Revealing Biological Modules via Graph Summarization

Saket Navlakha, Michael C. Schatz, and Carl Kingsford

August 1, 2008

Abstract

The division of a protein interaction network into biologically meaningful modules can aid with automated complex detection and prediction of biological processes and can uncover the global organization of the cell. We propose a novel graph summarization (GS) technique, based on graph compression, to cluster protein interaction graphs into biologically relevant modules. The method is motivated by defining a biological module as a set of proteins that have similar sets of interaction partners. We show this definition, put into practice by a GS algorithm, reveals modules that are more biologically enriched than those found by other methods. We also apply GS to predict complex memberships, biological processes, and co-complexed pairs and show that in most settings GS is preferable over existing methods of protein interaction graph clustering.

1. Introduction

High-throughput experimental protocols have provided a noisy, incomplete picture of the network of interactions between proteins in the cells of many organisms [33, 42]. These putative protein associations, determined largely by yeast two-hybrid [8] and coimmunoprecipitation [10, 19], need to be analyzed to uncover new biology. It is a central computational problem to derive from these networks the sets of proteins that form complexes or are involved in the same biological processes. A common approach to this task is to partition the interaction graph into modules — subsets of proteins — based on the connectivity pattern of the nodes, exploiting the fact that, despite the large amount of noise, proteins with related processes tend to be situated near one another within the network [4, 11, 13, 15, 29]. These modules can be mined to uncover the physical, logical, and evolutionary organization of the cell.

Despite intense interest [6, 20, 27, 30, 32, 34, 39, 40, 43], the proper definition of a network module has remained elusive. It is likely that the appropriate definition must be motivated by the application domain on a case-by-case basis. In the context of uncovering complexes and proteins involved in similar biological processes within protein interaction networks, a natural definition of a module is a set of proteins that all have a similar set of interaction partners. If two proteins interact with similar partners, they likely have related cellular roles. For example, every pair of members in an isolated clique representing a stable complex interact with nearly exactly the same set of proteins: the other members of the clique. Conversely, unrelated proteins are more likely to be widely separated in the network and to thus share few, if any, common interaction partners.

Motivated by this definition of a module, we investigate the application of a technique called *graph summarization* (GS, [22]) to the problem of biological module detection. In general, GS seeks to produce a minimal cost representation of an input graph, compressing it according to the well-known minimal description length (MDL, [28]) ideology. GS has been used to reduce the cost of large networks by 50% or more without loss of information [22]. It achieves such savings by representing nodes with common edge patterns as *supernodes* connected by *superedges* in a summary graph, in conjunction with a *correction list* of topologically inconsistent or missing edges. Graph compression schemes have been previously suggested for community detection [30, 43], although this is their first application in the context of biological networks. In our context, supernodes are composed of proteins with highly similar patterns of interactions. These supernodes are thus natural candidates for biologically meaningful modules.

Detection of modules and complexes within biological networks has received much attention recently, with several widely used algorithms having been developed. The molecular complex detection algorithm (MCODE [2]) is a graph-theoretic clustering algorithm which tries to identify densely connected subgraphs in networks. The Markov clustering algorithm (MCL [38]) is based on simulating flow expansion and flow contraction on graphs. MCL has been applied [7] to the detection of protein families and in a recent comparison of graph clustering algorithms was shown to perform the best at detecting MIPS [12] complexes embedded within an simulated interaction graph [3]. Recently, Newman's spectral algorithm [23] has become a popular choice for community detection, especially within social networks. It is based on modularity [24], a measure of the number of edges falling within modules minus the expected number in a random network (though there

is evidence that this definition prefers modules of a certain characteristic size [9]). Other graph clustering algorithms based on superparamagnetic clustering [34], highly connected subgraphs [25], and restricted neighborhood search clustering [18] have also been applied to biological networks. All of these algorithms use the graph topology only, an approach we follow here as well. This allows us to assess and improve the ability to extract biological information using only interaction data. Any improved network-based analysis can subsequently be incorporated into a more comprehensive, integrative system [16, 26, 35, 36].

We show that our GS approach can recover both stable protein complexes and broader modules enriched for biological processes from a protein interaction network of *Saccharomyces cerevisiae*. Using several evaluation metrics, we show that GS outperforms other standard complex- and module-detection methods, such as MCL [7,38], MCODE [2], and Newman's spectral partitioning algorithm [23]. In particular, a higher percentage of MIPS [12] complexes and interesting Gene Ontology annotations [1, 21] are represented in a statistically significant manner by the modules constructed by GS than those constructed by the other methods.

The modules identified by GS are also more useful for annotating proteins with unknown complex membership and biological processes. We evaluate several different schemes for labeling unannotated proteins within a module, based on transfer of the majority, plurality, or statistically enriched (hypergeometric) annotations. With nearly every approach for both complexes and processes, GS has the highest F-score, which demonstrates its ability to make precise predictions covering large portions of the proteome. Further, GS generally predicts annotations for the most number of proteins under leave-one-out cross-validation testing as compared with existing methods.

These improvements over MCL, MCODE, and Newman's algorithm hold for both an unfiltered yeast network derived from IntAct [17] and generally for a smaller, higher-confidence network constructed by eliminating edges from the network that have weaker support in the literature. Further, while our GS approach works well overall, the predictions made are not a strict superset of those found by MCL or MCODE, and a large fraction of predictions are made only by one method. Thus a combination of multiple methods may be useful to maximize coverage.

Finally, in addition to identifying modules, the corrections list produced by the GS algorithm can also be used to predict co-complexed pairs by identifying missing and false edges in the network. This use of GS generalizes the popular method of predicting edges to complete defective cliques

3

(DCC [41]). Our testing shows GS is more precise than DCC over a wide range of parameters for DCC. In addition, unlike DCC which only predicts missing edges, GS accurately predicts edges to remove from the network as well.

