1983;5(3):269–87. doi:10.1016/0378-8733(83)90028-X.
50. Garey MR, Johnson DS. Computers and Intractability: A Guide to the Theory of NP-Completeness. New York: W. H. Freeman; 1979.
51. Lund C, Yannakakis M. On the hardness of approximating minimization problems. J ACM. 1994;41:960–81. doi:10.1145/185675.306789.
52. Moon JW, Moser L. On cliques in graphs. Israel J Math. 1965;3(1):23–8.
53. Batagelj V, Zaversnik M. An O(m) algorithm for cores decomposition of networks. CoRR cs.DS/0310049. 2003.
54. Charikar M. Greedy approximation algorithms for finding dense components in a graph. In: Jansen K, Khuller S, editors. APPROX. Lecture Notes in Computer Science. New York: Springer; 2000. p. 84–95.
55. Halldórsson MM, Radhakrishnan J. Greed is good: approximating independent sets in sparse and bounded-degree graphs. In: Leighton FT, Goodrich MT, editors. STOC. New York: ACM; 1994. p. 439–48.
56. Stark C, Breitkreutz BJ, Reguly T, Boucher L, Breitkreutz A, Tyers M. Biogrid: a general repository for interaction datasets. Nucleic Acids Res. 2006;34(suppl 1):535–9. doi:10.1093/nar/gkj109.
57. Xenarios I, Rice DW, Salwínski L, Baron MK, Marcotte EM, Eisenberg D. Dip: the database of interacting proteins. Nucleic Acids Res. 2000;28(1): 289–91.
58. Pu S, Wong J, Turner B, Cho E, Wodak SJ. Up to date catalogues of yeast protein complexes. Nucleic Acids Res. 2009;37(3):825–31. doi:10.1093/nar/gkn1005. http://nar.oxfordjournals.org/content/37/3/825.full.pdf+html.
59. Ruepp A, Brauner B, Dunger-Kaltenbach I, Frishman G, Montrone C, Stransky M, Waegele B, Schmidt T, Doudieu ON, Stümpflen V, Mewes HW. Corum: the comprehensive resource of mammalian protein complexes. Nucleic Acids Res. 2008;36(Database-Issue):646–50.
60. Song J, Singh M. How and when should interactome-derived clusters be used to predict functional modules and protein function? Bioinformatics. 2009;25(23):3143–50. doi:10.1093/bioinformatics/btp551. http://bioinformatics.oxfordjournals.org/content/25/23/3143.full.pdf+html.
61. Peng W, Wang J, Zhao B, Wang L. Identification of protein complexes using weighted pagerank-nibble algorithm and core-attachment structure. Comput Biol Bioinform IEEE/ACM Trans. 2014. doi:10.1109/TCBB.2014.2343954.
62. Zhang B, Park BH, Karpinets T, Samatova NF. From pull-down data to protein interaction networks and complexes with biological relevance. Bioinformatics. 2008;24(7):979–86. doi:10.1093/bioinformatics/btn036. http://bioinformatics.oxfordjournals.org/content/24/7/979.full.pdf+html.
63. Storey JD, Tibshirani R. Statistical significance for genomewide studies. Proc Natl Acad Sci. 2003;100(16):9440–5. doi:10.1073/pnas.1530509100.
64. Rolland T, Taşan M, Charloteaux B, Pevzner SJ, Zhong Q, Sahni N, Yi S, Lemmens I, Fontanillo C, Mosca R, et al. A proteome-scale map of the human interactome network. Cell. 2014;159(5):1212–26.
65. Shen X, Yi L, Yi Y, Yang J, He T, Hu X. Dynamic identifying protein functional modules based on adaptive density modularity in protein-protein interaction networks. BMC Bioinformatics. 2015;16(Suppl 12):5.
66. Vandin F, Upfal E, Raphael BJ. Algorithms for detecting significantly mutated pathways in cancer. J Comput Biol. 2011;18(3):507–22.
67. Leiserson MD, Vandin F, Wu HT, Dobson JR, Eldridge JV, Thomas JL, Papoutsaki A, Kim Y, Niu B, McLellan M, et al. Pan-cancer network analysis identifies combinations of rare somatic mutations across pathways and protein complexes. Nat Genet. 2015;47(2):106–14.
68. Lecker SH, Goldberg AL, Mitch WE. Protein degradation by the ubiquitin proteasome pathway in normal and disease states. JASN. 2006;17: 1807–19.
69. Chymkowitch P, Le May N, Charneau P, Compe E, Egly JM. The phosphorylation of the androgen receptor by tfiih directs the ubiquitin/proteasome process. EMBO J. 2010;30(3):468–79. doi:10.1038/emboj.2010.337.
70. Dumay-Odelot H, Marck C, Durrieu-Gaillard S, Lefebvre O, Jourdain S, Prochazkova M, Pflieger A, Teichmann M. Identification, molecular cloning, and characterization of the sixth subunit of human transcription factor tfiiic. J Biol Chem. 2007;282:17179–89.
71. Dahlqvist J, Klar J, Tiwari N, Schuster J, T orm a H, Badhai J, Pujol R, van Steensel MAM, Brinkhuizen T, Gijezen L, Chaves A, Tadini G, Vahlquist A, Dahl N. A single-nucleotide deletion in the POMP 5′ UTR causes a transcriptional switch and altered epidermal proteasome distribution in KLICK genodermatosis. Am J Hum Genet. 2010;86(4): 596–603. doi:10.1016/j.ajhg.2010.02.018.
72. D'Arcy P, Brnjic S, Olofsson MH, Frykn as M, Lindsten K, De Cesare PMPerego, Sadeghi B, Hassan M, Larsson R, Linder S. Inhibition of proteasome deubiquitinating activity as a new cancer therapy. Nat Med. 2011;17(12):1636–40.
73. Kadoch C, Hargreaves DC, Hodges C, Elias L, Ho L, Ranish J, Crabtree GR. Proteomic and bioinformatic analysis of mammalian swi/snf complexes identifies extensive roles in human malignancy. Nat Genet. 2013;45(6):592–601.
74. Locker N, Chamond N, Sargueil B. A conserved structure within the hiv gag open reading frame that controls translation initiation directly recruits the 40s subunit and eif3. Nucleic Acids Res. 2011;39(6):2367–377.
75. Xie T, He Y, Korkeamaki H, Zhang Y, Imhoff R, Lohi O, Radhakrishnan I. Structure of the 30-kda sin3-associated protein (sap30) in complex with the mammalian sin3a corepressor and its role in nucleic acid binding. J Biol Chem. 2011;286(31):27814–7824. doi:10.1074/jbc.M111.252494.
76. Tsai YC, Greco TM, Cristea IM. Sirtuin 7 plays a role in ribosome biogenesis and protein synthesis. Mol Cellular Proteomics. 2014;13(1): 73–83. doi:10.1074/mcp.M113.031377.
77. Li Y, Masaki T, Yamane D, McGivern DR, Lemon SM. Competing and noncompeting activities of mir-122 and the 5′ exonuclease xrn1 in regulation of hepatitis c virus replication. Proc Natl Acad Sci. 2013;110(5): 1881–6. doi:10.1073/pnas.1213515110.
78. Srihari S, Leong HW. Employing functional interactions for characterisation and detection of sparse complexes from yeast ppi networks. Int J Bioinform Res Appl. 2012;8(3-4):286–304.
79. Gillis J, Ballouz S, Pavlidis P. Bias tradeoffs in the creation and analysis of protein–protein interaction networks. J Proteomics. 2014;100:44–54.