

Towards Detecting Protein Complexes from Protein Interaction Data

Pengjun Pei¹ and Aidong Zhang¹ *

Department of Computer Science and Engineering
State University of New York at Buffalo
Buffalo NY 14260, USA

Abstract. High-throughput methods for detecting protein-protein interactions (PPI) have given researchers an initial global picture of protein interactions on a genomic scale. These interactions connect proteins into a large protein interaction network (PIN). However, both the size of the data sets and the noise in the data pose big challenges in effectively analyzing the data. In this paper, we investigate the problem of protein complex detection, i.e., finding biologically meaningful subsets of proteins, from the noisy protein interaction data. We identify the difficulties and propose a “seed-refine” approach, including a novel subgraph quality measure, an appropriate heuristics for finding good seeds and a novel subgraph refinement method. Our method considers the properties of protein complexes and the noisy interaction data. Experiments show the effectiveness of our method.

1 Introduction

Proteins must interact with other molecular units to execute their function. Discovering proteins that interact in a cell is key to elucidate its functional networks. Recent advances in biotechnology have made it possible to detect protein interactions on a global scale [10, 17, 7, 9]. We can construct a protein interaction network (PIN) [7] from existing protein-protein interaction data by connecting each pair of vertices (proteins) involved in an interaction.

Proteins are likely to form closely-coupled protein complexes as functional units to participate in a certain biological process. Detecting the protein complexes from protein interaction network will help find the building modules of the protein network. These complexes can be roughly considered as dense subgraphs of the protein interaction network. However, protein complexes are likely to overlap and the interaction data are very noisy. Therefore, intelligent methods are in great demand to effectively detect protein complexes.

This paper identifies the difficulties of this problem and proposes some favorable properties of the methods for this purpose. Then we propose a ‘seed-refine’

* This research was partially supported by National Science Foundation Grants DBI-0234895, IIS-0308001 and National Institutes of Health Grant 1 P20 GM067650-01A1.

approach, including a novel subgraph quality measure, a heuristic to find good seeds, and a novel method to control subgraph overlapping. Experiments show the effectiveness of our method. Finally, we conclude the paper and propose some future work.

2 Challenges in protein complex detection

Though the problem of detecting protein complexes from interaction data shares some commonality with clustering problem, there are some additional difficulties:

- Protein complexes may overlap with each other. Therefore, the traditional paradigm for clustering of putting each protein into one single cluster [14] does not suit our problem well. Instead, we would prefer finding ‘dense’ subgraphs.
- Protein complexes generally correspond to small but dense subgraphs. Divisive hierarchical clustering approaches like [4] are more useful in finding large protein clusters representing biological processes. For small dense subgraphs, we would prefer using extensive local search and optimization.
- Protein interaction data are very noisy. How to define the quality of a subgraph in the presence of noisy edges is a non-trivial task. Normal quality measurement of subgraphs either considers only the worst connected vertices or the averaged overall density.

To sum up, we need a seed-refine approach for protein complex detection. Specifically, we need to generate some promising seed subgraphs, followed by refining these seed subgraphs based on some quality measure. We can stop the refining process when either the quality is smaller than a predefined value, or preferably, the quality achieves a local maximum. The latter is much more preferred because it will not require a predefined threshold.

3 A seed-refine approach for finding protein complexes

Throughout the paper, we use an unweighted, undirected graph $G = (V, E)$ to represent the protein interaction network where V represents the set of vertices (proteins) and E represents the set of edges (interactions). An *induced subgraph* is a subset of the vertices of the graph together with all the edges of the graph between the vertices of this subset. As we only consider an induced subgraph of the original graph, we abbreviate the term and simply call it a *subgraph*. We denote the set of neighbors of a vertex v in graph G as $N(v) = \{u | (u, v) \in E\}$.

3.1 Subgraph quality definition

As observed in [13, 8], topology features can provide some insight on the biological significance of the interactions. For proteins that interact with a lot of

other proteins, the biological significance of these interactions might be questionable. Therefore, the degree of a vertex should be taken into consideration when evaluating the strength of an edge. Correspondingly, the quality of a subgraph $G' = (V', E')$ is related to not only $|V'|$ and number of inside links $|E'|$ but also outside links $|E_{out}| = \{(u, v) | u \in V', v \notin V'\}$. Previous quality definitions like density ($Density = 2 * |E'| / (|V'| * (|V'| - 1))$) [15], k-core [1, 2] and cliques disregard outside links.

