BIOINFORMATICS ORIGINAL PAPER

Vol. 27 no. 1 2011, pages 111–117
doi:10.1093/bioinformatics/btq652

Systems biology

Advance Access publication November 25, 2010

Identification of protein complexes from co-immunoprecipitation data

Guy Geva and Roded Sharan*

School of Computer Science, Tel Aviv University, Tel Aviv 69978, Israel

Associate Editor: Burkhard Rost

ABSTRACT

Motivation: Advanced technologies are producing large-scale protein–protein interaction data at an ever increasing pace. A fundamental challenge in analyzing these data is the inference of protein machineries. Previous methods for detecting protein complexes have been mainly based on analyzing binary protein–protein interaction data, ignoring the more involved co-complex relations obtained from co-immunoprecipitation experiments.

Results: Here, we devise a novel framework for protein complex detection from co-immunoprecipitation data. The framework aims at identifying sets of preys that significantly co-associate with the same set of baits. In application to an array of datasets from yeast, our method identifies thousands of protein complexes. Comparing these complexes to manually curated ones, we show that our method attains very high specificity and sensitivity levels (∼80%), outperforming current approaches for protein complex inference.

Availability: Supplementary information and the program are available at http://www.cs.tau.ac.il/∼roded/CODEC/main.html.

Contact: roded@post.tau.ac.il

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Received on December 17, 2008; revised on February 25, 2009; accepted on March 16, 2009

1 INTRODUCTION

Procedures such as yeast two hybrid and co-immunoprecipitation (CoIP) (Mann *et al.*, 2001) are routinely employed nowadays to detect new protein–protein interactions, producing large-scale protein interaction networks for various species. The networks provide a step stone for finding protein complexes—the fundamental units of macromolecular organization (Alberts, 1998).

The discovery of protein complexes based on yeast two hybrid data is a challenging task, since a protein complex may share common members with other complexes, and not all members of a certain protein complex directly interact with one another. CoIP data, however, can be used for finding complexes by itself since co-immunoprecipitation experiments directly test complex co-membership: a *bait* protein is tagged and a purification of its complex co-members (*prey* proteins) is made followed by mass spectrometry.

Surprisingly, most methods for detecting protein complexes are based on treating protein interaction data as binary, i.e. interactions are between pairs of proteins only. This is commonly done by translating non-binary CoIP associations, of a bait to the set of preys obtained by tagging it, into binary interactions using the spoke model (Bader and Hogue, 2002), where a purification is translated into a set of pairwise interactions between the bait and each of the precipitated preys.

One of the most commonly used algorithms for this task is the Molecular Complex Detection (MCODE) algorithm by Bader and Hogue (2003). MCODE detects densely connected components of the protein network. It is based on weighing vertices by the density of their local neighborhoods. Starting from a high weight vertex, a local expansion is done in a greedy fashion. Another common clustering algorithm is the Markov clustering algorithm (MCL) (Enright *et al.*, 2002). MCL simulates random walks on the protein interactions network. Random walks are performed iteratively; after sufficiently many iterations, the probability that a walk that starts in a dense area of the graph will end in the same dense area is high. In order to magnify this effect, MCL applies, after each walk, an inflation step that separates high probability connections from low probability ones. Eventually, the process converges and a cluster structure of the graph is formed. MCL was shown to outperform other clustering algorithms for protein complex detection (Brohee and van Helden, 2006). Recently, Rungsarityotin *et al.* (2007) presented a new clustering method based on Markov random fields (MRF). MRF applies a statistical model that assumes that the membership of each protein in a given cluster is only dependent on the membership status of its neighbors. Finally, Friedel *et al.* (2009) presented an unsupervised approach to find protein complexes that uses a bootstrapping mechanism to derive reliability scores for interactions between proteins. The resulting weighted network is then clustered using MCL.

The only unsupervised approach we are aware of that uses CoIP data directly is that of Scholtens *et al.* (2005). This approach is called Local Modeling and is probabilistic in nature. It relies on building a directed network of bait–prey relationships and searching for subnetworks in which all protein pairs that were tested for a bait–prey relation are connected. Such 'fully' connected subnetworks are shown to correspond well to protein complexes.

Supervised methods for identifying protein complexes have also been developed. Gavin *et al.* (2006) defined a 'socio-affinity' scoring system that measures the log ratio of the number of times two proteins are seen together in CoIP purifications, relative to what would be expected from their frequency in the dataset. These scores are used for clustering the proteins employing various

*To whom correspondence should be addressed.

clustering algorithms and parameters. Result sets that exhibit poor correspondence to manually curated complexes are discarded. Complex cores are identified as those stable parts of complexes that are not affected by the clustering algorithms/parameters. Collins *et al.* (2007) devised another scoring system for protein pairs, which combines the evidence in each purification for bait–prey and prey–prey relationships. They applied hierarchical clustering to these scores to produce putative complexes. Pu *et al.* (2007) used the (Collins *et al.*, 2007) scoring system together with the MCL algorithm to produce complex predictions. Another scoring system was used by Hart *et al.* (2007) in combination with the MCL algorithm to derive protein complexes. Another scoring scheme was developed by Zhang *et al.* (2008), who used a maximum clique finding algorithm to derive complex predictions.

