Interaction Graph Mining for Protein Complexes Using Local Clique Merging

Xiao-Li Li[1]
xlli@i2r.a-star.edu.sg

Soon-Heng Tan[1,2]
soonheng@i2r.a-star.edu.sg

Chuan-Sheng Foo[1,3]
chuansheng.foo@stanford.edu

See-Kiong Ng[1]
skng@i2r.a-star.edu.sg

[1] Institute for Infocomm Research, 21 Heng Mui Keng Terrace, Singapore 119613
[2] School of Computing, National University of Singapore, Singapore 119260
[3] Computer Science Department, Stanford University, Stanford CA 94305-9025 USA

Abstract

While recent technological advances have made available large datasets of experimentally-detected pairwise protein-protein interactions, there is still a lack of experimentally-determined protein complex data. To make up for this lack of protein complex data, we explore the mining of existing protein interaction graphs for protein complexes. This paper proposes a novel graph mining algorithm to detect the dense neighborhoods (highly connected regions) in an interaction graph which may correspond to protein complexes. Our algorithm first locates local cliques for each graph vertex (protein) and then merge the detected local cliques according to their affinity to form maximal dense regions. We present experimental results with yeast protein interaction data to demonstrate the effectiveness of our proposed method. Compared with other existing techniques, our predicted complexes can match or overlap significantly better with the known protein complexes in the MIPS benchmark database. Novel protein complexes were also predicted to help biologists in their search for new protein complexes.

Keywords: protein complex, protein interaction graph, local clique, merging

1 Introduction

Biological processes in the cell are mostly undertaken by complex interactions between protein molecules. Many involved bio-molecular entities called protein complexes—these are molecular aggregations of proteins assembled from multiple (typically non-pairwise) stable protein-protein interactions. Protein complexes can vary widely in size, and they play crucial roles in many cellular processes. Prominent examples include the ribosomes for protein biosynthesis, the proteasomes for breaking down proteins, and the nuclear pore complexes for regulating proteins passing through the nuclear membrane. With individual proteins as molecular units, protein complexes correspond to higher-order functional units in biological processes. In fact, many proteins are functional only after they are assembled into a complex. Elucidating such protein complexes is therefore an important research focus in cell and molecular biology.

However, despite recent advances in protein interaction detection technologies, only a very small subset of the many possible protein complexes has been experimentally determined [14]. On the other hand, many high throughput experimental techniques (e.g. yeast-two-hybrid) have enabled the detection of pairwise protein-protein interactions *en masse*. Large sets of interaction data are now readily available in public databases for data mining and knowledge discovery such as mining potential domain-domain interactions [12], interacting motifs [16], as well as protein complexes [1, 10].

Typically, protein interaction data are modeled as an undirected graph where the vertices represent unique proteins and edges denote interactions between two proteins. Previous work by Tong and Drees [17] has revealed that protein complexes generally correspond to dense regions (densesubgraphs) or even cliques (fully connected subgraphs) in proteininteraction graphs. In fact, Spirin and Mirny [15] attempted to detect protein complexes and functional modules by finding cliques in protein interaction networks. However, their approach is limited by the use of cliques which can be problematic given that current available interaction data is incomplete. Other than clique-mining approaches, several clustering algorithms have also been proposed to identify the dense regions in a given graph by partitioning graphs into disjoint clusters [6, 19]. However, as they require that each vertex (protein) belongs to one specific cluster, these algorithms are not suitable for finding complexes in interaction graphs because a protein may be involved in multiple complexes.

Recently, Bader and Hogue [1] proposed the MCODE algorithm that utilizes connectivity values in protein interaction graphs to mine for protein complexes. The algorithm first computes the vertex weighting (vertex weighting step) from its neighbor density and then traverses outward from a seed protein with a high weighting value (complex prediction step) to recursively include neighboring vertices whose weights are above a given threshold. However, since the highly weighted vertices may not be highly connected to each other, the algorithm does not guarantee that the discovered regions are dense. In fact, in the post preprocessing step of the MCODE algorithm, there was a need to filter for 2-core (graph of minimum degree 2), which means that the vertices in the discovered regions were not always dense or even at least 2-degrees.