Meanwhile, we would prefer a subgraph in which each vertex contributes similarly to the quality of the subgraph so that every vertex is likely to be an authentic part. Comparatively, the density considers only the average quality. Cliques and 'k-core' define the subgraph quality by the worst-connected vertex. These definitions are too stringent and therefore will miss a lot of potentially biologically meaningful subgraphs.

We first define the quality of a vertex in a subgraph, denoted as $Q(v, G')$: For a vertex $v \in V'$, the number of edges within the subgraph $G' = (V', E')$ is $|N(v) \cap V'|$. Under the null hypothesis that the set of neighbors of v is chosen randomly from the vertices of the graph, i.e., V , the probability of observing at least $|N(v) \cap V'|$ neighbors within the subgraph G' can be expressed as:

$$PV_{v, G'} = \sum_{i=|N(v) \cap V'|}^{\min(|N(v)|, |V'|)} \binom{|N(v)|}{i} \times \binom{|V'| - |N(v)|}{|V'| - i} / \binom{|V'|}{|V'|}.$$

We define $Q(v, G')$ as the minus log of this probability, i.e., $Q(v, G') = -\log(PV_{v, G'})$.

Then we seek to combine $Q(v, G')$ values for all $v \in V'$. Since we prefer a subgraph in which each vertex contributes similarly to the quality of the subgraph, we treat the logs of $Q(v, G')$ for all $v \in V'$ as a random sample from a Normally distributed population with mean μ .¹ We estimate the sample mean, denoted as \bar{x} and variance, denoted as s^2 : $\bar{x} = \frac{\sum_{v \in V'} \log(Q(v, G'))}{|V'|}$, $s^2 = \frac{\sum_{v \in V'} (\log Q(v, G') - \bar{x})^2}{|V'| - 1}$. Then the sampling distribution of $\frac{\bar{x} - \mu}{\sqrt{s^2 / |V'|}}$ follows Student's t distribution with $|V'| - 1$ degrees of freedom [3]. We use the lower boundary of the 95% confidence interval of the population mean as our subgraph quality measure:

$$Q(G') = e^{\bar{x} - t_{|V'| - 1} * \sqrt{s^2 / |V'|}},$$

where $t_{|V'| - 1}$ is the cut-off value of the t distribution for 95% confidence interval with $|V'| - 1$ degrees of freedom. This quality measure gives a boundary of the underlying population mean, and therefore, is a statistically meaningful combination of the quality of each vertex.

3.2 The seed-refine algorithm

Our algorithm iteratively finds an initial **seed graph** centered on an edge (u, v) , denoted as $G_{(u, v)}^{(0)}$ and refines it until no quality improvement can be achieved.

¹ The log transformation of the quality values is used to stabilize variance and thus to make the sample satisfy the Normal distribution requirement.

We call this optimized subgraph a **refined subgraph**, denoted as $G_{(u,v)}$. The set of refined subgraphs, denoted as GS , represents our predicted complexes. We use $Visited$ to represent the set of edges that have been covered by refined subgraphs. The “seed-refine” algorithm is illustrated in Algorithm 1 in Figure 1.

After we get GS , a simple postprocessing can be applied to filter out those subgraphs with quality less than a threshold.

3.3 Finding seed subgraphs

We define two layer seeds: we use an edge (u, v) not in previously refined subgraphs as the **seeding edge** and find corresponding **seeding vertices**, denoted as $SV_{(u,v)}$: $SV_{(u,v)} = \{w | (u, w) \in E \setminus Visited, (v, w) \in E \setminus Visited\}$.

The seed subgraph is the subgraph induced by $SV_{(u,v)} \cup \{u, v\}$. This definition guarantees that those edges in $Visited$ can not be used as part of the seed graph. Since each of the seeding vertices is connected to both vertices of the seeding edge, the seed subgraph can be regarded as **centered on** the seeding edge (u, v) . Therefore, we denote it as $G_{(u,v)}^{(0)}$. Given all candidate seed subgraphs with at least 3 vertices, we choose the most promising one: the one with the largest number of vertices. If there is a tie, we choose the one with highest quality, i.e., we define the function $isMorePromising(G1, G2) = true$ iff

$$(|V(G1)| > |V(G2)|) \text{ or } (|V(G1)| = |V(G2)| \text{ and } (Q(G1) > Q(G2))).$$

The subroutine is described in Algorithm 2 in Figure 1.