In this article, we propose a novel method for inferring protein complexes from CoIP data, which we call CODEC (COmplex DEtection from Coimmunoprecipitation data). We represent the data using a bipartite graph, where one set of vertices corresponds to the prey proteins, and the other one corresponds to the bait proteins. Edges connect a bait to its associated preys. Ideally, protein complexes should be manifested as fully connected bipartite subgraphs of this graph, as also argued in Scholtens *et al.* (2005). In practice, experimental noise results in false positive and false negative associations in the CoIP data. In addition, for proteins that occur both as baits and preys in the data, we expect that if the bait (prey) instance is included in a complex, also its corresponding prey (bait) instance will be part of the complex. Thus, a complex is expected to appear as a dense bipartite subgraph such that every participating protein has both its bait and prey instances present.

To identify those dense balanced bipartite subgraphs of the bait–prey graph, we adapted the SAMBA biclustering algorithm (Tanay *et al.*, 2002). We applied CODEC to three datasets from three large-scale experiments in yeast (Gavin, 2002; Gavin *et al.*, 2006; Krogan *et al.*, 2006), identifying thousands of protein complexes. We evaluated CODEC and compared it to extant approaches by using a collection of manually curated complexes from the MIPS (Mewes, 2002) and GO (Cherry *et al.*, 1998) databases. First, we compared CODEC to the three clustering approaches: MCODE, MCL and MRF. We show that CODEC outperforms these approaches on two large-scale datasets, attaining higher values of specificity and sensitivity. We did not include a comparison to the bootstrap method of Friedel *et al.* (2009) as the software was not readily available. Second, we show that CODEC can be useful even when supervised approaches are applicable, comparing it to two representative supervised approaches: those of Gavin *et al.* (2006) and Collins *et al.* (2007). Remarkably, CODEC outperforms these approaches as well, even though they use curated information in the protein complex identification process. Finally, we show that CODEC compares favorably to the Local Modeling approach (Scholtens *et al.*, 2005), and at the same time it is much more scalable, allowing the analysis of much larger datasets.

2 METHODS

2.1 Data acquisition

We downloaded CoIP data for three datasets: (i) Gavin *et al.* (2006), which contains 1993 bait proteins, 2670 prey proteins and 19 277 bait–prey relationships; (ii) Krogan *et al.* (2006), which contains 2233 bait proteins, 5219 prey proteins [94 prey proteins were omitted from the raw data, since they were suspected as non-specific contaminants (Krogan *et al.*, 2006)] and 40 623 bait–prey relations; and (iii) Gavin (2002), which contains 455 bait proteins, 1364 prey proteins and 3413 bait–prey relations.

MIPS complexes were obtained from the MIPS database (Mewes, 2002) (February 2007 download). Only manually annotated complexes were used (category 550 was excluded). From the 243 manually annotated MIPS complexes, we considered only complexes at level 3 or lower. Higher level complexes were collapsed to level 3. Overall, the data contained 229 complexes. Gene ontology (GO) complexes were obtained from the Saccharomyces Genome Database (Cherry *et al.*, 1998) (March 2007 download). The GO dataset contained 193 complexes.

2.2 Graph construction and statistical data modeling

We represent the CoIP data using a bipartite graph $G=(U,V,E)$, where vertices on one side (U) represent purifications with specific baits, and vertices on the other side (V) represent the union of the set of preys detected in all the purifications and the set of baits. For convenience, we name the vertices according to the proteins they represent. Edges connect baits to their associated preys. In addition, every purification with a bait u is connected to u on the prey side. A candidate protein complex corresponds to a connected subgraph $H'=(U',V',E')$ of this graph, where $V'\subseteq V$ is the set of member proteins in the complex, and $U'\subseteq U$ is a set of purifications.

We use a likelihood ratio score to evaluate a candidate protein complex. The score measures the fit of a subgraph to a protein complex model versus the chance that the subgraph arises at random. The protein complex model assumes that each edge in the subgraph occurs with high probability p_c, independently of all other vertex pairs. This assumption ignores possible dependencies between bait–prey associations, but allows computing candidate complex scores in an efficient manner. The null model assumes that each edge (u,v) occurs with probability $p_{u,v}$, independently of all other vertex pairs, where $p_{u,v}$ is the probability of observing an edge between u and v in a random bipartite graph with the same vertex degrees as G. In practice, we use $p_c=0.9$ as recommended in Tanay *et al.* (2002). $p_{u,v}$ is approximated by $\frac{d(u)d(v)}{|E|}$ (Itzkovitz *et al.*, 2003), where $d(v)$ denotes the degree of a vertex v. Thus, the score of H' is:

$$L(H')=\sum_{(u,v)\in E'}\log\frac{p_c}{p_{u,v}}+\sum_{(u,v)\notin E'}\log\frac{1-p_c}{1-p_{u,v}}$$

By setting the weight of each edge (u,v) to be $\log\frac{p_c}{p_{u,v}}>0$ and the weight of each non-edge (u,v) to be $\log\frac{1-p_c}{1-p_{u,v}}<0$, we have that the score of a subgraph is the sum of weights of its vertex pairs.

