Deterministic Graph–Theoretic Algorithm for Detecting Modules in Biological Interaction Networks

Roger L. Chang*, Feng Luo†, Stuart Johnson‡ and Richard H. Scheuermann§
*Bioinformatics Graduate Program, University of California San Diego, La Jolla, CA 92093, USA
†Department of Computer Science, Clemson University, Clemson, SC 29634 USA
‡Texas Advanced Computing Center, University of Texas Austin, Austin, TX 78758 USA
§Division of Biomedical Informatics, University of Texas Southwestern Medical Center, Dallas, TX 75390 USA
§Contact author: *richard.scheuermann@utsouthwestern.edu*

***Abstract*—Accumulating evidence suggests that biological systems exhibit modular organization. Accurate identification of modularity is vital for understanding this organization. A recent approach, Modules of Networks (MoNet), introduced an intuitive module definition and a clear detection method based on a ranked list of edges generated by the Girvan-Newman (G-N) algorithm. The resulting modules from a yeast network showed significant association with known biological processes, indicative of the method's utility. Despite MoNet's usefulness, systematic bias in the method leads to varied results across trials. MoNet modules also exclude some network regions. Such deficiencies limit meaningful analysis of a network. To address these shortcomings, we have developed a deterministic G-N algorithm (dG-N) and a new agglomerative algorithm, Deterministic Modularization of Networks (dMoNet). dMoNet simultaneously processes structurally equivalent edges while preserving the intuitive foundations of the MoNet algorithm. dMoNet also includes all network nodes in the identified modules, thus generating hypotheses with respect to a full network. A GO-based method for comparison of dMoNet to other modularization methods, including a currently favored algorithm, shows an overall better performance by dMoNet when analyzing large-scale yeast and human protein interaction networks. This comparison method comprises a rational evaluation of the quality of functional modules identified from large-scale interaction networks. The code and resulting data from this work are available upon request.**

***Index Terms*—Algorithms, graph theory, interaction networks, modules.**

I. Introduction

Network analysis offers an effective and manageable approach for studying biological system organization. A number of studies provide evidence suggesting that the organization of biological systems exhibits a modular structure [3], [12], [21], [22]. The general understanding of a module is a group of system components that together perform a distinct semi-autonomous function within the context of the entire system. The overall function of a complete biological system is to maintain a living state and to carry out essential life activities. Therefore the intuitive role of modules in the context of biological systems most closely corresponds to discrete biological processes, for example as defined by the Gene Ontology (GO) [1]. Accurately identifying the modular structure of biological networks is vital for understanding the organization of biological systems and the functional relationships among system processes.

Biological systems exhibit complexity spanning many dimensions and levels of granularity. However, it is reasonable to initially approach the problem of identifying the modular structure of a network by focusing on the most fundamental network features, the nodes and edges of the network graph, which often correspond to the molecular constituents of the system and their physical or functional relationships. Considering a system at this level excludes dynamics, directionality and other qualitative and quantitative parameters. A recent study of biological network module organization introduces an intuitive definition of modules and a clear method for identifying them from these fundamental structural features of networks [17]. The method, Module of Network (MoNet), draws from the informative measure of edge betweenness defined by Girvan and Newman for social networks [10], adapted from the vertex betweenness measure [8]. Edge betweenness is defined as the number of shortest paths between all pairs of nodes that run through an edge. This measure quantifies an edge's relative capacity as a mediator between all network components. Edges lying between modules tend to have higher betweenness than edges contained within them. MoNet employs the divisive Girvan-Newman (G-N) algorithm [10], [19] in order to harness the informative value of edge betweenness by iteratively removing the highest betweenness edge in the network and creating an edge list ranked by the order of edge removal. MoNet is an agglomerative algorithm that uses the ranked list of edges generated by the G-N algorithm in reversed removal order to assemble modules by iteratively evaluating the addition of these edges to the network and then testing the modularity of the resulting subgraphs by assessing the ratio of internal to external edges. Thus the MoNet module definition captures the intuitive understanding of a module, i.e. a module has more internal interactions than external interactions, which distinguishes it from the rest of the network. The modules isolated from the *S. cerevisiae* core protein-protein interaction network [29] by the MoNet method showed significant association with GO biological processes, indicative of the method's utility. MoNet is also capable of detecting modules with a variety of topological motifs that cannot be detected using alternative methods based on finding highly-connected dense subgraphs.

Despite its utility, the MoNet method suffers from some fundamental methodological problems. The manner of processing edges that tie for highest betweenness in the G-N algorithm is to randomly select just one of these edges for removal. This is an illogical procedure because these ties in fact represent structural equivalence of edges. This approach introduces random bias into the decomposition of the network and also results in different stochastic solutions to the modular decomposition of a network with each run. The first aim of this study was to correct this error by amending the G-N algorithm. In addition, the random biases created through the G-N algorithm decomposition of the network are perpetuated by the MoNet algorithm, leading to variable results across trials. Neither the G-N nor MoNet algorithm can account for structural equivalence of multiple edges. In an agglomerative algorithm such as MoNet, processing multiple edges can require the evaluation of adding to the network edges that connect any number of subgraphs. The MoNet algorithm can evaluate exactly one or two such subgraphs per iteration. Therefore, the second corrective measure in this study was to develop a new algorithm that can process iterations involving n number of subgraphs and structurally equivalent edges while preserving the intuitive essence of the original MoNet method. Finally, a feature of the MoNet algorithm leads to exclusion of a large portion of a network from the identified modules, preventing the algorithm from maximizing the size of all modules and limiting the number of functional hypotheses that can be generated. To address these faults, we have created an improved deterministic version of the G-N algorithm (dG-N), and a new module-assembly algorithm, Deterministic Modularization of Networks (dMoNet), that together compose a deterministic and more intuitive method for identifying modules.