3.4 Refining subgraphs

Given a seed subgraph, the subroutine *refineSubGraph* tries all possible actions of adding one vertex to or removing one vertex from the subgraph and takes the action that achieves highest quality improvement. In this process, we require that (i) the subgraph contains the seeding edge (u, v) , and (ii) the subgraph remains connected. This process is repeated until no quality improvement action can be found. The pseudocode is listed in Algorithm 3 in Figure 1.

3.5 Analysis of the algorithm

Since the subgraph quality improves monotonously in the *refineSubGraph* subroutine, the refinement recursion can end. Also, since we increase the set $Visited$ after finding one refined subgraph, the algorithm will end after visiting all edges.

For computational complexity, notice that we are finding dense but *small* subgraphs and we require that the subgraph always contains the seeding edge. Therefore the refined subgraph will be still close to the seeding edge, suggesting that the *refineSubGraph* recursion will not take too much time. Also, considering the sparsity of the PIN, computational time is not a serious issue here.

Notice that our objective is to predict a ‘reasonable’ number of protein complexes without excessive overlapping. The control of subgraph overlapping is

Algorithm1 *SRA*: Seed-Refine Algorithm for Protein Complex Detection

Input: $G = (V, E)$: protein interaction network
Output: GS : the set of predicted protein complexes

1. Initialization: $GS \leftarrow \phi, Visited \leftarrow \phi$
2. $G_{(u,v)}^{(0)} \leftarrow findSeedSubgraph(G, Visited)$
3. **while** $G_{(u,v)}^{(0)} \neq empty$ **do**
4. $G_{(u,v)} \leftarrow refineSubGraph(G_{(u,v)}^{(0)})$
5. $GS \leftarrow GS \cup \{G_{(u,v)}\}$
6. $Visited \leftarrow Visited \cup E(G_{(u,v)})$
7. $G_{(u,v)}^{(0)} \leftarrow findSeedSubgraph(G, Visited)$
8. **end while**
9. **return** GS

Algorithm2 *findSeedSubgraph*: Find a Seed Subgraph

Input: $G = (V, E)$: protein interaction network, $Visited$
Output: $G_{(u,v)}^{(0)}$: a seed subgraph centered on (u, v)

1. Initialization: $G_{(u,v)}^{(0)} \leftarrow empty, Candidate \leftarrow E \setminus Visited$
2. **for all** edge $(i, j) \in Candidate$ **do**
3. Construct the seed subgraph centered on (i, j) : $G_{(i,j)}^{(0)}$
4. **if** $|V(G_{(i,j)}^{(0)})| \geq 3$ **and** $isMorePromising(G_{(i,j)}^{(0)}, G_{(u,v)}^{(0)})$ **then**
5. $G_{(u,v)}^{(0)} \leftarrow G_{(i,j)}^{(0)}$
6. **end if**
7. **end for**
8. **return** $G_{(u,v)}^{(0)}$

Algorithm3 *refineSubGraph*: Refine a subgraph

Input: $G_{(u,v)}^{(i)}$: a subgraph centered on (u, v)
Output: $G_{(u,v)}$: a refined subgraph centered on (u, v)

1. Generate graphs $\{G'_{(u,v)}\}$ by adding a vertex to/deleting a vertex from $G_{(u,v)}^{(i)}$
2. **for all** graph $G'_{(u,v)}$ **do**
3. $G_{(u,v)}^{(i+1)} \leftarrow \underset{G'_{(u,v)}}{\operatorname{argmax}}(Q(G'_{(u,v)}))$
4. **end for**
5. **if** $Q(G_{(u,v)}^{(i+1)}) > Q(G_{(u,v)}^{(i)})$ **then**
6. **return** $refineSubGraph(G_{(u,v)}^{(i+1)})$
7. **else**
8. **return** $G_{(u,v)}^{(i)}$
9. **end if**

Fig. 1. A seed-refine algorithm for protein complex detection.