There are two exceptions to setting the edge weights: (i) an edge of the form (v,v) is assigned zero weight. (ii) We call a vertex whose corresponding protein serves as a bait in some purification, but never detected as a prey, *artificial*. For such a vertex, we consider two weighting schemes. The first, which we call w_0, sets all weights involving artificial vertices to 0, based on the assumption that these cases represent proteins that cannot be detected as preys due to experimental limitations. The second scheme, which we call w_1, treats such vertices the same as all other vertices, resulting in all the weights involving artificial vertices being non-positive.

2.3 The algorithm

Our algorithm for protein complex identification employs a greedy search heuristic, which starts from high weight seeds and expands them using local search. We describe these phases below.

2.3.1 Seed definition. Recall that we seek heavy subgraphs of the bait–prey graph with the additional requirement that these subgraphs are consistent, namely that a bait instance of a protein is included if and only if the prey instance of the same protein is included. As seeds, we use complete bipartite subgraphs (bicliques) of the bait–prey graph, augmented by additional vertices so that the consistency requirement is satisfied. For a

prey $v \in V$, denote its corresponding bait (if such exists) by $m(v)$. Similarly, for a bait $u \in U$, denote its corresponding prey (which might be artificial) by $m(u)$. Then a prey subset $S \subseteq V$ with a set of common (bait) neighbors $N(S)$ induces the following consistent seed:

$$C(S) = S \cup N(S) \cup \{m(v) : v \in S \cup N(S)\}$$

2.3.2 Seed identification. We start by identifying a high weight seed around each protein. To find consistent seeds, we adapt the algorithm in Tanay *et al.* (2002). Basically, as shown in Tanay *et al.* (2002), the heaviest biclique in a bipartite graph can be identified by an iterative algorithm. At each iteration, the neighborhood of a vertex $u \in U$ is scanned, and each subset of its neighbors is credited by the weight from u to the vertices of this subset. After scanning all vertices in U, the subset that attained the highest weight induces the heaviest biclique. In our case, we have a further consistency requirement. Hence, we have to augment each of the possible seeds by appropriate vertices. To this end, we add a post-processing step to the algorithm above which updates the weight of every subset according to the consistent seed it induces.

For computational efficiency, we limit the size of the scanned subsets to 2–4. We only scan subsets that contain the prey vertex that corresponds to u. Each candidate seed is scored by its log-likelihood ratio. We retain the 500 000 highest scoring candidates and store them in a heap to prevent duplicates.

2.3.3 Greedy expansion. This phase iteratively applies modifications to the seed so as to expand it and increase its weight. Seeds are sorted by their log likelihood in a descending order. The greedy expansion is applied to the seeds by that order. At each iteration, all possible vertex additions to the seed and vertex deletions from the seed are considered, where baits are coupled to their corresponding preys to maintain consistency under these modifications. The modification that improves the score the most is accepted. This process continues until the score of the subgraph cannot be further improved. For efficiency reasons, this phase is applied only to seeds that were not contained in previous expanded subgraphs.

2.3.4 Filtering the results. We focus on clusters with at least three preys. We evaluate the significance of a cluster by comparing the score of its corresponding subgraph to those obtained on randomized instances. Specifically, we create random graphs with the same vertex degrees as G by using the Maslov–Sneppen procedure (Maslov and Sneppen, 2002). The procedure switches a pair of edges (u, v) and (u', v') with (u, v') and (u', v), provided that the latter did not exist in the first place. The switches are done $100m$ times, where m is the number of edges in the original graph (Milo *et al.*, 2003). Our algorithm is applied to these randomized instances to compute a null distribution of subgraph scores. We use this distribution to compute a P-value for each of the clusters and retain only clusters whose P-value is smaller than a threshold.

To avoid redundant solutions, we filter putative protein complexes with high similarity to one another. The similarity is measured based on the intersection of the prey sets of the compared clusters. Specifically, for two putative complexes V_1 and V_2 we measure their similarity as $|V_1 \cap V_2|/min\{|V_1|, |V_2|\}$. If the similarity exceeds a predefined threshold, then the subgraph with the higher P-value is discarded. We used 80% as the similarity filtering threshold [as in (Sharan *et al.*, 2005)]; a lower value of 50% yielded a similar performance (see Supplementary Table S2).