More recently, King *et al.* [10] proposed to use a restricted neighborhoods search clustering algorithm (RNSC) to predict protein complexes by partitioning the protein-protein interaction network using a cost function. However, like many clustering algorithms, their results depended heavily on the quality of the initial clustering which is random or user-defined. Also, relatively fewer complexes were predicted by this algorithm.

In this paper, we propose an effective and efficient algorithm to predict protein complexes from protein-protein interaction graphs using local clique merging. First, we locate local cliques in an interaction graph using our proposed polynomial time algorithm. Our algorithm then merges the local cliques into bigger dense graphs for protein complex identification. Note that we predict complexes based on dense graphs rather than cliques. As there are no requirements for the dense graphs to be fully connected, our algorithm is less sensitive to incomplete protein interaction data than conventional clique detection methods. We evaluated our method using yeast protein interaction data and found that the F-measures of the protein complexes predicted by our approach were significantly higher than those detected by other computational techniques.

2 Method

In this paper, we propose to mine an interaction graph G_{ppi} for protein complexes using a Local Clique Merging Algorithm (LCMA) to identify maximal dense subgraphs in G_{ppi} for protein complexes. Let us define the concept of a dense graph based on the clustering coefficient of the graph. Suppose we have a subgraph $G' = (V', E')$, where V' is a subset of vertices (proteins) and E' a subset of edges (interactions) from G_{ppi}. Theoretically, the maximum number of edges for the undirected graph is $|V'| * (|V'| - 1)/2$. We define the density (cluster coefficient) of a graph as follows.

Definition 1 *The density of the graph $G' = (V', E')$ is defined as:*

$$cc(G') = \frac{|E'|}{|V'| * |V' - 1|/2} = \frac{2 * |E'|}{|V'| * |V' - 1|}. \tag{1}$$

Basically, $0 \leq cc(G') \leq 1$. If $cc(G') = 1$, then G' is the fully connected graph or a clique, which has the maximum number of edges.

Our LCMA algorithm consists of two basic steps to find maximal dense subgraphs in G_{ppi}. The first step computes the local cliques for all the vertices in G_{ppi}. This is based on the observation that a maximal dense region covering vertices $\{v_1, \ldots v_k\}$ in G_{ppi} must necessarily contain the local cliques (if any) of the vertices from $\{v_1, \ldots v_k\}$. Then, in the second step, we merge these local cliques to form maximal dense graphs.

2.1 Mining for Local Cliques

The first step of our LCMA algorithm is to find the local cliques in the graph G_{ppi}. For each vertex from graph G_{ppi}, we first get its initial local neighborhood graph - namely, v_i, all its neighbors and the edges between the neighbors in graph G_{ppi}.

Definition 2 *Let a graph $G = (V, E)$. For each vertex $v_i \in V$, its local neighborhood graph $G_{v_i} = (V_{v_i}, E_{v_i})$, where $V_{v_i} = \{v_i\} \cup \{v | v \in V, (v, v_i) \in E\}$, $E_{v_i} = \{(v_j, v_k) | (v_j, v_k) \in E, v_j, v_k \in V_{v_i}\}$.*

Our proposed LCMA algorithm then graphically uncovers the local cliques for each vertex by iteratively removing loosely connected neighboring vertices. This is an iterative process that stops when the density (cluster coefficient) of the local neighborhoods cannot be increased further. Figure 1 shows an example of mining the local cliques for a vertex 1. In the graph, the vertex 1's neighbors are: 6, 7, 2, 4, 5, and 3. Note we sort all the neighbors according to their degrees in the vertex's local neighborhood graph. Then we iteratively remove a neighbor vertex if it increases the density—here, the neighbors 6, 7, 2 are removed correspondingly. This results in the final local clique shown in the circled area, which is a clique that consists of vertex set D = {1, 3, 4, 5} and density $cc(D) = 1$ ($|V| = 4,$ *and* $|E| = 6$).

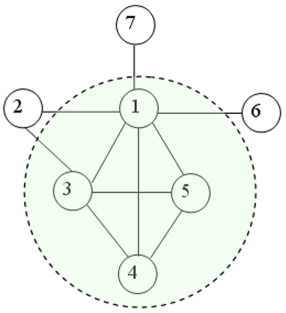

Figure 1: An example of mining a local clique from a local neighborhood graph.