2. Methods

2.1 Graph Summarization

The goal of graph summarization (GS) [22] is to produce a compressed representation of the input network with minimal cost. Given an input graph $G = (V_G, E_G)$, the summarizer produces a new graph $H = (V_H, E_H)$, an onto mapping f from V_G to V_H, and a list of *corrections*. Each node in H is called a *supernode* and is composed of one or more original nodes from V_G collapsed into a single node. Edges between supernodes u and v in E_H imply an edge between all pairs of original nodes contained in u and v. In other words, superedge $\{u, v\}$ implies all edges $f^{-1}(u) \times f^{-1}(v)$ existed in the original graph.

To account for errors that would be introduced by this superedge expansion rule, the summarizer also produces a list $\mathcal{C}$ of corrections. This list augments the superedges with edges from the original network that must be added or subtracted to fully reconstruct the original network. Edges that are implied by a superedge but that are missing from E_G are recorded as edges to subtract from the compressed network (negative corrections); on the other hand, rare edges that are not covered by any superedge are recorded as edges that must be added (positive corrections). Together the supernodes and superedges effectively summarize the original network and reveal the major topological structure, while the corrections list reveals the exceptions to that overall structure. The original graph G can then be exactly reconstructed from H by expanding each supernode into its constituent nodes, adding all edges implied by the superedges, and then applying the positive and negative corrections from the corrections list.

In accordance with the MDL principle, GS attempts to find the representation $H, \mathcal{C}$ of G that minimizes the number of superedges plus the number of corrections, $|E_H| + |\mathcal{C}|$. For example, if G is composed of two nearly complete cliques of size m and n and connected by a single edge, then G will be compressed into H composed of two supernodes with self-edges, and a corrections list containing the two missing edges from the cliques (negative corrections) and the individual

4

edge between the cliques (positive correction). The cost of the H is the number of superedges (2) plus the number of edges on the correction list (3) giving a dramatic saving over the original cost $\binom{m}{2} + \binom{n}{2} + 1$.

Here, a compression of the protein interaction network is found using the greedy method described in Navlakha *et al.* [22]. This method iteratively merges the pair of nodes that results in the greatest cost reduction (generally the pair with the most similar set of edges), much like bottom-up hierarchical clustering. The greedy algorithm, however, naturally terminates when the cost of merging any pair of nodes increases the cost of H. In the context of finding modules within protein networks, we add a self-edge to each node before compression so that every member of a clique will have exactly the same neighbors. Once the compressed graph is found, the supernodes of proteins with similar interaction partners are taken to represent biological modules.

2.2 Interaction Network

We constructed a protein interaction network called Y_{ppi} for the yeast *Saccharomyces cerevisiae* from data deposited in the IntAct [17] database. The Y_{ppi} nework includes all deposited edges and contains 5,492 proteins with 40,332 interactions. Most of these interactions were derived from yeast-two-hybrid and TAP experiments, while a smaller number were obtained thorough traditional low-throughput assays.

2.3 Complex and Biological Process Annotations

We assess the biological quality of the modules found by GS by their ability to recapitulate known protein complexes and biological processes. We use known annotations for assessment and prediction only — no annotations were used when constructing the modules.

Yeast complex data was taken from the MIPS [12] complex catalog. This set of complexes has been widely used to assess computational methods [16,26,41]. We ignore complexes from the "550" section of the MIPS tree, which represent computationally inferred annotations. We are interested in making predictions which are as specific as possible, and therefore use the leaf set of complexes in the catalog. The leaf set contains 267 complexes, of which 266 are represented by at least one protein in the Y_{ppi} network.

We obtained known biological process annotations from the Saccharomyces Genome Database

(SGD [5]) corresponding to the biological process sub-ontology of the Gene Ontology (GO) [1]. We used the ancestor relationships of GO to determine which proteins participate in each biological process. To ensure that we are detecting processes at an interesting level of specificity, we focus on a comprehensive, expert-curated subset of biologically interesting annotations selected by Myers *et al.* [21]. All analysis of biological processes in this paper is done using this set of 295 terms, 182 of which are represented in the Y_{ppi} network by at least one protein.

2.4 Measuring Enrichment of Biological Processes and Complex Membership

The goal of module-detection algorithms is to discover modules from an interaction network that are "enriched." Proteins in a module should all be part of the same complex or biological process. We measure the enrichment of a given annotation F in a given module M with the hypergeometric P-value, computed by $\sum_{i=k}^{m} \frac{\binom{f}{i}\binom{n-f}{m-i}}{\binom{n}{m}}$, where n is the number of nodes in the network, m is the number of nodes in M, f is the number of nodes in the network annotated with F, and k is the number of nodes in M annotated with F. The computed hypergeometric P-values are Bonferroni corrected to account for multiple testing. An annotation with a P-value of less than 0.001 was considered enriched.

We use two measures to assess the quality of the modules. The first is the percentage of complexes or biological processes that are enriched in at least one module. This measures the "diversity of annotations" present in the modules, with large values indicating that a wide spectrum of biology is represented by the modules. Small values, in contrast, suggest the modules recapitulate only a few biological annotations (such as the ribosome). Conversely, we measure the percentage of the reported modules enriched for at least one annotation, as used previously in [37].

2.5 Predicting New Annotations

Another test of the quality of a module decomposition is how well it can be used to make new predictions for membership within complexes or biological processes. Given a decomposition of proteins into sets of modules, we employ several module-assisted prediction techniques to infer new annotations for proteins of unknown complexes or biological processes. Each technique is based on transferring an annotation that is common to many proteins in a module to every protein in the module.