A recent study comparing the performance of several graph modularization algorithms on a protein interaction network [5] found that Markov Clustering (MCL) [7], [27] outperformed Restricted Neighborhood Search Clustering, Super Paramagnetic Clustering and Molecular Complex Detection in terms of general performance, showing the most accurate clustering of protein complexes and the most robustness to random graph alterations that simulate data noise. Comparison of dMoNet to MCL results was performed to evaluate the utility of dMoNet for identifying functional modules in biological networks in the context of current leading modularization algorithms. The evaluation methodology implemented extends the use of GO annotation enrichment within modules such that a clearer comparison can be drawn between sets of modules derived from the same data set using different modularization methods.

II. Materials and Methods

A. Module definitions

In this study we employed the general module and simple module definitions proposed with the original MoNet method [17], with a simple adaptation of the simple module definition to accommodate the dG-N algorithm.

Definition 1: A module in general is a graph or subgraph that contains more internal than external edges, i.e. the ratio of internal to external edges is > 1.

Definition 2: A simple module is a module that if separated by removing edge sets in order of removal by the dG-N algorithm can generate at most one module.

The criterion in Definition 2 preserves the module property that the edge betweenness for edges included in modules is lower than for edges between modules. In addition to these module definitions, we used the Weak module definition [20] for the purpose of comparative evaluation of results.

Definition 3: A Weak module is a subgraph in which the sum of edges connecting a node to the rest of the subgraph across all subgraph nodes is greater than the sum of edges connecting a node to nodes not belonging to the same subgraph across all subgraph nodes.

B. Algorithms

dG-N algorithm: To address the problem of ties arising through the original G-N algorithm, we propose a simple amendment to the algorithm. The algorithm proceeds normally unless multiple edges tie for highest edge betweenness in a single iteration. For these cases, instead of randomly selecting a single edge from this set for removal from the network, the entire set of edges is removed as a unit slating them for simultaneous processing by the agglomerative phase of the dMoNet method. This amendment eliminates the random bias introduced by the original G-N algorithm, thereby eliminating the potential for variable modules across trials. The following is a summary of the dG-N algorithm.

1) A Breadth First Search (BFS) is performed to assign a distance and number of shortest paths to every node with respect to a starting node.
2) The Betweenness contributions of each edge going up the BFS tree are calculated.
3) Repeat Steps 1 and 2 until each network node has been used as the starting node.
4) Sum the betweenness contributions for each edge with respect to each starting node to obtain the betweenness for all edges in the network.
5) Identify the edge(s) with the highest betweenness in the network, and remove this edge set from the network.
6) Repeat Steps 1-5 until no edges remain in the network. Obtain the order of edge set deletion.

The algorithmic complexity of dG-N, like G-N, is $O(M^2N)$, where M equals the number of network edges, and N equals the number of network nodes.

dMoNet algorithm: The original MoNet algorithm was not designed to process structurally equivalent edges and thus perpetuates any random bias introduced by the G-N algorithm leading to varied results. dMoNet extends MoNet's algorithmic foundation, subgraph merging based on the module definition, to scenarios involving multiple highest betweenness edges, generated by the dG-N algorithm, and more than two subgraphs. In addition, dMoNet maximizes the size of simple modules identified. This results in all nodes of the network being placed into modules. This is accomplished by preserving the ability of any subgraph to merge with other nonmodule subgraphs despite prior attempts to merge modules to other modules. The following is a summary of the dMoNet algorithm.

1) A list of edge sets is created in the reverse order of edge set deletion by the dG-N algorithm.
2) The dMoNet algorithm is initialized by setting each node as a singleton subgraph with no edges.
3) An edge set is removed from the top of the list of edge sets.
4) Edges in the edge set are placed into edge-groups based on the subgraph pairs they connect.
5) Each edge-group is evaluated for addition to the graph based on the following criteria.
    a) The edge-group connects nodes within the same subgraph, or
    b) The edge-group connects nodes in two separate subgraphs, and
        i) Both subgraphs are nonmodules, or
        ii) The subgraphs are a module and a nonmodule. There is no indirect merging of two or more modules defined in previous iterations through a combination of this edge-group and any other edges in this edge set, and the merging of these subgraphs would still yield a module.
6) All edge-groups that satisfied the addition criteria are permanently added to the graph.
7) All edge-groups that failed to satisfy the addition criteria are not added to the graph and are henceforth disregarded.
8) Repeat steps 3-7 until no edge sets are left in the edge set removal list.

Steps 1-3 are performed to prioritize the addition of edges starting with the edges most likely to lie inside of modules. Edges grouped in step 4 are structurally equivalent with respect to the likelihood of being included in a module; therefore each edge-group can be considered as a unit, to be added or not added to the graph in total. Criterion 5a permits the addition of edges that contribute to the modularity of a subgraph without risking the undesired merging of modules. Criterion 5b permits the construction of modules by merging subgraphs without merging modules, maximizing the size of simple modules defined while preserving the modular state of previously defined modules. Addition of edges in step 6 after all of the evaluations have been performed prevents any arbitrary bias for addition of edges, evaluating the addition criteria only for the initial state of the graph for this iteration. In step 7, edges which were not found to contribute to the modularity of the relevant subgraphs in a given iteration are not added.

The algorithmic complexity of dMoNet is O(M), like MoNet, where M equals the number of network edges.

C. Interaction network data

Two network data sets were selected for this analysis. The *S. cerevisiae* core protein interaction network downloaded from the Database of Interacting Proteins, version ScereCR20041003 [29], was generated by filtering high-throughput protein interaction data using the Expression Profile Reliability Index and the Paralogous Verification Method [6]. The largest component of this filtered interaction network consists of a 2440 interconnected proteins and 6241 interactions.