2.3.5 Implementation and running time We implemented CODEC using the microsoft .net framework 2.0 and the C# programming language. CODEC was applied to three datasets, as detailed above, on a Intel core 2 duo 1.86 GHz processor with 1 GB memory. The running time ranged from minutes to hours, depending on the size of the dataset. The running time of CODEC on the smallest (Gavin, 2002) dataset was 5 min; the application to the medium (Gavin *et al.*, 2006) dataset took 3 h; finally, the run on the largest (Krogan *et al.*, 2006) dataset lasted 30 h.

2.4 Quality assessment

We assess the quality of the produced complexes by measuring their specificity and sensitivity with respect to a set of gold standard (known) complexes. To this end, for each output cluster we find a known complex with which its intersection is the most significant according to a hypergeometric score. The hypergeometric score is compared with those obtained for 10 000 random sets of proteins of the same size, and an empirical P-value is derived. These P-values are further corrected for multiple hypothesis testing using the false discovery rate procedure (Benjamini and Hochberg, 1995). We say that a cluster is a significant match to a complex if it has a corrected P-value lower than 0.05.

Let C be the group of clusters from the examined result set, excluding clusters that do not overlap any of the true complexes. Let $C^* \subseteq C$ be the subset of clusters that significantly overlap a known complex. The *specificity* of the result set is defined as $|C^*|/|C|$. Let T be the set of true complexes, excluding complexes whose overlap with the examined dataset is less than 3 proteins and ensuring a maximum inter-complex overlap of 80%. Let $T^* \subseteq T$ be the subset of true complexes with a significant match by a cluster. The *sensitivity* of the result set is defined as $|T^*|/|T|$. The *F-measure* is a measure combining the specificity and sensitivity measures. It is defined as the harmonic average of these two measures:

$$2 * \frac{\text{specificity} * \text{sensitivity}}{\text{specificity} + \text{sensitivity}}$$

In addition, we also used the *Accuracy* measure suggested by Brohee and van Helden (2006). This measure also evaluates the quality of complex predictions against a gold standard set. The accuracy measure is the geometric mean of two other measures: *positive predictive value (PPV)* and *sensitivity*. PPV measures how well a given cluster predicts its best matching complex. Let $T_{i,j}$ be the size of the intersection between the i-th annotated complex and the j-th complex prediction. Denote

$$\text{PPV}_{i,j} = \frac{T_{i,j}}{\sum_{i=1}^{n} T_{i,j}} = \frac{T_{i,j}}{T_j}$$

where n is the number of annotated complexes, and T_j is the sum of the sizes of all of cluster j intersection sizes. The PPV of a single cluster j is defined as

$$\text{PPV}_j = \max_{i=1}^{n} \text{PPV}_{i,j}$$

The general PPV of the complex prediction set is defined by

$$\text{PPV} = \frac{\sum_{j=1}^{m} T_j \text{PPV}_j}{\sum_{j=1}^{m} T_j}$$

where m is the number of complex predictions. The sensitivity measure used by Brohee *et al.* (which is different from the one defined above) represents the coverage of a complex by its best-matching cluster. Denote

$$Sn_{i,j} = \frac{T_{i,j}}{N_i}$$

where N_i is the number of proteins in the annotated complex i. Complex-wise sensitivity is defined as

$$Sn_i = \max_{j=1}^{m} Sn_{i,j}$$

The sensitivity of a complex set is defined as

$$Sn = \frac{\sum_{i=1}^{n} N_i Sn_i}{\sum_{i=1}^{n} N_i}$$

The Accuracy measure can be influenced by small and insignificant intersections of a predicted complex and an annotated one. For example, if a predicted complex intersects only one annotated complex, and the size of the intersection is 1, the PPV of that predicted complex will be 1.0. Thus, we used a threshold to limit the effect of such small intersections, and evaluated the different solutions under varying thresholds ranging from 0 to 10. For each such threshold t, all intersections of size at most t were not included in the Accuracy computation.

2.5 Parameter tuning

The input for the MCL and MCODE clustering algorithms was the set of interactions resulting from connecting a bait protein to its preys [the spoke model (Bader and Hogue, 2002)] for each of the datasets. For setting the parameters of the algorithms, we used the values recommended by Brohee and van Helden (2006). Specifically, we used the inflation parameter 1.8 for MCL. For MCODE, we used the parameters depth = 100, node score percentage = 0, Haircut = TRUE, Fluff = FALSE and percentage for complex fluffing = 0.2. MRF was applied using the spoke model, using the parameters suggested by Rungsarityotin *et al.* (2007), i.e. $K = 698$ and $\psi = 3.5$.

We obtained the Local Modeling implementation from the bioconductor http://www.bioconductor.org. The parameters used to run Local Modeling are the default parameters mentioned in Scholtens *et al.* (2005).