The details of our LCMA algorithm to mine local cliques (LC) from local neighborhood graphs are in Algorithm 1. In the algorithm, steps 7 to 20 comprise the main loop. The algorithm first computes the densities λ for $AL(v) \bigcup \{v\}$. Then, it iteratively removes the loosely connected vertices until the density of the local neighborhood graph does not increase. In particular, step 11 finds the vertex with the least degree in v's neighborhood and then checks if it should be removed. If the density of the graph is increased after its removal (step 12), only then it is removed in step 13, followed by steps 14 and 15 to update the vertex degrees in the local neighborhood graph and assign the new density to the refined graph. Otherwise, there are no loosely connected vertices to remove and we stop the removing process. Finally, steps 21 to 25 remove all cliques consisting of only 2 vertices. This filter ensures that the our discovered graph is dense (fully connected) and contains at least three proteins.

The time complexity of this local clique finding step is $\mathcal{O}(m * log(m))$, where m is the number of edges in graph G_{ppi}. The main time cost is mainly for building the sorted adjacency lists (steps 3 to 6). However, the subsequent iterative removal of the loosely connected vertices can be performed in linear time after we have sorted the adjacency lists.

Next, we show that each local dense neighborhood in the output set LC is a clique.

Theorem 1 *Every graph in LC is a clique.*

Proof: For each vertex $v \in G_{ppi}$, our algorithm first finds the vertex v' with the least degree in v's local neighborhood graph $G_v = (V_v, E_v)$, i.e.

$$v' = \arg\min_{v_i} deg(v_i), v_i \in V_v. \tag{2}$$

The inner loop (steps 10 to 19) of the algorithm terminates when the removal of the vertex v' does not result in an increase in the density of the resultant graph. This condition is stated in equation 3.

$$\frac{2|E_v|}{|V_v|(|V_v| - 1)} \geq \frac{2(|E_v| - deg(v'))}{(|V_v| - 1)(|V_v| - 2|)}. \tag{3}$$

Algorithm 1 LCMA algorithm(step 1):Mining local cliques LC from local neighborhood graphs

```
BEGIN
Set $LC = \emptyset$;
for each vertex $v \in G_{ppi}$ do
    Construct $v$'s Adjacency List $AL(v)$;
    Sort $AL(v)$ according to their degree in $v$'s neighborhood;
end for
for each vertex $v \in G_{ppi}$ do
    Compute the density (clustering coefficient) $\lambda = cc(AL(v) \cup \{v\})$;
    $Stop = False$;
    repeat
        Find the vertex $v'$ with minimum degree in $AL(v)$, $v' = \arg\min_{v_i} d(v_i)$;
        if $(cc(AL(v) \cup \{v\} - \{v'\}) > \lambda)$ then
            remove $v'$ from $AL(v)$;
            update the degree for vertices that connected with $v'$ in $AL(v)$;
            $\lambda = cc(AL(v) \cup \{v\} - \{v'\})$;
        else
            $Stop = True$;
        end if
    until $(Stop = True)$
end for
for each graph $g \in LC$ do
    if $|g| \leq 2$ then
        $LC = LC - \{g\}$;
    end if
end for
END
```

Solving equation 3 for $|E_v|$ yields

$$|E_v| \leq \frac{deg(v')|V_v|}{2}. \tag{4}$$

Since each edge connects two vertices,

$$|E_v| = \frac{\sum_{i=1}^{|V_v|} deg(v_i)}{2}. \tag{5}$$

From (2), $deg(v') \leq deg(v_i)$, $\forall v_i \in V_v$, and $deg(v) = |V_v| - 1$. Therefore,

$$|E_v| \geq \frac{(|V_v| - 1)deg(v') + (|V_v| - 1)}{2}, \tag{6}$$

$$|E_v| \geq \frac{(|V_v| - 1)(deg(v') + 1)}{2}. \tag{7}$$

Combining (4) and (7)

$$\frac{deg(v')|V_v|}{2} \geq \frac{(|V_v| - 1)(deg(v') + 1)}{2}, \tag{8}$$

$$deg(v') \geq |V_v| - 1. \tag{9}$$

Equation 9 shows that the algorithm terminates only when the degree of v' is $|V_v| - 1$, the maximum possible. This implies that all other vertices will also have degree $|V_v| - 1$. In other words, the resulting graph is a fully connected one, namely a clique. □

2.2 Merging for Maximal Dense Neighborhoods

In an interaction graph with potentially incomplete interaction data, it is more likely for a large protein complex to be presented as a maximal dense neighborhood consisting of various local cliques than as a single large clique within the interaction graph. To detect such dense graphs which can match the larger complexes better, the LCMA algorithm performs a merging step after the local cliques have been identified, adopting the heuristic of merging overlapping neighborhoods with comparable sizes.