The Homo sapiens protein interaction network was downloaded from the BioGRID database version 2.0.27 [25], where it was generated from literature curation of protein interaction data. This data was filtered, isolating only direct and physical interactions detected strictly between human proteins, and all loops and duplicate edges were removed. The largest component of the resulting network includes 6656 proteins and 19022 interactions.

D. Implementation

The implementation of the dG-N algorithm was adapted from a C program capable of running the original G-N in parallel [31] and run on the Lonestar supercomputer cluster at the Texas Advanced Computing Center (TACC) [26].

The dMoNet algorithm was implemented in a JAVA program configured to run serially. Because of the lower order algorithmic complexity of dMoNet in comparison to dG-N, this was a more efficient choice than parallelizing the dMoNet code.

E. Module evaluation

Although the modules that result from the MoNet and dMoNet algorithms are defined based on network topology, the ultimate goal in subdividing complex biological networks into modules is that the modules should reflect underlying biological processes. Indeed, in a comparison of results from different module definition trials or methods, the quality of the methods should be judged based on their ability to define modules that reflect these functional processes. In this study we utilize an evaluation method based on the Cluster Assignment for Biological Inference (CLASSIFI) algorithm [15].

CLASSIFI uses Gene Ontology annotation [1] to identify associations between gene or protein clusters and biological processes, cellular components, and molecular functions based on the probability of genes or proteins with shared GO term annotation being grouped together by chance assuming a hypergeometric distribution. CLASSIFI captures all GO terms from all three categories for all gene or protein members as well as the parent terms of these terms in the GO hierarchy and calculates co-clustering p-values for every term in every module. The lower the p-value of a GO term, the more statistically significant the enrichment of the GO term is in the module. Different module sets are generated following independent runs of the original MoNet algorithm or following runs of different modularization methods. Each module within a module set is composed of a list of protein members. CLASSIFI is run on each full module set, including the nonmodules from the trial to obtain proper CLASSIFI p-values. For the purpose of evaluating modules in this study based on association with biological processes alone, the CLASSIFI output was filtered to remove all results pertaining to cellular components and molecular functions.

The evaluation method compares the proportions of module sets and module members that are biologically significant in that they correspond to biological processes. This method considers only the lowest biological process GO term CLASSIFI p-value for each module. The proportion of modules and the proportion of nodes included in modules are calculated over a wide range of cutoffs restricting the CLASSIFI p-values, ranging from a very permissive to a very restrictive cutoff. For a given cutoff, the greater the proportion of modules or nodes included in modules, the higher the quality of the module set. The proportion of modules with significant CLASSIFI p-values is a measure of how meaningful the functional groupings represented in the modules are, and the proportion of nodes included in modules with significant CLASSIFI p-values is a measure of the ability to draw meaningful inferences about the function of individual network components from the modules.

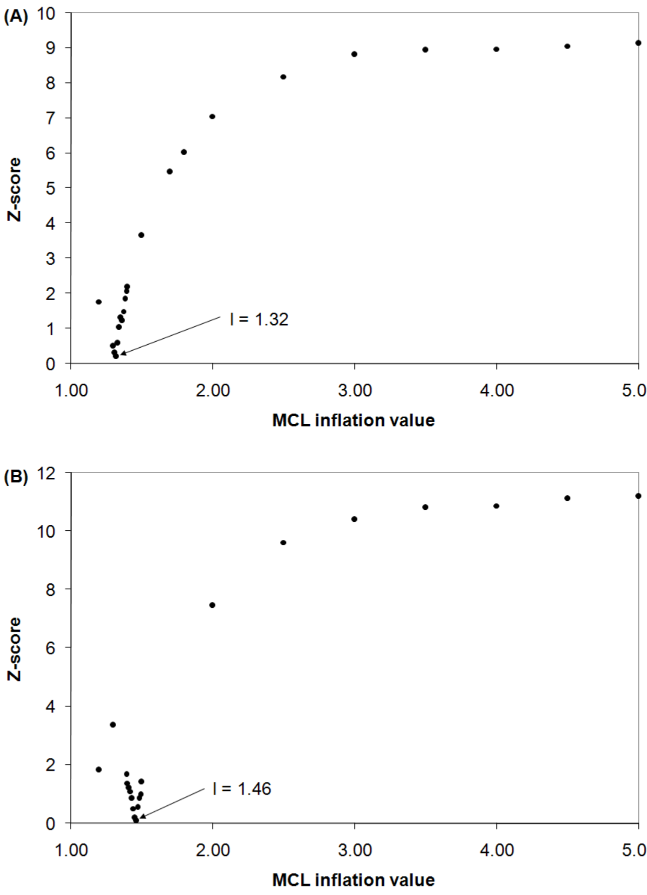

Fig. 1. Wilcoxon tests to determine MCL inflation parameter value. The dot plots represent the results of the Wilcoxon rank sum tests between distributions of dMoNet subgraph sizes and MCL cluster sizes over a series of MCL trials in which the inflation parameter was varied. The arrows point to the lowest Z-score results, representing the MCL trials that most closely correspond in size distribution to the dMoNet result for the yeast (A) and human (B) networks.

F. Framing the comparison to MCL results

The MCL algorithm [7], [27] imposes a flow simulation method on a graph, calculating successive powers of the adjacency matrix. An inflation parameter I enhances the contrast between subgraphs with strong and weak flow. The inflation parameter can be varied between 1.2 and 5.0. As the flow simulation converges, the graph is partitioned into clusters, preserving subgraphs with high flow that are separated by edges with no flow, i.e. have been removed from the graph. A key weakness of this method is that it uses no module definition and makes no assumptions about which resulting clusters are more likely to be functional modules than others.