achieved by the two layer seeds: the seeding edge is used as the center of the seed subgraph. It is fixed in each refinement iteration to prevent the subgraph from being attracted towards another dense area far from the original seed. The seeding vertices give the preliminary shape of the seed subgraph for further refinement. After finding one refined subgraph, we prevent the edges in the subgraph from existing in later seed subgraphs. Therefore, the next seed subgraph tends to be a bit far from already discovered refined subgraphs. Also notice that we do not prohibit the inclusion of edges in previously refined subgraphs in the refinement process. This gives the possibility of overlapping subgraphs. Our choice of *isMorePromising* function is more likely to select larger subgraphs as seeds and thus less likely to branch out into dense regions of the graph that have already been discovered. To conclude, we design an algorithm that allows outputting overlapping subgraphs but methodologically makes it possible only when there is strong evidence to do so.

4 Experiments

Due to space limit, we only report the performance of our algorithm for two data sets: *PreHTMS* includes all yeast interactions except high-throughput mass spectrometry studies. *HTP* data set includes purely large scale studies.

To assess our predictions, we use the curated protein complexes in MIPS [11] (including 267 complexes with at least two proteins) and manually curated Gavin complexes [7] (including 221 complexes) as the ground truth. Similar to [2], for a predicted complex $G' = (V', E')$, we find the best-match complex in the ground truth complex set, denoted as $GT' = (GTV', GTE')$ and use $MatchRatio = \frac{|V' \cap GTV'|}{|V'|} * \frac{|V' \cap GTV'|}{|GTV'|}$ to evaluate the match. We consider a predicted complex *matches* a ground truth complex if $MatchRatio > 0.2$. For a total number of N predicted complexes and M ground truth complexes, suppose CN predicted complexes match CM ground truth complexes, we define $precision = \frac{CN}{N}$ and $recall = \frac{CM}{M}$.¹ We report the precision and recall of our method in Figure 2. For reference, we also list the result from [2].

Figure 2(a) shows that our method outperforms MCODE method. Also notice that our quality threshold is used in the postprocessing step and therefore, we only need to run the main algorithm once and can choose different thresholds on the unfiltered results afterwards. Though MCODE scores each vertex only once, its complex-finding subroutine still needs to run for different parameter choices.

Comparing our results in Figure 2 (a) and (b), we notice that the performance of the PreHTMS data set is higher than that of HTP data set when assessed by MIPS protein complexes. This is because PreHTMS includes more

¹ Our definition of the precision equals the specificity in [2]. However, their sensitivity is defined as $\frac{CN}{CN+M-CM}$. In our experiments, the difference has little effect on the final performance comparison or parameter selection.

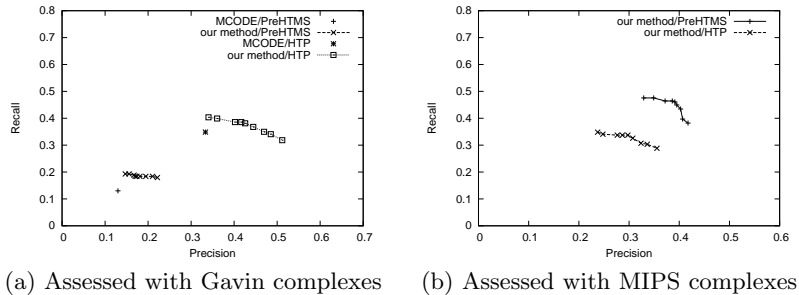


Fig. 2. Algorithm Performance. PreHTMS (9049 interactions among 4325 proteins) includes several large scale studies [10, 17, 16, 12, 5, 6] and the small scale studies in DIP [18]. HTP includes 12243 interactions among 4554 proteins from [10, 17, 5, 6, 9, 7, 16]. We use the spoke model [1] to extract binary interactions from the raw purifications in [9, 7]. For our approach, we choose the postprocessing quality threshold over the range from 6 to 10 in 0.5 increments and report the results. In (a), we use Gavin protein complexes for the assessment. We list reported results from [2] for MCODE using its most optimized parameter settings. In (b), we use MIPS protein complexes for the assessment. Since we use a later version of MIPS complexes, the performance of our method and MCODE is not directly comparable, thus we omit the MCODE result.

reliable interactions. However, HTP data set has higher performance when assessed by Gavin protein complexes. This is because the HTP data itself includes the interaction data inferred from the Gavin raw purifications.