The first approach, "majority," transfers an annotation to a protein if more than 50% of the other annotated proteins in the module have that annotation. If no annotation exists on more than 50% of the proteins (or if there exists only one annotated protein in the module), no predictions are made. Relaxing the requirement for a strict majority leads to the second method of annotation transfer, here called "plurality." Under this scheme, a protein is predicted to have the most common annotation within its module. The third approach transfers all annotations that are statistically enriched within a module to every protein of the module. An annotation with a Bonferroni-corrected hypergeometric P-value of < 0.001 was considered enriched. In all cases, modules consisting of a single protein are ignored. These schemes are applied separately for complex membership and biological process annotations.

We tested the efficacy of these schemes for each module detection algorithm using leave-one-out cross-validation. For every annotated protein p, all of its annotations were forgotten, the majority, plurality, or enriched annotations were computed for its module and then transferred to p as predicted annotations. If multiple annotations were transfered, each transfered annotation was considered as one prediction. A prediction was correct if the protein is known to belong to that complex or biological process, and incorrect otherwise. Naturally, given the incomplete state of knowledge, some "incorrect" predictions may in fact represent correct, new biology.

We measure performance by the precision and recall of the predictions made. Precision is the percent of predicted annotations that are correct. Recall is the number of correct annotations made divided by the total number of possible correct annotations. There are a large number of possible annotations over all the proteins in the network, which generally leads to low recall for all methods. Even relatively low recall values, however, represent hundreds of correct annotations inferred using only interaction topology. We also report the widely-used F-score to evaluate a method's balance between precision and recall. The F-score is a number between 0 and 1 (the larger the better) and is computed as the harmonic mean of the precision and recall: $2 \cdot \text{precision} \cdot \text{recall}/(\text{precision} + \text{recall})$.

2.6 Clustering Parameter Variation

We compare the results of GS with those obtained by MCL, MCODE, and the spectral partitioning algorithm of Newman (which we label here as NSP). Both MCL and MCODE make use of parameter values that can affect the clustering they produce. MCL uses a single parameter I that indirectly

controls the sizes of the obtained modules. Larger values of I favor smaller modules. We tried 9 different values of I between 1.8 and 4.6, and chose $I = 3.8$ because this value maximized the complex enrichment and predictive performance on Y_{ppi} using both the majority and plurality rules. We also report the predictive performance using the suggested default value of $I = 2.0$, which was considerably worse. A recent survey [3] found the value of 1.8 for this parameter was most effective, but with our data set, this value produced clusters that were too large, and did not perform as well as $I = 3.8$. The performance of MCL presented below should thus be considered *trained*.

MCODE supports several parameters (degree cutoff, node score cutoff, haircut, fluff, k-core, max depth). Experiments were run using both the default parameters as well as the parameters suggested in [3]. The latter set of parameters produced clusters with the greatest predictive precision (although less than 6% of the proteins were clustered). The "node score cutoff" had the greatest effect on the modules. It influences the cluster size and was set to 0.0 as in [3] to favor small clusters. Again, because parameters were selected based on their performance on the test set, the results for MCODE should also be considered *trained*. While an exhaustive search of parameter space may reveal a set of parameters for which performance is improved, it is unclear in practice how these parameters should be set without fitting to a training set.

GS and NSP require no parameters to be set.

3.   Results and Discussion

3.1   Application of Graph Clustering Techniques

Each of the MCODE, MCL, NSP, and GS methods were run on the unweighted Y_{ppi} network. GS, MCL, and NSP give a complete partitioning of the graph, but MCODE will not necessarily place all proteins in a module. In all cases, modules that contained only a single protein were discarded. All numbers presented in this section are for the tuned versions of MCL and MCODE. The module decompositions were very different amongst the four algorithms.

NSP divides the network into only 8 modules, consistent with the previous observation that the modularity statistic does not easily find small modules within larger graphs [9]. While these large modules produced by NSP do yield some biological information (see below), they are generally difficult to interpret and do not likely correspond to natural biological divisions of the graph.

MCODE also produced far fewer modules than either MCL or GS. MCODE created 80 modules covering only 308 proteins (less than 6% of the total network), while MCL covered $4,383$ proteins with 1185 modules. GS produced the largest number of modules: $1,632$ modules that covered $4,997$ proteins. GS thus placed nearly all proteins in modules of size ≥ 2. The average size of the modules produced by GS was 3.1 and 1,043 of the modules contained only two proteins. The largest module produced by GS contained 129 proteins.

Figure 1 shows a visualization of the high-level summary structure returned by GS on the Y_{ppi} network. For clarity, we only show the summary induced by the supernodes containing at least two proteins and with at least one superedge (which may be a self-loop). Shown at the bottom are some of the supernodes GS found which represent cliques or near-cliques, indicated by a supernode with a self-loop.

3.2 Comparison of Complex and Biological Process Enrichment of the Modules

A primary test of the quality of a module decomposition is its similarity to the modules induced by known biology. We use two different evaluation measures to assess the modules produced by each method (see Section 2.4 for details). A summary of the performance of these measures on both protein complexes and biological processes is shown in Figure 2.

The modules produced by GS cover the largest number of complexes and biological processes (Figure 2a). Out of the 266 complexes appearing at least once in our network, 152 (57.1%) are enriched in at least one GS module. MCL has 40.9% of the complexes enriched in at least one module. MCODE is limited in its ability to make predictions for many complexes because it only clusters 308 proteins and thus only covers 17.3% of the complexes. Only 16.5% are enriched in some NSP module, despite NSP clustering all nodes. This indicates that GS performs well on many different biological units and is not simply highly successful for a few large complexes.