When creating complex estimates from Collins *et al.* (2007), we used MCL with the same parameters as described above, and used the PE values as the input to the MCL algorithm.

3 RESULTS AND DISCUSSION

3.1 CODEC overview

CODEC is based on reformulating the protein complex identification problem as that of finding significantly dense subgraphs in a bipartite graph. We construct a bipartite graph whose vertices on one side represent prey proteins, and vertices on the other side represent bait proteins. Edges connect a bait protein to its associated preys. Ideally, a complex should appear as a fully connected bipartite subgraph (biclique) in this graph. In practice, due to experimental noise, a complex will appear as a dense bipartite subgraph. We note that further experimentation using methods such as cross-linking and sequential CoIP can improve the detection process, but is far more costly.

In addition, we impose a consistency requirement: some proteins occur in the data both as baits and as preys. For such proteins, we require that if a certain prey (bait) vertex is included in the subgraph, so must be the corresponding bait (prey). These definitions are exemplified in Figure 1. The example dataset contains 10 proteins marked as P1-P10 (Fig. 1a). Four purifications are made. The proteins used as baits are P3, P4, P5 and P7. There are two sets of preys that are supported by more than one bait: {P2,P3,P4,P5} and {P5, P6, P7, P8}. It can be hypothesized that these sets correspond to two protein complexes, shown in Figure 1b. In both cases, the consistency requirement is satisfied. The missing edge between P5 and P2 is a likely false negative, since both P3 and P4 interact with P2. There may be additional complexes in this toy example, but there is only weak evidence for their existence since they are detected as preys by a single bait protein.

We adapted the SAMBA algorithm (Tanay *et al.*, 2002) to find putative complexes, henceforth called *clusters*. As further detailed in the Methods, the algorithm relies on a scoring component and a search heuristic to identify high scoring subgraphs. The scoring of a subgraph is based on a likelihood ratio score, which measures the density of the subgraph versus the chance that its connections arise at random. We experimented with two scoring variants: a permissive one, w_0, and a stricter one, w_1 (see Section 2). In all the applications below, we report on the results of both variants. The search heuristic starts from small bicliques and expands them using greedy search. Unlike SAMBA, the search procedure also ensures that the consistency requirement is met by coupling together the prey and bait instances of a protein.

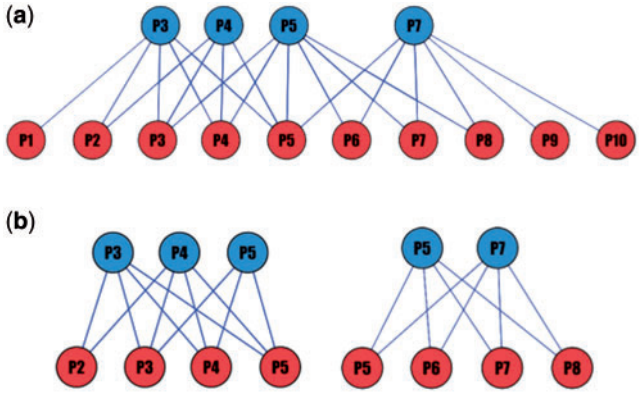

Fig. 1. An example data set. (**a**) An input bait–prey graph. Baits are colored in blue and preys are colored in red. (**b**) Two possible protein complexes and their corresponding subgraphs.

The significance of the identified clusters is evaluated by comparing their scores to those obtained on randomized instances, where the edges of the bipartite graph are shuffled while maintaining node degrees. We retain only significant clusters and further eliminate redundant clusters with high overlap among them.

3.2 Application and evaluation

As a first test of CODEC, we applied it to two recently published large-scale CoIP datasets in yeast. The first dataset from Gavin *et al.* (2006) contains 1993 bait proteins and 2670 prey proteins, and its edge density in the bipartite graph model is 0.006. The second dataset from Krogan *et al.* (2006) contains 2233 bait proteins and 5219 prey proteins, and its edge density in the bipartite graph model is 0.003. This dataset has a much lower bait to prey ratio than the former one and, thus, serves as a different test case for our method. CODEC was applied to the two original datasets; no proteins were filtered.

The application of CODEC to the first dataset using the w_0 weighting scheme yielded clusters with 12 baits and 22 preys on average The average edge density within an output cluster was very high (0.65). When using the stricter w_1 scheme, a similar number of clusters was obtained, but the clusters were much smaller (4.5 baits and 13.5 preys on average).

The application of CODEC to the second dataset using the w_0 weighing scheme produced clusters with 4 baits and 16 preys on average. The average interaction density within the output clusters was high (0.54). When using the w_1 scheme, the number of clusters dropped by 3-fold although their sizes remained similar to the w_0 case.

The size distributions of the obtained protein clusters in each of two applications are provided in Supplementary Table S1.