It has been suggested that modules in biological networks follow a hierarchical organizational scheme [13], [21]. However, most modularization algorithms, including dMoNet, are concerned primarily with identifying functional modules that lie at a low level in this hierarchy for the purpose of generating immediately useful and more easily testable hypotheses.

dMoNet is nonparametric in that the single parameter, the modularity threshold, is set to 1.0 as a default, which reflects the intuitive module definition. Thus only one result can be obtained for any given run of dMoNet on a graph. MCL, however, requires that the inflation parameter be set by the user, influencing the number and sizes of clusters obtained. In order to perform a fair comparison to dMoNet, a single MCL trial must be selected, one that most closely approximates the same hierarchical level of modules as dMoNet. Assuming that both algorithms perform reasonably well when partitioning the network into modules, it should be possible to distinguish the MCL trial that most closely corresponds to the same module hierarchical level as dMoNet by analysis of the distribution of resulting cluster sizes.

For both the yeast and human networks, a number of trials were performed to determine the ideal inflation value to choose which would allow for the most appropriate comparison of MCL to dMoNet results. First, trials were run widely varying the inflation value across the suggested range from 1.2 to 5.0. Lists of MCL cluster sizes were compiled from each result, removing sizes of fewer than three nodes so as not to allow very small clusters that are not likely to represent actual modules to bias the distribution of sizes. A Wilcoxon rank sum test [28] was used to compare each list of MCL cluster sizes to a similar list of subgraphs obtained using dMoNet (Fig. 1). The two trials with the lowest rank sum Z-scores were chosen to define the inflation value range limits for the next set of trials. This procedure was repeated until the limiting range of inflation values was 0.1. Then trials were run using all inflation values within this range with a precision of 0.01. From these trials, the ideal inflation value for comparison was chosen by again performing rank sum tests and identifying the single MCL trial with the cluster size distribution most similar to that of the dMoNet result. For the yeast and human protein interaction networks, the ideal inflation values determined were 1.32 (Fig. 1A) and 1.46 (Fig. 1B), respectively.

III. Results

A. Nondeterministic MoNet results

Three problems were identified in the original MoNet method for identifying functional modules from networks: the potential for the introduction of random bias in the decomposition results, observed variability in results across multiple trials leading to a lack of reproducibility in results for many networks, and failure to maximize the size of identified modules. The former two derive directly from how the G-N algorithm handles ties in highest betweenness. The third stems from a subgraph-merging criterion based on the G-N removal order that is too restrictive.

The original G-N algorithm permits the introduction of nonstructural random bias into the network decomposition. In the case of the cyclic graph in Fig. 2A, all edges have identical betweenness values in the first iteration of the G-N algorithm, and thus one of the edges is chosen at random for removal. Subsequent iterations also exhibit ties, leading to 384 possible edge removal lists that could be generated by the original G-N algorithm. When these lists are used for re-assembling the network and evaluation of modularity by MoNet, four possible results could be obtained. In the case of a slightly more complex graph (Fig. 2B), 103680 possible edge removal lists could be generated, leading to any of three distinct results by MoNet. The dMoNet result for each of these small graphs is deterministic both with respect to the edge removal list and the resulting modules.

B. Effects of MoNet and dMoNet on network disassembly

Of the 6241 iterations required to complete the G-N decomposition of the large component of the yeast core protein interaction network, 2576 exhibit ties in highest betweenness value. The question arises of whether the tying edges are actually removed in order during consecutive subsequent iterations in the G-N algorithm. If this were the case, the random selection of edges for removal might have minimal or no impact on the final result. Although addressing this question is technically challenging, a simple statistic suggests that this is frequently not the case. If the random removal of a tying edge had no impact on the next iteration in the G-N algorithm, the number of tying edges should decrease by exactly one after each subsequent iteration. The consecutive removal interruptions are cases where there is a decrease other than one in the number of tying edges between a tie and the subsequent iteration. This could indicate that an edge other than one of those that tied during the first iteration is removed during the second iteration or that some of the edges that tied during the first iteration no longer tie after the removal of the edge in the first iteration. Either possibility leads to a random bias in the structure of the subnetwork as a direct result of randomly selecting an edge for removal from the set of tying edges. There are 1460 such occurrences out of the 2576 ties for the G-N run of the yeast network indicating that random bias would be expected to have a significant impact.

The change included in the dG-N algorithm results in still other quantifiable effects on the network disassembly when compared with the G-N algorithm. The number of iterations required to disassemble the large component of the yeast protein interaction network is reduced from 6241 to 3301 due to the simultaneous removal of tying edges. This also results in a substantial decrease in the number of iterations with tying edges from 2576 to 283.

In order to identify differences between the yeast network results of the original MoNet and dMoNet, the lists of edges retained in modules at the end of separate runs were compared. The initial large component of the yeast protein-protein interaction network contained 6241 interaction edges. The separation of this network into a series of modules with dMoNet retained 4728 ($\sim$76%) of these edges. In 13 independent runs of the original MoNet, between 3403 ($\sim$54%) and 3423 ($\sim$55%) edges were retained. dMoNet gives identical results from multiple runs, whereas the original MoNet could potentially give different results with each run. A comparison of the overlap in retained edge lists indicated that between $\sim$99.53% and $\sim$100% of the edges were identical between the 13 separate runs of the original MoNet. The overlap in retained edge lists between dMoNet and the 13 runs of MoNet was consistently much smaller, ranging between 83.68% and 83.97%. One feature that contributes to this difference in percent overlap relates to the observation that fewer edges are consistently retained using MoNet than dMoNet. This observation and the observation that across all comparisons of dMoNet to MoNet, the dMoNet modules consistently contain all but a single edge included in the MoNet result provide evidence that the dMoNet modules represent expansions of MoNet modules to include more of the network. Thus dMoNet performs better at maximizing the size of modules and covers the whole of the original network.