All of the 8 modules produced by NSP are enriched for at least one complex and typically enriched for several. This suggests that the graph partition produced by NSP does yield some useful information. However, because the modules are so large (average size 687) it isn't clear how to make use of them. Further, as we will see below, the percentage of modules enriched is not a good indicator of a method's ability to predict annotations.

GS also has the largest percentage (70.9%) of biological processes enriched in at least one

module (Figure 2b). MCL is again the second best-performing method with 63.7% of processes enriched. NSP and MCODE again have the largest percentage of their few modules enriched for some biological process. This, combined with the similar results for complex enrichment, suggests that GS gives the best coverage of possible biology. This is in contrast with MCODE, which produces good modules, but leaves 94% of the graph unclustered.

3.3 Improvement in Module-Assisted Annotation Prediction

Ultimately, the goal of dividing an interaction network into modules is to learn new biology. We test the utility of the various module-decomposition methods for predicting new complex and biological process annotations using three different methods for annotation transfer within a module (see Section 2.5). Figure 3 shows the performance of these prediction schemes for both complex membership and biological processes. Because MCL and MCODE both require parameter tuning, we include in Figure 3 the results of both approaches with and without tuning. The tuned versions are marked with a + symbol, and show a significant improvement in performance when compared to the untuned versions.

Complex membership prediction.   The most conservative annotation transfer technique is to label each protein in a module with all the majority annotations within the module. Of the three annotation transfer methods, the majority approach results in fewer correctly predicted annotations, although these annotations tend to be more accurate. For complex membership, GS makes more correct predictions than MCL (305 for GS compared with 241 for MCL), and has higher recall (22.1% vs. 17.5%) and precision (91.9% vs. 79.3%). MCODE makes slightly more precise predictions than GS (92.7% vs. 91.9%) but has a much lower recall (9.2% vs. 22.1%). Further, the predictions made by GS covers a larger number of proteins than MCL and MCODE (265 for GS, 219 for MCL, 104 for MCODE). NSP performs the worst in all cases due to its large module sizes.

When the transfer rule is relaxed to permit predictions based on the most common complex annotations within a module (plurality), many more correct predictions are made, though the predictions are less accurate. For complex membership, GS is again able to make more correct predictions (598) than MCL (436) and MCODE (164). Although MCODE's predictions are considerably more accurate than GS (86.8% vs. 60.5%), GS's recall is almost four times that of MCODE's (11.9% vs.

43.4%). GS also predicts complexes for substantially more proteins than MCODE (530 vs. 140). This indicates that MCODE is able to make a few, precise predictions, but is not nearly as good as GS for covering a greater part of the proteome (see F-measure below). In general, the plurality rule makes less precise predictions compared with majority, but greatly improves recall. It also makes predictions for roughly twice as many different proteins than majority.

A better trade-off between precision and recall is obtained by using the hypergeometric enrichment to select the complexes that should be transfered within the modules. Using this method, the GS modules make 330 correct predictions, more than with any other algorithm. It also has the highest recall (23.9% compared with MCL at 20.0%) and the highest precision (86.6% compared with MCODE at 83.6%). NSP makes predictions for slightly more proteins than GS (293 vs. 285) but NSP otherwise suffers from both low precision (10.6%) and low recall (23.1%). While the hypergeometric approach does not produce quite as many correct predictions as using the plurality rule, it makes more accurate predictions.

Figure 3 shows that the transfer schemes using the GS modules are all Pareto optimal: no other method dominates them in both precision and recall. For each transfer rule, GS also has the largest F-score compared to any other method. For example, using the majority rule, GS has a F-score of 35.7% compared to the next best at 28.7% for MCL. In general, MCL consistently does second-best; MCODE suffers from the fact that its clustering itself only covers 6% of the total nodes in the network; and NSP fairs poorly.

Repeating the above experiments using more general complex annotations coming from level 3 of the MIPS complex catalog tree showed little change in the results. Hence, GS appears to work well for complexes with varying levels of specificity.

Biological processes prediction. Using the three annotation transfer schemes to predict biological process annotations for proteins reveals a similar pattern as with complexes. For the methods that produce smaller module sizes (GS and MCL), plurality makes the highest number of correct predictions, although at the expense of much lower accuracy compared with majority and hypergeometric. Interestingly, the hypergeometric rule makes fewer biological process predictions than majority, but is slightly more accurate – the opposite of what happens for complex prediction.

GS makes the largest number of correct predictions under all schemes except the hypergeometric

11

(685 compared with 779 for MCL), but our accuracy is 26% higher than that of MCL (88.0% compared with 62.5%). The accuracy of both GS and MCL is slightly lower for predicting biological processes than for predicting complex membership.

In general, the hypergeometric produces more correct predictions than majority at a slight loss in precision for complex annotations. For biological processes, the opposite is true: majority produces more correct predictions at a slight loss in precision. Although always lower in accuracy, the plurality rule makes more correct predictions and has a larger coverage than majority or hypergeometric for both biological processes and complexes. Thus, the choice of annotation transfer approach must be made with a desired level of predictions, precision, and coverage in mind.

Comparison of prediction sets. Because of their very different philosophies and approaches, GS, MCODE, and MCL complement each other well for the detection of biologically meaningful modules. GS produces a large number of correct predictions that none of the other methods make. Of the 305 correct complex predictions made by GS using the majority rule, 135 of the predictions (or 44%) are not made by either MCL or MCODE. Similarly, 39% of MCL's predictions are not made by either GS or MCODE. Of MCODE's 127 predictions, 23% are unique. The same is true using the hypergeometric rule for annotation transfer: Of the predictions made by GS, MCL, and MCODE, 44%, 38%, and 11% respectively are unique to each method. These numbers also carry over similarly to predicting biological processes: using the majority rule, 40% (550), 36% (371), and 17% (107) of the predictions made by GS, MCL, and MCODE, respectively are unique.