To assess the quality of our results, we measured their specificity and sensitivity with respect to a collection of manually curated complexes taken from the MIPS (Mewes, 2002) database (see Section 2). Specificity is defined as the fraction of clusters that significantly overlap a known complex; sensitivity is defined as the fraction of known complexes that significantly overlap an identified cluster. We computed receiver operating characteristic (ROC) curves for the two datasets, which plot the sensitivity and (1 − specificity) values over a range of *P*-value cutoffs for the output clusters (Figures 2 and 3). In each plot, we chose the point that maximizes

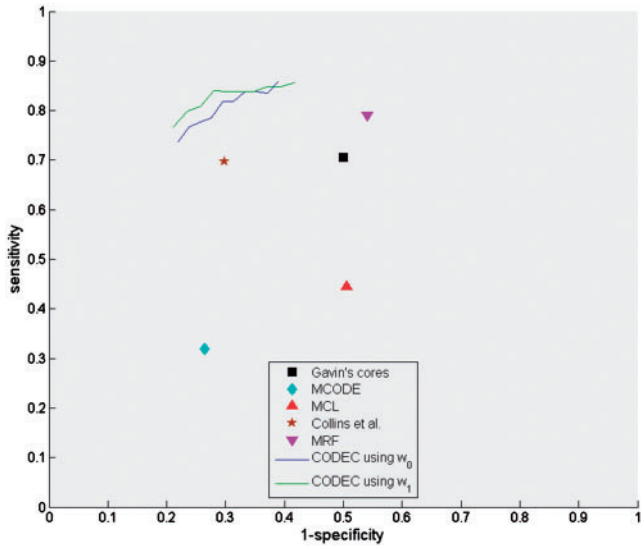

Fig. 2. A comparison of protein complex identification approaches on the data of Gavin *et al.* (2006). For each method shown is the sensitivity of the output solution as a function of one minus its specificity. For CODEC shown are two receiver operating characteristic (ROC) curves, corresponding to different weighting strategies (w_0 and w_1). The evaluation is based on a comparison to known protein complexes from the MIPS database (Mewes, 2002). The CODEC plots were smoothed using a cubic spline.

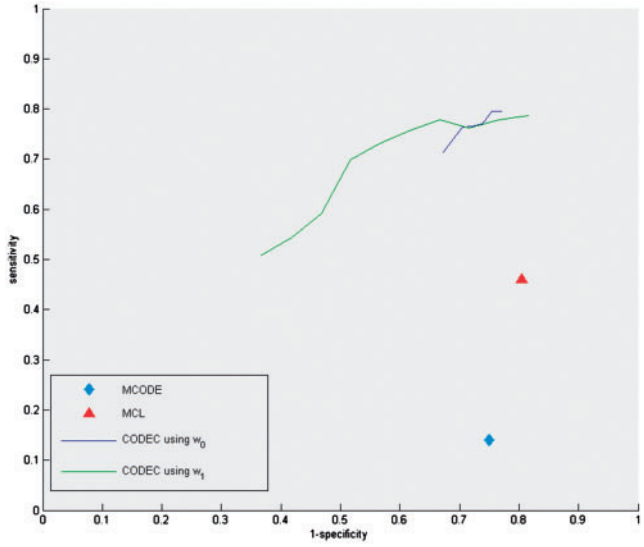

Fig. 3. A comparison of protein complex identification approaches on the data of Krogan *et al.* (2006). See legend of Figure 2 for details.

the sum of sensitivity and specificity (Coffin and Sukhatme, 1997) as the *P*-value cutoff for the output clusters. The results attained are summarized in Table 1.

We compared CODEC to three clustering algorithms: MCODE, MCL and MRF (Table 1 and Figures 2 and 3). On both datasets, CODEC outperformed MCODE and MCL, yielding significantly higher sensitivity values. The cluster set provided by Rungsarityotin *et al.* (2007) was computed by applying MRF using the spoke model to the Gavin *et al.* (2006) dataset (the MRF results with the matrix model were inferior and, hence, were not used in the comparison). CODEC and MRF achieved similar sensitivity scores, but at the same time CODEC attained significantly higher specificity.

Qualitatively similar results were obtained when evaluating the collections of protein complexes based on known complexes from the GO (Cherry *et al.*, 1998) database (see Supplementary Figures S1 and S2). When using an alternative evaluation measure—the Accuracy measure suggested by Brohee and van Helden (2006)—CODEC was again shown to outperform MCL and MCODE, while providing results that were only slightly better than those of MRF (see Supplementary Figures S3 and S4). Notably, all the tested methods perform worse on the data of Krogan *et al.* because of its low bait to prey ratio.