C. Module comparisons and evaluations

For evaluation and comparison purposes, an average MoNet result was created from the 13 yeast network trials including module edges present in a majority of these results; this result will henceforth be the only MoNet result discussed. MoNet identified 99 simple modules with a total of 1687 nodes and 3420 edges, and dMoNet identified 81 simple modules containing a total of 2440 nodes and 4728 edges from the large component of the yeast network. Of the dMoNet modules, 67 have exactly equivalent corresponding MoNet modules. The remaining 14 dMoNet modules correspond to multiple MoNet modules and also contain a large number of nodes not included

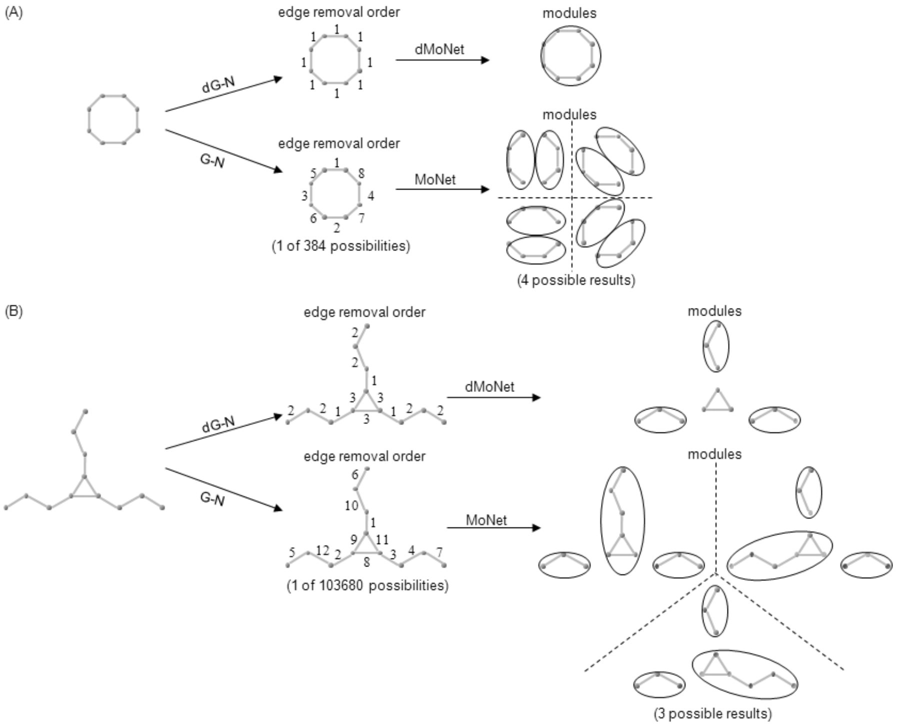

Fig. 2. Deterministic module identification using dMoNet. The modular decomposition of a cyclic graph (A) and a linear-ring hybrid graph (B) by dMoNet and MoNet. Possible edge removal orders based on the G-N or dG-N algorithm are indicated. For the dG-N algorithm a single removal order is possible in all cases; for the original G-N algorithm, 384 and 103680 removal orders are possible, respectively. This results in single solutions to the modular decomposition problems using dMoNet but multiple possible solutions using MoNet that are quite distinct.

in MoNet modules.

MoNet identified 157 simple modules from the large component of the human protein interaction network, including 3187 total nodes and 6967 total edges. The dMoNet decomposition of the human network identified 123 modules containing all 6656 network nodes and 15516 edges from the large component of the human network. These simple observations obviate a difference between the dMoNet and MoNet results.

The performance of dMoNet on both the yeast and human networks was evaluated further by comparison with usage of the Weak module definition of Radicchi [20], the original MoNet (a single trial was run for the human network), and the MCL algorithm [5], [7], [27] (using ideal inflation values as described in the Methods) using GO annotation co-clustering as assessed by the CLASSIFI algorithm (see Methods). CLASSIFI analysis of the yeast dMoNet modules showed that 69 out of the 81 modules showed significant co-clustering of proteins that share biological process GO annotation beyond what would be expected by chance, using a p-value cutoff of 2.05E-05, which is the threshold corresponding to a 5% chance of committing a Type I error based on the Bonferroni correction [4]. Similar analysis of the human dMoNet modules showed that 81 out of the 123 modules showed significant co-clustering of proteins sharing biological process GO annotation, using a Bonferroni threshold of 7.51E-06.

The modules resulting from the different decompositions were compared in terms of the proportions of modules with significant biological process term co-clustering (Fig. 3A and Fig. 4A). In these analyses, the proportion of modules with co-clustering p-values was determined at various p-value cutoffs. For example, ∼33% (0.33 proportion) of yeast Weak modules have a GO term p-value below 1.00E-10. In this analysis, dMoNet shows a higher proportion of modules over most significance cutoffs for the yeast network than the alternative methods, although as the cutoff becomes stricter, the proportion becomes similar to that of MCL with $I = 1.32$ and MoNet. In contrast, the proportion of Weak modules with significant GO term co-clustering is always lower than the proportion for dMoNet regardless of the p-value cutoff used. For the human network, the proportion of significant dMoNet modules clearly exceeds that of all other methods across the range of cutoffs. Notably, although MCL performed fairly well on the yeast network, in this evaluation of the human network it performed worse than all of the other methods.