The large number of unique and correct predictions made by each method suggests each are able to uncover new biology that the other methods cannot. It also suggests that combining predictions from various methods may be useful, and emphasizes the point that no one approach to clustering is likely to be universally applicable.

3.4 High-Confidence Network

Protein interaction networks derived from high-throughput experiments are known to be noisy, with false positive rates potentially reaching 90% [14]. As a result, we constructed an additional high-confidence yeast interaction network, $Y_{high\text{-}conf}$, that includes edges from IntAct that are associated with more than one PubMed identifier. (If the same identifier was listed twice by IntAct in support

of an interaction, it was counted twice and thus included in $Y_{\text{high-conf}}$.) Two experiments are likely to both capture a true interaction, but are less likely to both return a false interaction. The $Y_{\text{high-conf}}$ network contains 2,604 proteins and 8,341 interactions, fewer than half the proteins of the Y_{ppi} network. Using $Y_{\text{high-conf}}$, we reproduce the results described above to probe the effects of noise on the performance of the module-detection methods. The qualitative performance is similar to using the more comprehensive unfiltered network (Table 1). The GS approach makes more accurate complex predictions using any of the three annotation transfer schemes when compared with MCL and NSP. MCODE generally makes slightly more precise predictions, but at a large loss in recall and predicted annotations for different proteins.

3.5 Predicting Co-Complexed Pairs

GS has the benefit of producing both a set of modules and a list of corrections to that modular structure. Analysis of the corrections list can lead to further insights. For example, we can use the corrections to predict missing edges that are indicative of co-complexed proteins. We consider negative corrections (where an edge must be removed from the summary to match the original graph) as predictions of pairs that are co-complexed. Positive corrections (where an edge must be added to restore the input graph) are predictions of noisy or erroneous edges that should be filtered. This edge prediction approach thus has the highly desirable feature of making predictions for both edges to add and edges to remove. GS applied this way can be thought of as a generalization of the popular method of completing defective cliques (DCC, [41]) for predicting edges within apparent protein complexes.

We evaluated the performance of the GS co-complex predictions by comparing the edges predicted for $Y_{\text{high-conf}}$ to a gold standard set of co-complexed pairs composed of 11,014 edges between proteins annotated from the same MIPS complex (at the leaves of the hierarchy) and a negative set containing 2,705,720 edges between proteins annotated with different subcellular localizations (taken from [41]). For this test we used the unmodified graph summarization algorithm that does not add self edges before compression.

The DCC algorithm takes two parameters k and l controlling the overlap and size of defective cliques considered. We compared the results of the GS method to the results of the DCC algorithm over a range of parameters for k ($4 \leq k \leq 10$) and l ($2 \leq l \leq 5$). The precision of GS (66.7% for

13

224 predicted new edges) is better than DCC under all parameters for DCC (between 37.1% for 2317 predictions using $k = 4$, $l = 5$, and 62.5% for 39 predictions using $k = 9$, $l = 2$). Further, GS has very high specificity for accurately filtering incorrect edges from the network (97.4% for 3331 predicted edges). Because DCC does not predict edges to filter from the network, it is not applicable to compute its specificity. On Y_{ppi}, DCC achieved higher precision for some parameters than GS, although the precision of both GS and DCC was generally poor. This is most likely because of the larger number of false edges in Y_{ppi} obfuscated the true complexes.

4. Conclusion

While the definition of a biological network "module" will likely remain unsettled for some time to come, the results presented here suggest it is biologically meaningful and informative to define a module as a set of proteins that have similar interaction partners. Such a definition generalizes cliques, defective cliques, and other types of dense subgraphs. Based on the minimum description length principle, the GS procedure has no parameters that must be optimized, unlike both MCL and MCODE. Using GS to predict membership in protein complexes and biological processes led to increased performance compared with other approaches, even when their parameters are tuned to fit the data. The GS modules also cover a larger fraction of complexes and biological processes than other methods. GS also works well for predicting co-complexed pairs. The general utility of graph summarization for extracting meaning biological modules suggests that GS will be a useful technique for the analysis of biological networks.

A. Figure Captions

Figure 1.   A visualization (made in Cytoscape [31]) of the summary structure returned by GS on the Y_{ppi} network. Circles represent supernodes, with sizes proportional to their member proteins. Lines represent superedges.

Figure 2.   Plot comparing each approach's ability to identify modules that are enriched for (A) complexes, and (B) biological processes. NSP and MCODE have the greatest percentage of modules enriched, but NSP only returns 8 modules, and MCODE only clusters 6% of the network. Moreover, it is not clear how meaningful it is to have a large module that is only enriched for one annotation. The modules returned by GS are enriched for a greater variety of annotations.

Figure 3.   Precision-Recall plots showing the predictive performance using each approach for (A) complexes, and (B) biological process annotations, using the majority, plurality, and hypergeometric transfer rules. The lines highlight the best performing methods. All transfer schemes for GS are Pareto-optimal. The versions of MCL and MCODE with tuned parameters are indicated with a + sign following their names.