3.3 Comparison to extant CoIP-based approaches

The results above demonstrate the utility of using CoIP data for protein complex identification. Next, we compared CODEC to extant protein complex inference methods that use such data. As a first test, we compared CODEC to two other methods that use CoIP data for scoring pairs of putatively interacting proteins. The first (Gavin *et al.*, 2006) computes cores of complexes based on 'socio-affinity' scoring system that measures the log ratio of the number of times two proteins are seen together in CoIP purifications, relative to what would be expected from their frequency in the dataset. The second (Collins *et al.*, 2007) scores pairs of proteins using a purification enrichment (PE) score, which combines the evidence in each purification for bait–prey and prey–prey relationships. We used these PE scores as input to the MCL algorithm [as suggested in Brohee and van Helden (2006)]. Importantly, both methods use manually curated information (known protein complexes from MIPS) to tune their parameters.

We conducted the comparison on the the Gavin *et al.* (2006) dataset, for which we had the complex cores from Gavin *et al.* (2006). The results are summarized in Table 2 and depicted in Figure 2. Notably, even though the methods of Gavin *et al.* (2006) and Collins *et al.* (2007) use prior biological information in the inference process, CODEC outperforms both, attaining higher sensitivity and specificity values. The most pronounced difference is with respect to the specificity of Gavin's cores (78% versus 51%).

Our final comparison was to the Local Modeling method (Scholtens *et al.*, 2005). The available implementation of the method could not run on the datasets of Gavin *et al.* (2006) and Krogan *et al.* (2006) due to their relatively large size. Hence, we used a smaller data set as a test case (Gavin, 2002), containing 455 bait proteins and 1364 prey proteins. The protein complexes inferred by Local Modeling are partitioned into three categories: complexes that are supported by multiple baits (marked as MBME), complexes that are supported by a single bait (marked as SMBH) and complexes that contain two baits where only one of the baits identifies the other bait as its prey. We focused on the 272 MBME complexes, which represent the highest confidence predictions. As can be seen in Table 3 and Figure 4, CODEC outperforms local modeling, attaining higher specificity and sensitivity. When including in the Local Modeling solution also the SMBH complexes (336 in total) the sensitivity increased to 93%, at the price of a decrease in specificity

Table 1. Comparison to MCODE, MCL and MRF

	Gavin *et al.* (2006)				Krogan *et al.* (2006)			
	Number of Complexes	Specificity (%)	Sensitivity (%)	*F*-measure (%)	Number of Complexes	Specificity (%)	Sensitivity (%)	*F*-measure (%)
CODEC using w_0	1082	77.5	77	77	8348	30	**76.2**	43
CODEC using w_1	1005	78.5	**79**	**78.5**	2973	**46.5**	72	**56.5**
MCODE	73	73.5	32	44.5	130	25	14	18
MCL	411	49.5	44.5	47	818	19.5	46	27.5
MRF	698	**79.7**	46.7	59	–	–	–	–

A comparison of CODEC, MCODE, MCL and MRF on the datasets Gavin *et al.* (2006) and Krogan *et al.* (2006). The best result in each column appears in bold.

Table 2. Comparison to Collins *et al.* and Gavin *et al.* 2006

	Number of Complexes	Specificity (%)	Sensitivity (%)	*F*-measure (%)
CODEC using w_0	1082	77.5	77	77
CODEC using w_1	1005	**78.5**	**79**	**78.5**
Gavin *et al.* 2006	480	51.5	70.5	59.5
Collins *et al.*	258	70	69.5	69.5

A comparison of CODEC and the methods of Collins *et al.* and Gavin *et al.* on the dataset of Gavin *et al.* (2006). The best result in each column appears in bold.

Table 3. Comparison to Local Modeling

	Number of Complexes	Specificity (%)	Sensitivity (%)	*F*-measure (%)
CODEC using w_0	185	**80**	**85**	**82.5**
CODEC using w_1	180	79.5	81	80
Local Modeling	272	73	67	70

A comparison of CODEC to the Local Modeling approach on the dataset of Gavin (2002). The best result in each column appears in bold.

(to 69%). Overall, these results are comparable to those of CODEC, although providing a slightly worse *F*-measure (79% compared with CODEC's 82.5%).

4 CONCLUSION

We have provided a novel algorithm for identifying protein complexes from co-immunoprecipitation data, which is based on reformulating the problem as that of finding heavy subgraphs in a bipartite graph. We have shown that our approach, which uses non-binary co-complex information, is superior to clustering methods that dissect binary protein–protein interaction data. Our algorithm was also shown to outperform existing approaches for inferring protein complexes from CoIP data. All complex predictions made by CODEC can be found at http://www.cs.tau.ac.il/~roded/CODEC/main.html. An interesting open challenge is to combine yeast two-hybrid data into the inference process. Such a combined approach is expected to become increasingly important as protein–protein interaction databases continue to grow in size and species coverage.