Another way to assess these co-clustering results is to measure the proportion of nodes included in modules annotated with GO terms that fall below different p-value cutoffs (Fig. 3B and Fig. 4B). For both networks, in comparison with Weak modules, dMoNet shows a higher proportion of nodes included in modules with significant GO term co-clustering across the entire range of significance cutoffs. Across most of the range of cutoffs, the proportion of significant yeast

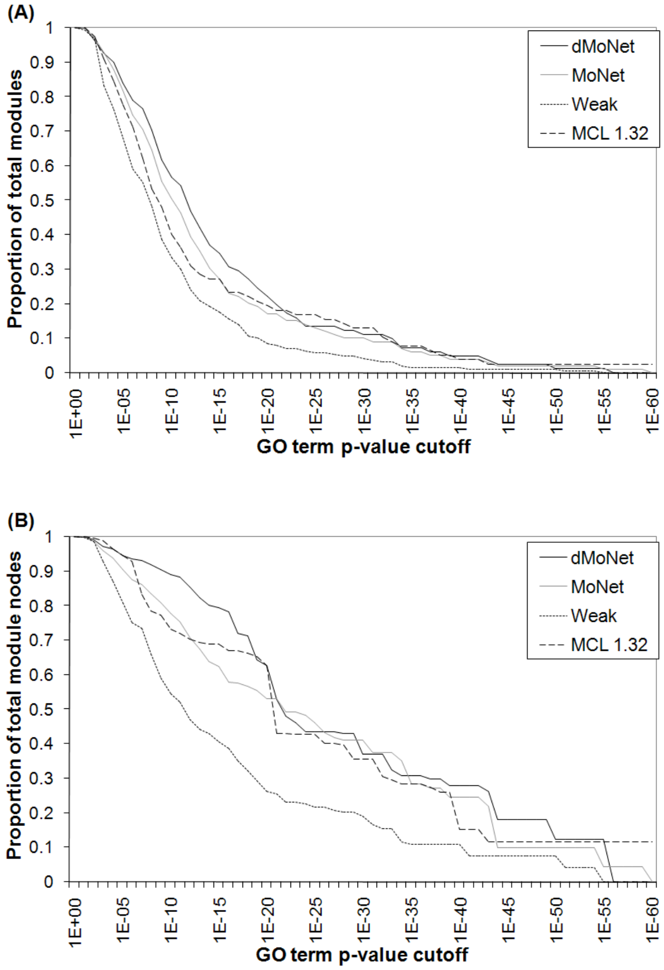

Fig. 3. Proportion of yeast modules and nodes included in modules with significant co-clustering of GO terms. The comparison of the proportions of modules (A) and module nodes (B) within each set of results for the yeast network that are annotated with at least one biological process GO term with an associated co-clustering p-value lower than the given cutoff is shown.

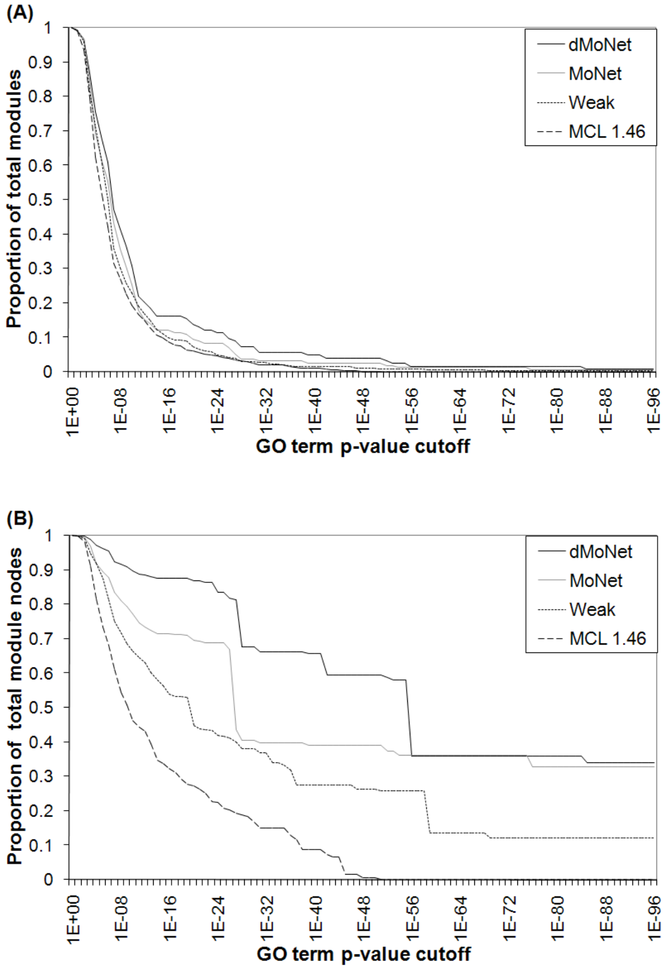

Fig. 4. Proportion of human modules and nodes included in modules with significant co-clustering of GO terms. The comparison of the proportions of modules (A) and module nodes (B) within each set of results for the human network that are annotated with at least one biological process GO term with an associated co-clustering p-value lower than the given cutoff is shown.

dMoNet module nodes exceeds that of MCL and MoNet by a fairly wide margin, although there exist a few exceptions for which the MCL or MoNet proportion exceeds that of dMoNet to a much lesser extent. The major exceptions are at the most restrictive cutoff for which both MCL and MoNet retain a single significant yeast module with 285 nodes and 76 nodes, respectively. For the human network, the proportions of significant dMoNet, MoNet, and Weak module nodes far exceed that of MCL across the entire range of cutoffs. An important nuance of this result is that the dMoNet human module 001 with the lowest p-value (5.67E-192) is very large (2271 nodes). However, if one were to remove the contribution that this module gives to the proportion of significant module nodes, the dMoNet proportion would still approximate or exceed that of MCL over most of the range of cutoffs. Taken together, these data indicate that module identification using dMoNet overall produces modules that more closely resemble biological processes as defined in the scientific literature.