Table 1 gives the details of a correctly predicted protein complex using PreHTMS data set. The $PV_{v,G'}$ scores for these five vertices in the subgraph, as listed in the table, are all very high and similar. The final quality of the subgraph is 13.7. This subgraph corresponds to the ‘STE5-MAPK complex’ in MIPS.

Table 1. STE5-MAPK complex correctly predicted by our method.

Protein	Neighbors in the Subgraph	Total Degree	$PV_{v,G'}$ Score
FUS3	STE5, STE7, STE11	12	14.9
KSS1	STE5, STE7, STE11	10	15.5
STE5	FUS3, KSS1, STE7, STE11	11	21.2
STE7	FUS3, KSS1, STE5	5	18.0
STE11	FUS3, KSS1, STE5	13	14.6

5 Conclusion and future work

In this paper, we have investigated the problem of finding protein complexes from protein interaction network and proposed a novel method. Experiments have shown the effectiveness of our method.

Similar to [2], our method can be used in a *directed mode* to find the complex that a specified protein is part of. This directed mode enables researchers to focus on the proteins of interest. Our method for generating seeds and subgraph refinement may also alleviate the problem of seed subgraph branching out into denser regions in the graph in [2]. We plan to investigate this usage of the algorithm in the future.

References

1. G. D. Bader and C. W. Hogue. Analyzing yeast protein-protein interaction data obtained from different sources. *Nat Biotechnol*, 20:991–997, 2002.
2. G. D. Bader and C. W. Hogue. An automated method for finding molecular complexes in large protein interaction networks. *BMC Bioinformatics*, 4:2, 2003.
3. M. Bland. *An Introduction to Medical Statistics*. Oxford University Press, USA, 2000.
4. C. Ding et al. A unified representation of multiprotein complex data for modeling interaction networks. *Proteins*, 57:99–108, 2004.
5. B. L. Drees et al. A protein interaction map for cell polarity development. *J Cell Biol*, 154:549–571, 2001.
6. M. Fromont-Racine et al. Genome-wide protein interaction screens reveal functional networks involving sm-like proteins. *Yeast*, 17:95–110, 2000.
7. A. C. Gavin et al. Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature*, 415:141–147, 2002.
8. D. S. Goldberg and F. P. Roth. Assessing experimentally derived interactions in a small world. *Proc. Natl. Acad. Sci. USA*, 100:4372–4376, 2003.
9. Y. Ho et al. Systematic identification of protein complexes in *saccharomyces cerevisiae* by mass spectrometry. *Nature*, 415:180–183, 2002.
10. T. Ito et al. Toward a protein-protein interaction map of the budding yeast: A comprehensive system to examine two-hybrid interactions in all possible combinations between the yeast proteins. *Proc. Natl. Acad. Sci. USA*, 93(3):1143–7, 2000.
11. H. W. Mewes et al. Mips: a database for genomes and protein sequences. *Nucleic Acids Res*, 30:31–34, 2002.
12. J. R. Newman, E. Wolf, and P. S. Kim. A computationally directed screen identifying interacting coiled coils from *saccharomyces cerevisiae*. *Proc Natl Acad Sci U S A*, 97:13203–13208, 2000.
13. R. Saito et al. Interaction generality, a measurement to assess the reliability of a protein-protein interaction. *Nucleic Acids Res*, 30:1163–1168, 2002.
14. M. P. Samanta and S. Liang. Predicting protein functions from redundancies in large-scale protein interaction networks. *Proc Natl Acad Sci U S A*, 100:12579–12583, 2003.
15. V. Spirin and L. A. Mirny. Protein complexes and functional modules in molecular networks. *Proc Natl Acad Sci U S A*, 100:12123–12128, 2003.
16. A. H. Tong et al. A combined experimental and computational strategy to define protein interaction networks for peptide recognition modules. *Science*, 295:321–324, 2002.
17. P. Uetz et al. A comprehensive analysis of protein-protein interactions in *saccharomyces cerevisiae*. *Nature*, 403:623–627, 2000.
18. I. Xenarios et al. Dip, the database of interacting proteins: a research tool for studying cellular networks of protein interactions. *Nucleic Acids Res*, 30:303–305, 2002.