Definition 3 *Neighborhood Affinity. Given two neighborhoods (subgraphs) A and B, we define the Neighborhood Affinity NA between them as*

$$NA(A,B) = \frac{|A \cap B|^2}{|A| * |B|}. \tag{10}$$

Equation 10 quantifies the degree of similarity between neighborhoods. If two neighborhoods have larger intersection sets and similar sizes, then they are more similar and have bigger neighborhood affinity. Note that if one neighborhood's size, e.g. $|B|$, is much bigger than $|A|$, then $NA(A,B)$ will be small since $|A \cap B|/|A| < 1$ and $|A \cap B| \ll |B|$.

Given a set LC of local cliques, our LCMA algorithm tries to merge the local cliques that have a similarity greater than a threshold ω. The merging process is performed iteratively to update LC. The details of the local clique merging step of our LCMA algorithm are presented in algorithm 2. The input of the algorithm are the local cliques in LC, while the output—the predicted complexes—will be stored in the set C. Steps 4-24 comprise the main loop of the algorithm. In steps 6-14, we compute, for each local clique n_i, its neighborhood similarity with all other local cliques in LC. Neighborhoods with similarities greater than ω are then merged with n_i. Both merged complexes(step 10) and the local cliques that did not undergo merging (steps 15-17) are added into C. In steps 18-23, we use the average density (AD) in each iteration as a stop criteria — if the new average density of the current iteration does not decrease much (i.e. less than 95% of last average density) and there are new complexes produced from merging, then we update the average density value and continue the merging process. Otherwise the algorithm will stop.

The time complexity of this local clique merging step is $\mathcal{O}(l * r^2 * v)$, where l is the number of iterations(a constant and can thus be ignored), r the size of LC (which is smaller than the vertex number $|V|$), and v the average number of proteins in the local cliques. Usually v has small values, e,g, the average protein number of complexes in MIPS is around 6.38 [11].

3 Experimental Results

In this section, we evaluate our proposed LCMA algorithm on an actual interaction graph generated from the experimental protein-protein interaction data of yeast. We have chosen to use yeast interaction data to infer protein complexes as it is currently the organism with the most comprehensive experimental datasets available publicly. In order to show the effectiveness of our algorithm, we will compare our results with the results from existing algorithms using the same datasets.

3.1 Dataset for Protein Interaction Graph

We use the same dataset collected by Bader and Hogue[1] for their MCODE algorithm to construct our protein interaction graph for mining complexes. The dataset was assembled from all machine-readable resources: Uetz [18], Ito [8], Drees [3], Fromont-Racine [4], Ho [7], Gavin [5], Tong [17], Mewes(MIPS) [11], Costanzo(YPD) [2]. In total, it consists of 15,143 experimentally determined protein-protein interactions among 4,825 yeast proteins.

Algorithm 2 LCMA algorithm(step 2):Mining maximal dense neighborhoods by merging local cliques

```
BEGIN
$Stop = False$;
Average Density AD= $\sum_{i=1}^{|LC|} cc(n_i)/|LC|,\ n_i \in LC$;
repeat
  $C = \emptyset$;
  for each $n_i \in LC$ do
    $S = \{n_i\}$;
    for each $n_j \in LC, j \neq i$ do
      if $NA(n_j, n_i) > \omega$ then
        $S = S \cup \{n_j\}$;
      end if
    end for
    $C = C \cup S$;
  end for
  for each $n_i \in LC$ not used in a merge do
    $C = C \cup \{n_i\}$;
  end for
  if $\sum_{i=1}^{C} cc(n_i)/|C|\ > 0.95 * AD, n_i \in C$ then
    $AD = \sum_{i=1}^{|C|} cc(n_i)/|C|,\ n_i \in C$;
    $LC = C$;
  else
    $Stop = True$;
  end if
until (Stop = True) or (no new complexes are produced from merging)
END
```