Table 1.   Predicting MIPS complexes and Gene Ontology biological processes in $Y_{\text{high-conf}}$. Columns list number of proteins clustered (**n**) and number of modules (**m**). For each transfer method, the number of proteins for which at least one correct prediction is made (**prot**), the recall (**R**), and precision (**P**) are given. Methods marked with + used tuned parameters.
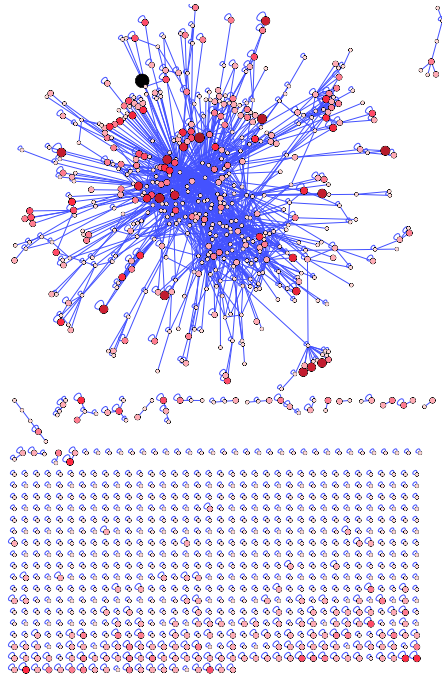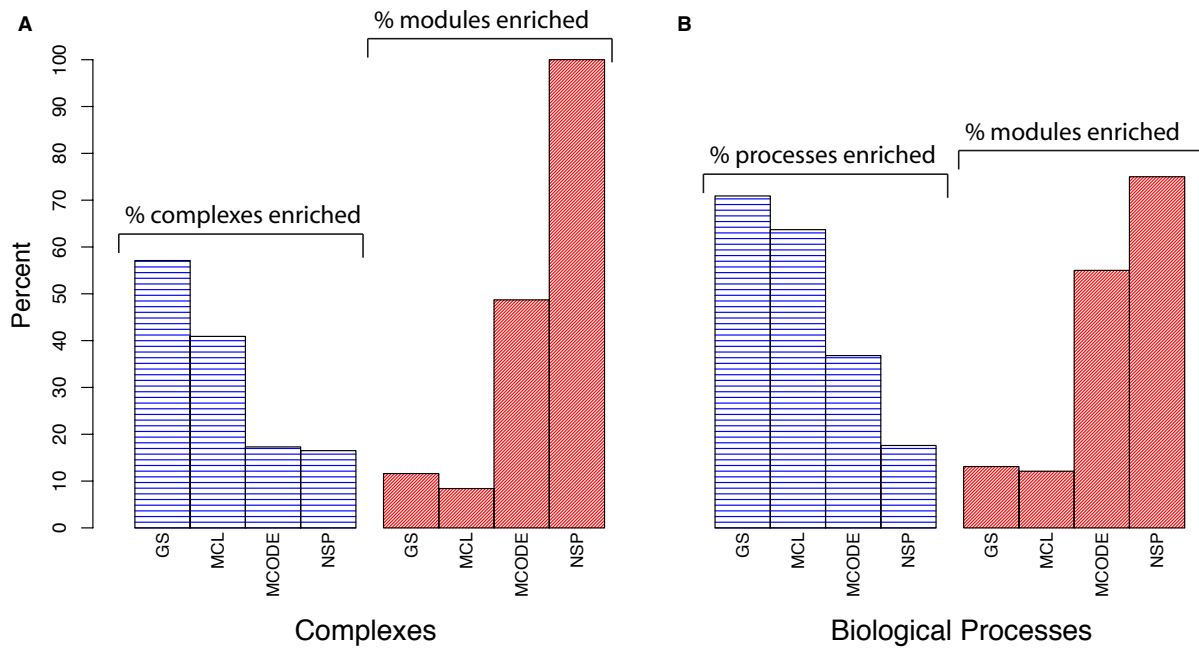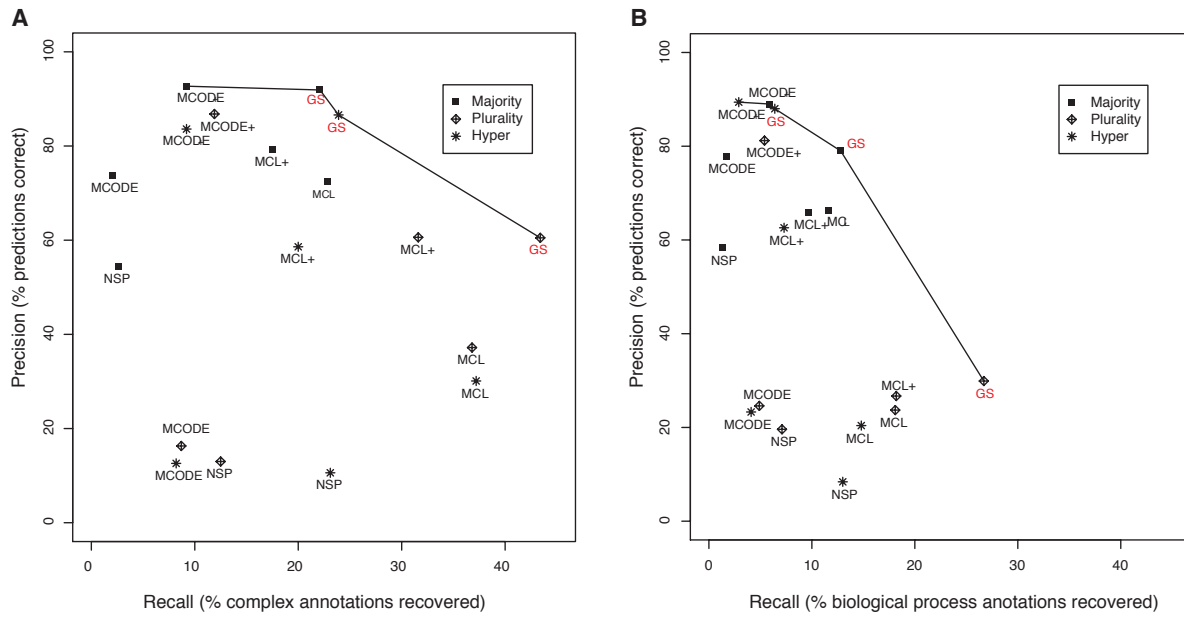
B. Figures and Tables