IV. Discussion

A. Applications of dMoNet

Several potential applications of dMoNet exist in studying protein or gene function. dMoNet, in combination with CLASSIFI, can be used to automatically infer the functional grouping of proteins at a genomic scale based on high-throughput experimentation. dMoNet can also be applied to predict novel functional classification for proteins based on their membership in modules with high scoring biological process classification. To illustrate this type of application, two parallels between dMoNet modules from the yeast network and characterized protein complexes were closely examined (Fig. 5). Data for these complexes was obtained from the BIND [2] and MIPS/CYGD [11], [18] databases. Protein complex memberships were determined via tandem affinity

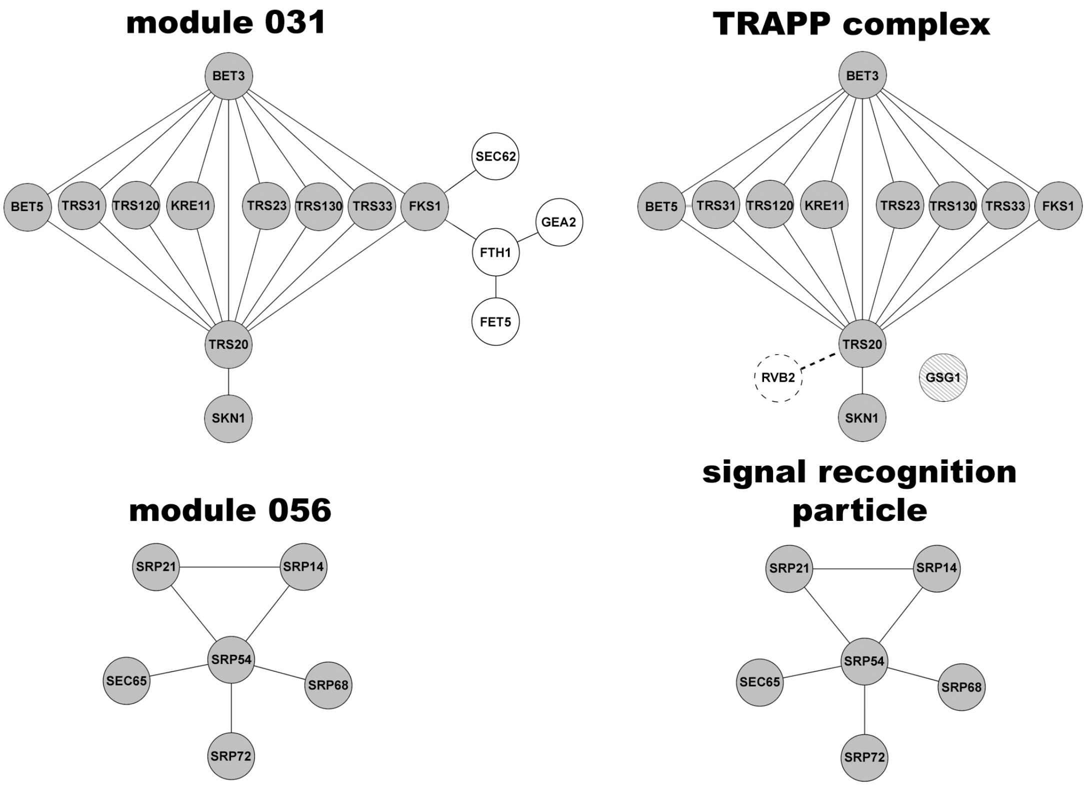

Fig. 5. Correspondence between yeast dMoNet modules and known protein complexes. The gray, hatched and dotted nodes nodes represent members of the complex. The dotted node indicates a protein that was not assigned to the corresponding module, and the hatched node indicates a protein that was not present in the yeast network data. White nodes represent proteins assigned to the module that are not known to participate in the corresponding complex. Dotted edges indicate interactions not retained through the decomposition. A disconnected node represents a lack of complete interaction data to show how the protein physically links to the rest of the complex.

purification [9], and their interactions via yeast two-hybrid experiments [29].

In the simplest case, the additional classification evidence provided by dMoNet could confer another measure of confidence in previously characterized functional relationships. For example, the dMoNet yeast module 056 is identical both by protein membership and known interaction structure to the signal recognition particle (SRP), which aids in protein targeting to the ER membrane. This exact correspondence to the known SRP provides evidence affirming the characterized structure.

In other cases, dMoNet results that conflict with existing experimentally determined functional grouping could suggest experimental false positive or false negative results. dMoNet yeast module 031 closely corresponds to the TRAPP complex, involved in ER to Golgi membrane traffic. The corresponding module excludes the RVB2 protein but includes additional proteins SEC62, FTH1, FET5, and GEA2. This may suggest the incomplete or mischaracterization of the TRAPP complex, or it may identify more key proteins involved in ER to Golgi membrane traffic.

When insufficient annotation exists concerning a given group of proteins, the GO term "biological_process_unknown" is identified by CLASSIFI as the lowest co-clustering p-value GO term, such as in dMoNet human module 003. This module contains 893 proteins, 101 of which are annotated as "biological_process_unknown". This dMoNet module may very well take part in a previously uncharacterized biological process, and the grouping of proteins in this module could aid in selecting targets for experimental studies to identify this putative novel process. As the GO annotation improves and expands, this evaluation method will make a better assessment of the quality of functional modules identified by different modularization methods.

Another interesting finding from the dMoNet decomposition of the human network is the aforementioned human module 001 with GO biological process annotation "nucleobase, nucleoside, nucleotide and nucleic acid metabolic process." This module has tumor protein p53 (TP53) as its hub protein. Within this module, TP53 is connected to 122 other proteins. This module is of particular interest because it happens to be the largest of the human dMoNet modules, with 2271 nodes, and also has the lowest GO term co-clustering p-value (5.67E-192), successfully clustering 1107 of the 1745 proteins in this data set annotated with this term. TP53 is a tumor suppressor gene that plays an important role in the regulation of the cellular response to DNA damage. TP53 is a DNA-binding protein containing DNA-binding, oligomerization and