3.2 Protein Complex Gold Standard and Evaluation Metric

The benchmark that we use to evaluate our method against is a dataset of known yeast protein complexes retrieved from the MIPS: Comprehensive Yeast Genome Database (`ftp://ftpmips.gsf.de/yeast/`). Entries in this dataset have been curated from the biomedical literature. Note that while it is currently one of the most comprehensive public datasets of yeast complexes available, it is still by no means a complete dataset — there are still many yeast complexes that remained to be detected. After filtering the predicted protein complexes from the dataset, we obtained a final set of 215 yeast complexes as our gold standard for evaluation. In it, the biggest protein complex, cytoplasmic ribosomes, contains 81 proteins while the average protein number of complexes is 6.38.

In addition to using the same MIPS dataset as the gold standard to evaluate the results, we also assess our proposed algorithm using the same evaluation metric adopted in the MCODE paper [1], where the definition of NA (definition 3) was used to determine matching between predicted complexes in P and MIPS complexes. For a predicted complex $p \in P$ and a complex $m \in$ MIPS, we consider the two complexes to be matching if $NA(p, m) \geq 0.2$ which is the same threshold used in MCODE.

The set of true positives (TP) is defined as $TP = \{p|NA(p, m) \geq 0.2, p \in P, m \in \text{MIPS}\ \}$, namely the predicted complexes with affinity NA (with MIPS complexes) at least 0.2. The set of false positives (FP) is defined as $FP = P - TP$, namely the predicted complexes that are not in TP. The set of false negatives (FN) is defined as $FN = \{m|\forall p(NA(p, m) < 0.2), p \in P, m \in \text{MIPS}\}$, namely the known MIPS complexes not matched by predicted complexes. Then the recall and precision can be defined as: $Recall = |TP|/(|TP| + |FN|)$ and $Precision = |TP|/(|TP| + |FP|)$.

In this paper, we use the F-measure to evaluate the performance of different techniques. F-measure

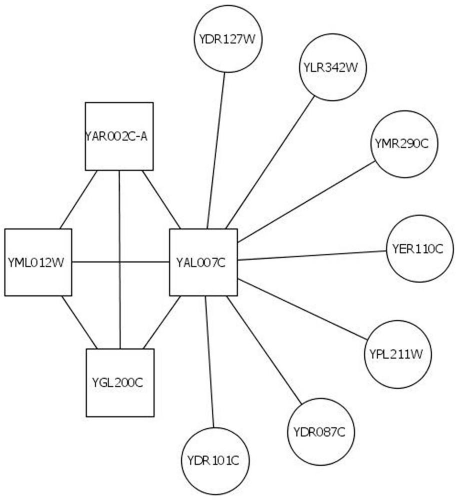

Figure 2: An example of mining a local clique.

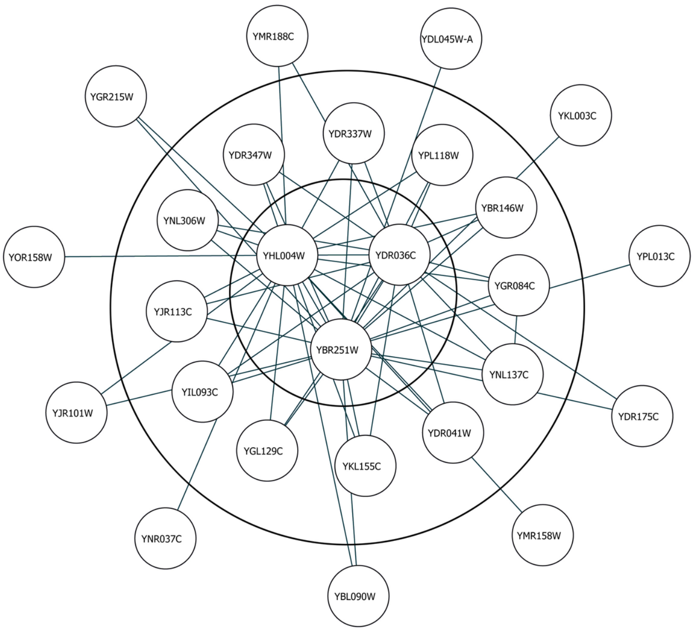

Figure 3: Merging multiple local cliques to match MIPS complex.

takes into account of both precision and recall and is defined as $F - measure = 2 * Precision * Recall/(Precision + Recall)$. Note that because our reference set MIPS is incomplete, some predicted complexes which probably are true complexes will be regarded as false positives (FP) if they do not match well with the current MIPS complexes. As such, the F-measure of the algorithms should not be taken at their absolute values but only as comparative measures.