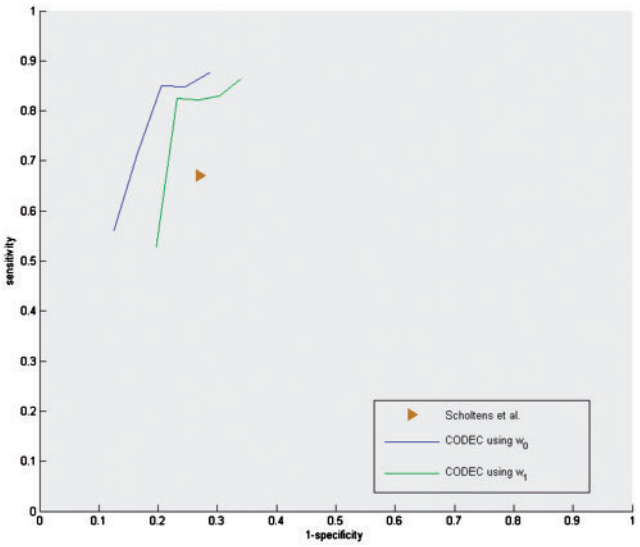

Fig. 4. A comparison of protein complex identification approaches on the data of Gavin (2002). See legend of Figure 2 for details.

Funding: Israel Science Foundation (grant no. 385/06) to R.S.

Conflict of Interest: none declared.

REFERENCES

Alberts,B. (1998) The cell as a collection of protein machines: preparing the next generation of molecular biologists. *Cell*, **92**, 291–294.

Bader,G.D. and Hogue,C.W. (2002) Analyzing yeast protein-protein interaction data obtained from different sources. *Nat. Biotechnol.*, **20**, 991–997.

Bader,G.D. and Hogue,C.W. (2003) An automated method for finding molecular complexes in large protein interaction networks. *BMC Bioinformatics*, **4**, 2.

Benjamini,Y. and Hochberg,Y. (1995) Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. Roy. Stat. Soc. B Met.*, **57**, 289–300.

Brohee,S. and van Helden,J. (2006) Evaluation of clustering algorithms for protein-protein interaction networks. *BMC Bioinformatics*, **7**, 488.

Cherry,J. *et al.* (1998) SGD: Saccharomyces Genome Database. *Nucl. Acids Res.*, **26**, 73–79.

Coffin,M. and Sukhatme,S. (1997) Receiver operating characteristic studies and measurement error. *Biometrics*, **53**, 823–837.

Collins,S.R. *et al.* (2007) Towards a comprehensive atlas of the physical interactome of Saccharomyces cerevisiae. *Mol. Cell Proteomics*, **6**, 439–450.

Enright,A. *et al.* (2002) An efficient algorithm for large-scale detection of protein families. *Nucleic Acids Res.*, **30**, 1575–1584.

Friedel,C.C. *et al.* (2009) Bootstrapping the interactome: Unsupervised identification of protein complexes in yeast. *J. Comput. Biol.*, **16**, 971–987.

Gavin,A.C. (2002) Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature*, **415**, 141–147.

Gavin,A.-C. *et al.* (2006) Proteome survey reveals modularity of the yeast cell machinery. *Nature*, **440**, 631–636.

Hart,G.T. *et al.* (2007) A high-accuracy consensus map of yeast protein complexes reveals modular nature of gene essentiality. *BMC Bioinformatics*, **8**, 236.

Itzkovitz,S. *et al.* (2003) Subgraphs in random networks. *Phys. Rev. E*, **68**, 026127.

Krogan,N.J. *et al.* (2006) Global landscape of protein complexes in the yeast saccharomyces cerevisiae. *Nature*, **440**, 637–643.

Mann,M. *et al.* (2001) Analysis of proteins and proteomes by mass spectrometry. *Annu. Rev. Biochem.*, **70**, 437–473.

Maslov,S. and Sneppen,K. (2002) Specificity and stability in topology of protein networks. *Science*, **296**, 910.

Mewes,H.W. (2002) Mips: a database for genomes and protein sequences. *Nucleic Acids Res.*, **30**, 31–34.

Milo,R. *et al.* (2003) On the uniform generation of random graphs with prescribed degree sequences. *ArXiv Condensed Matter, arXiv:cond-mat/0312028v2*.

Pu,S. *et al.* (2007) Identifying functional modules in the physical interactome of saccharomyces cerevisiae. *Proteomics*, **7**, 944–960.

Rungsarityotin,W. *et al.* (2007) Identifying protein complexes directly from high-throughput tap data with markov random fields. *BMC Bioinformatics*, **8**, 482.

Scholtens,D. *et al.* (2005) Local modeling of global interactome networks. *Bioinformatics*, **21**, 3548–3557.

Sharan,R. *et al.* (2005) Conserved patterns of protein interaction in multiple species. *Proc. Natl Acad. Sci. USA*, **102**, 1974–1979.

Tanay,A. *et al.* (2002) Discovering statistically significant biclusters in gene expression data. *Bioinformatics*, **18** (Suppl. 1), 136–144.

Zhang,B. *et al.* (2008) From pull-down data to protein interaction networks and complexes with biological relevance. *Bioinformatics*, **24**, 979–986.