Figure 1

Figure 2

Figure 3

Table 1

Annot	Method	Clusters		Majority			Plurality			Hyper		
		n	**m**	**prot**	**R**	**P**	**prot**	**R**	**P**	**prot**	**R**	**P**
Complexes	GS	2336	807	321	31.1	87.2	589	56.3	77.4	312	30.0	85.4
	MCL+	2323	615	386	36.4	84.0	571	55.2	71.1	395	38.6	67.7
	MCODE+	293	72	132	13.0	88.4	166	16.0	87.3	129	12.0	88.1
	NSP	2604	81	236	22.5	66.3	434	40.9	38.8	577	56.2	23.3
	MCL	2570	466	440	41.8	77.1	606	57.3	60.5	582	57.3	50.8
	MCODE	717	89	174	17.7	69.6	233	23.0	55.7	250	25.4	56.8
Processes	GS	2336	807	685	23.8	82.1	1308	41.9	51.9	345	8.9	86.3
	MCL+	2323	615	878	27.9	78.4	1328	37.7	53.4	632	18.9	82.6
	MCODE+	293	72	254	10.1	90.2	262	9.0	88.9	136	3.8	89.5
	NSP	2604	81	568	13.9	66.2	1014	17.2	43.3	1061	29.0	30.2
	MCL	2570	466	1048	31.6	75.6	1462	35.7	50.5	905	26.8	66.5
	MCODE	717	89	439	14.7	81.9	507	13.6	74.3	382	10.9	64.1

References

[1] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock. Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat Genet*, 25(1):25–29, May 2000.

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

[3] S. Brohee and J. van Helden. Evaluation of clustering algorithms for protein-protein interaction networks. *BMC Bioinformatics*, 7:488+, November 2006.

[4] D. Bu, Y. Zhao, L. Cai, H. Xue, X. Zhu, H. Lu, J. Zhang, S. Sun, L. Ling, N. Zhang, G. Li, and R. Chen. Topological structure analysis of the protein-protein interaction network in budding yeast. *Nucl. Acids Res.*, 31(9):2443–2450, May 2003.

[5] J. M. Cherry, C. Ball, S. Weng, G. Juvik, R. Schmidt, C. Adler, B. Dunn, S. Dwight, L. Riles, R. K. Mortimer, and D. Botstein. Genetic and physical maps of saccharomyces cerevisiae. *Nature*, 387(6632 Suppl):67–73, May 1997.

[6] A. Clauset, M. E. J. Newman, and C. Moore. Finding community structure in very large networks. *Phys. Rev. E*, 70(6):066111, Dec 2004.

[7] A. J. Enright, S. Van Dongen, and C. A. Ouzounis. An efficient algorithm for large-scale detection of protein families. *Nucleic Acids Res*, 30(7):1575–1584, April 2002.

[8] S. Fields and O. Song. A novel genetic system to detect protein-protein interactions. *Nature*, 340(6230):245–246, July 1989.

[9] S. Fortunato and M. Barthelemy. Resolution limit in community detection. *Proc. Natl. Acad. Sci. USA*, 104(1):36–41, January 2007.

[10] A.-C. Gavin, P. Aloy, P. Grandi, R. Krause, M. Boesche, M. Marzioch, C. Rau, L. J. Jensen, S. Bastuck, B. Dümpelfeld, A. Edelmann, M.-A. Heurtier, V. Hoffman, C. Hoefert, K. Klein,

M. Hudak, A.-M. Michon, M. Schelder, M. Schirle, M. Remor, T. Rudi, S. Hooper, A. Bauer, T. Bouwmeester, G. Casari, G. Drewes, G. Neubauer, J. M. Rick, B. Kuster, P. Bork, R. B. Russell, and G. Superti-Furga. Proteome survey reveals modularity of the yeast cell machinery. *Nature*, 440:631–636, January 2006.

[11] R. Guimera and L. A. N. Luis. Functional cartography of complex metabolic networks. *Nature*, 433(7028):895–900, February 2005.

[12] U. Guldener, M. Munsterkotter, G. Kastenmuller, N. Strack, J. van Helden, C. Lemer, J. Richelles, S. J. Wodak, J. Garcia-Martinez, J. E. Perez-Ortin, H. Michael, A. Kaps, E. Talla, B. Dujon, B. Andre, J. L. Souciet, J. De Mon tigny, E. Bon, C. Gaillardin, and H. W. Mewes. Cygd: the comprehensive yeast genome database. *Nucleic Acids Research*, 33(Supplement 1):D364+, January 2005.

[13] J.-D. J. Han, N. Bertin, T. Hao, D. S. Goldberg, G. F. Berriz, L. V. Zhang, D. Dupuy, A. J. M. Walhout, M. E. Cusick, F. P. Roth, and M. Vidal. Evidence for dynamically organized modularity in the yeast protein-protein interaction network. *Nature*, 430(6995):88–93, July 2004.

[14] T. G. Hart, A. K. Ramani, and E. M. Marcotte. How complete are current yeast and human protein-interaction networks? *Genome Biology*, 7:120+, December 2006.

[15] L. H. Hartwell, J. J. Hopfield, S. Leibler, and A. W. Murray. From molecular to modular cell biology. *Nature*, 402(6761 Suppl), December 1999.

[16] R. Jansen, H. Yu, D. Greenbaum, Y. Kluger, N. J. Krogan, S. Chung, A. Emili, M. Snyder, J. F. Greenblatt, and M. Gerstein. A bayesian networks approach for predicting protein-protein interactions from genomic data. *Science*, 302(5644):449–453, October 2003.