3.3 Experimental Results

Our LCMA algorithm aims to achieve high recall by trying to discover most of the dense regions in the protein interaction graph by using the local cliques for each vertex as seeds. Our algorithm also filters away the loosely connected vertexes which could correspond to experimental false positives [13], to enhance its robustness (hence precision) against the high error rates in current protein interaction data.

Figure 2 shows an example to illustrate the biological significance of mining the local cliques for a vertex YAL007C (a yeast protein) in our data. The vertices shown as rectangles form local cliques for vertex YAL007C. It is a clique containing 4 proteins that are fully connected with each other. All the other neighbors (shown as circle vertices) connect with YAL007C with one edge, which are possibly false positive interactions [13]. In fact, on further checking on the functional information of the proteins in the neighborhoods in MIPS (`http://mips.gsf.de/desc/yeast`), we found that all the proteins in the cliques have the same function label of "vesicular transport (Golgi network, etc.)"(MIPS code "20.09.07"), while all the other proteins have different functions (not listed). This example shows that the local cliques can lead to functionally consistent protein groups (hence more likely to be members of a complex for a specific biological function), and our algorithm was effective in removing the functionally inconsistent nodes from the neighborhoods.

The second step of our LCMA algorithm aims to further improve the precision by merging overlapping local cliques into maximal dense neighborhoods. Figure 3 shows an example in which an actual

MIPS complex cannot be matched by any individual local cliques but can only be discovered from the underlying interaction graph using the merging step to uncover the corresponding maximal dense neighborhood. The complex predicted by our method LCMA (shown in the figure within the larger circle) actually corresponds to a large MIPS complex consisting of 31 proteins. Let us look at how our algorithm arrives at this complex in closer details. First, in the algorithm's local clique mining step, 11 local cliques were found: other than YNL137C and YGR084C, each protein shown between the two circles in the figure forms a clique of size 4 with the three proteins in the inner circle, while YNL137C and YGR084C form a clique of size 5 with the inner proteins. The LCMA algorithm's second step then merges all the proteins within bigger circle into a bigger dense neighborhood of 15 proteins, of which 14 are parts of the MIPS complex. The remaining protein (YDR036C) in the bigger dense neighborhood have unknown function and could be an unknown component of the MIPS complex — its true membership, however, will have to be ascertained by the biologists experimentally. Obviously, we still have not discovered the entire 31-protein complex even with local clique merging to improve coverage. However, this is could be due to the incompleteness of the protein interaction dataset used to construct the underlying interaction graph.

Figure 4 shows the comparative results based on F-measures of different techniques, namely, the MCODE algorithm [1], and our LCMA algorithm with different values for the parameter ω for merging local cliques. For comparison, we also include the full clique results obtained by enumerating all non-redundant cliques(starting with an empty set, vertices were recursively added to the set as long as the clique condition was satisfied). Note that unlike our scalable LCMA algorithm, full clique-finding algorithms are not scalable to bigger protein interaction graphs.

In terms of F-measure, our LCMA algorithm achieves the best results. Compared with MCODE algorithm, LCMA algorithm with $\omega = 0$ was able to achieve 15.99% and 6.6% higher F-measure than MCODE algorithm and full clique results respectively. Our method can achieve 44.1% higher recall and 4.4% higher precision than MCODE. In addition, both LCMA and full clique results can achieve higher recall, but our LCMA algorithm can achieve 7.4% higher precision than clique results.

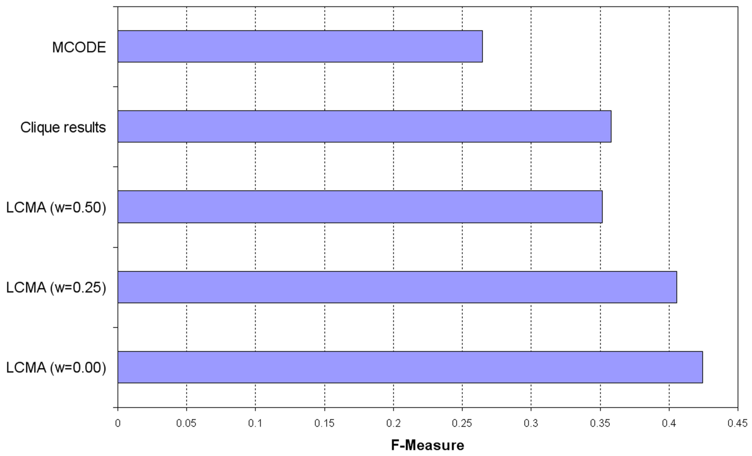

Figure 4: Comparison of different techniques to match MIPS complexes.

We also tested our algorithm with different ω values, namely $\omega = 0.5$, $\omega = 0.25$, and $\omega = 0$. We observed that generally the recall does not change a lot (71.3%-73.9%) while the precision increases with decreasing ω values (23.1%-30.2%). When $\omega = 0$, our LCMA algorithm predicted 873 complexes, of which 264 matched 108 MIPS benchmark complexes ($|TP|$=264, $|FP|$=609, $|FN|$=107). The average *Neighborhood Affinity* between predicted complexes that matched benchmark complexes and their best matched benchmark complex is 0.40.

Note that when ω is larger than 0.60, local cliques can seldom be merged because 0.60 or above is a strict threshold. For example, consider two local cliques A and B, both consisting of 12 proteins. Even if they have 9 overlapping proteins, the $NA(A, B)$ is only $9 * 9/(12 * 12) = 0.56$ which is below the threshold to qualify for merging. On the contrary, if ω is a smaller value, it is easier for two neighborhood local cliques to merge. When ω is set to 0, all pairs of the local cliques will be merged as long as there is a single common protein.

A more recent system [10] used a so-called "restricted neighborhoods search clustering" algorithm

(RNSC) to predict protein complexes from protein interaction graph. RNSC predicted only 45 complexes which matched 30 MIPS complexes. In comparison, our LCMA algorithm was able to identify many more of the MIPS complexes from the protein interaction graph.

Our LCMA algorithm also predicted complexes that do not match current MIPS complexes. Since the current MIPS complex set is largely incomplete, these unmatched complexes could potentially be real complexes for biologists to explore further. In fact, we tried to match the predicted complexes (LCMA with $\omega = 0$) that do not match MIPS complexes with complexes in the BIND database—out of 609 predicted complexes that did not match with MIPS complexes, 335 matched 339 BIND complexes (`http://www.blueprint.org`). As such, many of our current false positives are probably true positives, and the actual precision of our method should be higher than reported here.

4 Conclusions

Protein interaction graph mining can be used to identify graphical subcomponents for predicting protein complexes. In this paper, we have described an efficient and effective technique to mine protein complexes from protein interaction graphs by local clique merging. Evaluations of our proposed algorithm showed that it has the following advantages over other existing methods:

- *Higher F-measure (recall and precision).* By taking a bottom-up approach that considers the local cliques for each vertex in the interaction graph and then merging the overlapping local cliques for maximal dense neighborhoods, our proposed algorithm can discover much more protein complexes in protein interaction graph than other algorithms. Our algorithm also guarantees that the discovered complexes are dense graphs in the underlying interaction graph. This enhances the accuracy because highly connected graphs are more likely correspond to the complexes.

- *More efficient runtime.* Our algorithm is also more efficient than current approaches. The time complexity of our overall LCMA algorithm (two steps) is $\mathcal{O}(r^2 v)$ ($r < |V|$ is the number of local cliques and v usually has a small value) while the complexity of MCODE algorithm is $\mathcal{O}(nmh^3)$ [1] (n, m are the number of vertices and edges respectively, h is the vertex size of the average vertex neighborhood) and finding maximal cliques from graph is a NP-hard problem [9]. The running time of our LCMA algorithm is around 1 minute (when $\omega = 0$) on an Intel Pentium M 1.86GHz with 512MB RAM.