[17] S. Kerrien, Y. Alam-Faruque, B. Aranda, I. Bancarz, A. Bridge, C. Derow, E. Dimmer, M. Feuermann, A. Friedrichsen, R. Huntley, C. Kohler, J. Khadake, C. Leroy, A. Liban, C. Lieftink, L. Montecchi-Palazzi, S. Orchard, J. Risse, K. Robbe, B. Roechert, D. Thorney-croft, Y. Zhang, R. Apweiler, and H. Hermjakob. Intact–open source resource for molecular interaction data. *Nucleic Acids Research*, 35(Database issue), January 2007.

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

[19] N. J. Krogan, G. Cagney, H. Yu, G. Zhong, X. Guo, A. Ignatchenko, J. Li, S. Pu, N. Datta, A. P. Tikuisis, T. Punna, J. a. M. Peregrãn Alvarez, M. Shales, X. Zhang, M. Davey, M. D. Robinson, A. Paccanaro, J. E. Bray, A. Sheung, B. Beattie, D. P. Richards, V. Canadien, A. Lalev, F. Mena, P. Wong, A. Starostine, M. M. Canete, J. Vlasblom, S. Wu, C. Orsi, S. R. Collins, S. Chandran, R. Haw, J. J. Rilstone, K. Gandi, N. J. Thompson, G. Musso, P. St Onge, S. Ghanny, M. H. Y. Lam, G. Butland, A. M. Altaf-Ul, S. Kanaya, A. Shilatifard, E. O'Shea, J. S. Weissman, J. C. Ingles, T. R. Hughes, J. Parkinson, M. Gerstein, S. J. Wodak, A. Emili, and J. F. Greenblatt. Global landscape of protein complexes in the yeast saccharomyces cerevisiae. *Nature*, 440:637–643, March 2006.

[20] R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Trawling the web for emerging cyber-communities. *Comput. Netw.*, 31(11-16):1481–1493, 1999.

[21] C. L. Myers, D. R. Barrett, M. A. Hibbs, C. Huttenhower, and O. G. Troyanskaya. Finding function: evaluation methods for functional genomic data. *BMC Genomics*, 7:187+, July 2006.

[22] S. Navlakha, R. Rastogi, and N. Shrivastava. Graph summarization with bounded error. In *SIGMOD 2008: Proceedings of the 2008 ACM SIGMOD International Conference on Management of data*, pages 419–432, New York, NY, USA, 2008. ACM.

[23] M. E. J. Newman. Modularity and community structure in networks. *PNAS*, 103(23):8577–8582, June 2006.

[24] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)*, 69(2), 2004.

[25] N. Przulj, D. A. Wigle, and I. Jurisica. Functional topology in a network of protein interactions. *Bioinformatics*, 20(3):340–348, February 2004.

[26] J. Qiu and W. S. Noble. Predicting co-complexed protein pairs from heterogeneous data. *PLoS computational biology*, 4(4), April 2008.

[27] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi. Defining and identifying communities in networks. *Proc Natl Acad Sci U S A*, 101(9):2658–2663, March 2004.

[28] J. Rissanen. Modeling by shortest data description. *Automatica*, 14:465–471, 1978.

[29] A. W. Rives and T. Galitski. Modular organization of cellular networks. *Proceedings of the National Academy of Sciences*, 100(3):1128–1133, February 2003.

[30] M. Rosvall and C. T. Bergstrom. An information-theoretic framework for resolving community structure in complex networks. *PNAS*, 104(18):7327–7331, May 2007.

[31] P. Shannon, A. Markiel, O. Ozier, N. S. Baliga, J. T. Wang, D. Ramage, N. Amin, B. Schwikowski, and T. Ideker. Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Res*, 13(11):2498–2504, November 2003.

[32] R. Sharan, I. Ulitsky, and R. Shamir. Network-based prediction of protein function. *Mol Syst Biol*, 3, March 2007.

[33] B. A. Shoemaker and A. R. Panchenko. Deciphering protein-protein interactions. part i. experimental techniques and databases. *PLoS Computational Biology*, 3(3):e42+, March 2007.

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

[35] A. Tanay, R. Sharan, M. Kupiec, and R. Shamir. Revealing modularity and organization in the yeast molecular network by integrated analysis of highly heterogeneous genomewide data. *Proc Natl Acad Sci U S A*, 101(9):2981–2986, March 2004.

[36] O. G. Troyanskaya, K. Dolinski, A. B. Owen, R. B. Altman, and D. Botstein. A bayesian framework for combining heterogeneous data sources for gene function prediction (in saccharomyces cerevisiae). *Proc Natl Acad Sci U S A*, 100(14):8348–8353, July 2003.

[37] I. Ulitsky and R. Shamir. Identification of functional modules using network topology and high-throughput data. *BMC Systems Biology*, 1(1), 2007.

[38] S. van Dongen. A new cluster algorithm for graphs. In *281*, page 42. Centrum voor Wiskunde en Informatica (CWI), ISSN 1386-3681, 31 1998.

[39] K. Wakita and T. Tsurumi. Finding community structure in mega-scale social networks, 2007.

[40] C. Wang, C. Ding, Q. Yang, and S. R. Holbrook. Consistent dissection of the protein interaction network by combining global and local metrics. *Genome Biology*, 8:R271+, December 2007.

[41] H. Yu and A. Paccanaro. Predicting interactions in protein networks by completing defective cliques. *Bioinformatics*, 22(7):823–829, April 2006.

[42] X. Zhu, M. Gerstein, and M. Snyder. Getting connected: analysis and principles of biological networks. *Genes Dev*, 21(9):1010–1024, May 2007.

[43] E. Ziv, M. Middendorf, and C. H. Wiggins. Information-theoretic approach to network modularity. *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)*, 71(4), 2005.