As mentioned earlier, identifying protein complexes is critical for biological knowledge discovery since many important biological processes in the cell are carried out through the formation of protein complexes. However, there is currently a wide data gap between protein complexes and protein-protein interactions—technologies for detecting pairwise protein-protein interactions *en masse* have already become routine in the laboratories for generating large datasets of protein interaction data, while the technologies for detecting protein complexes still remained highly painstaking and costly. In this work, we have shown how we can exploit the abundant protein interaction data to bridge the data gap for protein complexes—as our predicted complexes were shown to match or overlap well with the known protein complexes in MIPS benchmark database, the unmatched complexes could potentially be real complexes. Our method can thus be used to identify novel protein complexes from protein interaction graphs to help biologists in their continuing search for new protein complexes.

References

[1] Bader, G. D. and Hogue, C. W. V., An automated method for finding molecular complexes in large protein interaction networks, *BMC Bioinformatics*, 4(1):2, 2003.

[2] Costanzo, M. C., Crawford, M. E., Hirschman, J. E., Kranz, J. E., *et al.*, YPD, PombePD and WormPD: Model organism volumes of the BioKnowledge library, an integrated resource for protein information, *Nucleic Acids Res.*, 29(1):75–79, 2001.

[3] Drees, B. L., Sundin, B., Brazeau, E., Caviston, J. P., *et al.*, A protein interaction map for cell polarity development, *J. Cell Biol.*, 154(3):549–571, 2001.

[4] Fromont-Racine, M., Mayes, A. E., Brunet-Simon, A., Rain, J. C., *et al.*, Genome-wide protein interaction screens reveal functional networks involving Sm-like proteins, *Yeast*, 17(2):95–110, 2000.

[5] Gavin, A.-C., Bosche, M., Krause, R., Grandi, P., *et al.*, Functional organization of the yeast proteome by systematic analysis of protein complexes, *Nature*, 415(6868):141–147, 2002.

[6] Hartuv, E. and Shamir, R., A clustering algorithm based on graph connectivity, *Information Processing Letters*, 76:175–181, 2000.

[7] Ho, Y., Gruhler, A., Heilbut, A., Bader, G. D., *et al.*, Systematic identification of protein complexes in Saccharomyces cerevisiae by mass spectrometry, *Nature*, 415(6868):180–183, 2002.

[8] Ito, T., Chiba, T., Ozawa, R., Yoshida, M., *et al.*, A comprehensive two-hybrid analysis to explore the yeast protein interactome, *Proc. Natl. Acad. Sci. USA*, 98(8):4569–4574, 2001.

[9] Karp, R., Reducibility among combinatorial problems, Miller, R. and Thatcher, J. (eds.), *Complexity of Computer Computations*, 85–103. Plenum, 1972.

[10] King, A. D., Przulj, N., and Jurisica, I., Protein complex prediction via cost-based clustering, *Bioinformatics*, 20(17):3013–3020, 2004.

[11] Mewes, H. W., Frishman, D., Gruber, C., Geier, B., *et al.*, MIPS: A database for genomes and protein sequences, *Nucleic Acids Res.*, 28(1):37–40, 2000.

[12] Ng, S.-K., Zhang, Z., and Tan, S.-H., Integrative approach for computationally inferring protein domain interactions, *Bioinformatics*, 19(8):923–929, 2003.

[13] Saito, R., Suzuki, H., and Hayashizaki, Y., Interaction generality, a measurement to assess the reliability of a protein-protein interaction, *Nucleic Acids Res.*, 30(5):1163–1168, 2002.

[14] Sear, R. P., Specific protein-protein binding in many-component mixtures of proteins, *Phys. Biol.*, 1(2):53–60, 2004.

[15] Spirin, V. and Mirny, L. A., Protein complexes and functional modules in molecular networks, *Proc. Natl. Acad. Sci. USA*, 100(21):12123–12128, 2003.

[16] Tan, S.-H., Sung, W.-K., and Ng, S.-K., Discovering novel interacting motif pairs from large protein-protein interaction datasets, *Proc. BIBE*, IEEE Computer Society, 568–575, 2004.

[17] Tong, A. H. Y., and Drees, B., A combined experimental and computational strategy to define protein interaction networks for peptide recognition modules, *Science*, 295(5553):321–324, 2002.

[18] Uetz, P., Giot, L., Cagney, G., Mansfield, T. A., *et al.*, A comprehensive analysis of protein-protein interactions in *Saccharomyces cerevisiae*, *Nature*, 403(6770):623–627, 2000.

[19] van Dongen, S., Graph clustering by flow simulation, Ph.D. thesis, University of Utrecht, 2000.