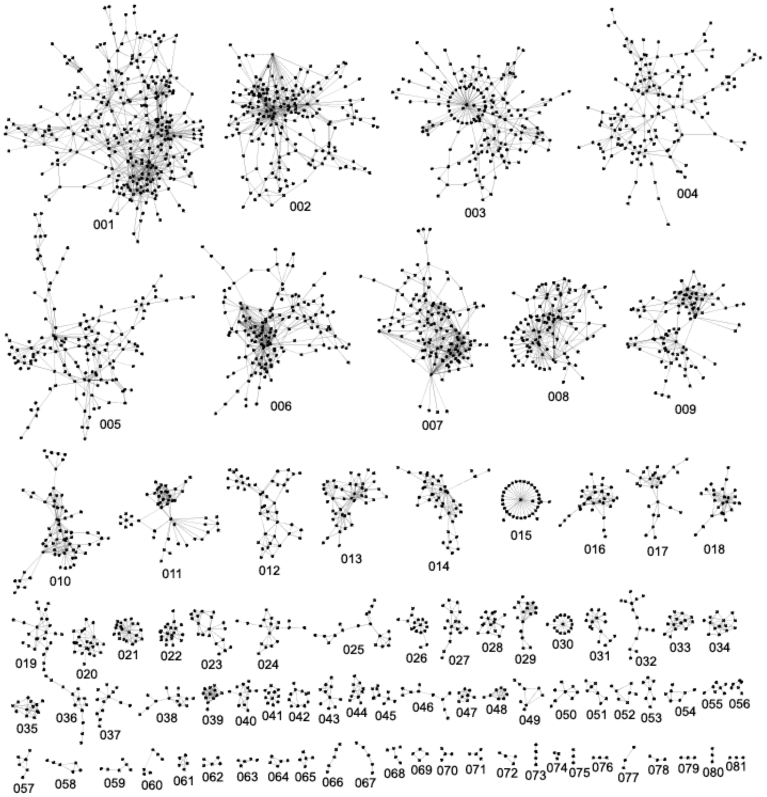

Fig. 6. Topologies of yeast dMoNet modules. The yeast dMoNet modules are numbered by descending size. These modules exhibit a variety of observed topologies including linear, star, ring, mesh, and complex topologies.

transcription activation domains. The members and structure of this module could be of great interest in cancer research since p53 is the most commonly mutated gene in spontaneous human cancers. The fact that p53 is also the most highly connected node in the largest human protein interaction module defined by dMoNet is intriguing. The corresponding MCL module containing TP53 is only 128 nodes in size, and its lowest co-clustering p-value (2.14E-23) GO term is also “nucleobase, nucleoside, nucleotide and nucleic acid metabolic process”, correctly clustering only 87 out of the 1745 proteins in the data set annotated with this term.

B. Significance of module topology

A potentially significant advantage that the dMoNet method has over certain other network structure-based module identification methods is that it allows the identification of modules with a variety of different topologies as evidenced by the yeast modules diagram (Fig. 6). The module images in Fig. 6 were produced using the Cytoscape network visualization tool [23]. These include not only modules that are highly connected dense modules with a mesh-like structure (e.g. yeast module 021), but also modules with star (e.g. yeast module 030), ring (e.g. yeast module 069) and linear (e.g. yeast module 066) topologies. In addition modules with more complex topologies are frequently observed. The measure of modularity in this study is defined by a comparison of internal to external links, rather than simply considering internal links. In this manner, the method relies on information in addition to the internal topology of subgraphs. It is for this reason that a wider variety of module topologies can be obtained by dMoNet than by methods based on detecting only densely connected subgraphs, which have a propensity for identifying modules that predominantly exhibit highly-connected mesh topologies and clique structures [14], [30], [24], [16]. The capability of detecting modules without an absolute bias towards dense topologies is critical because real biological systems exploit a variety of interaction motifs, which correspond to varied topologies. These topological motifs may represent certain underlying themes in biological systems.

The dMoNet modules contain examples of highly significant association with biological processes for modules exhibiting simple topologies and also for modules with more complex topologies. The dMoNet yeast module 067 is a 5-node linear module containing 5 out of 22 of the proteins in the network annotated with the term “double-strand break repair via non-homologous end joining” (p-value = 3.67E-11). The 4-node simple star yeast module 071 contains 3 out of 7 proteins annotated with the term “DNA topological change” (p-value = 5.78E-08). The 6-node dMoNet yeast module 062 has a topology dominated by two 3-member rings and contains 4 out of the 19 proteins annotated with the term “protein import into mitochondrion” (p-value = 1.31E-08). The 12-node dMoNet yeast module 039 has a topology dominated by a highly connected mesh containing 11 out of the 12 proteins annotated with the term “cyclin catabolic process” (p-value = 3.22E-28). The topologies of these examples show high contrast with one another, yet all of them show highly significant association with biological processes as defined by GO term annotation.

Acknowledgement

We would like to thank Q. Yang and S. Lonardi for providing the original parallelized G-N code, J. Cai for providing CLASSIFI support, M. Kong for statistics advice, and M. McGee for critical review of the manuscript. We would also like to acknowledge the Texas Advanced Computing Center (TACC) at the University of Texas at Austin for providing HPC resources that have contributed to the research results reported within this paper. This research was supported by the National Institutes of Health N01-AI40076 and N01-AI40041. F. L. is supported by NSF EPSCoT grant EPS-0447660.

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 et al. Gene Ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nature Genet*, vol. 25, pp. 25-9, 2000. Available: http://www.geneontology.org

[2] G.D. Bader, D. Betel, and C.W. Hogue. BIND: the Biomolecular Interaction Network Database. *Nucleic Acids Res*, vol. 31, no. 1, pp. 248-50, 2003. Available: http://bond.unleashedinformatics.com

[3] A.L. Barabasi and Z.N. Oltvai. Network biology: Understanding the cells functional organization. *Nat Rev Genet*, vol. 5, pp. 101-13, 2004.

[4] Bonferroni Correction. Available: http://mathworld.wolfram.com

[5] S. Brohee and J. Van Helden. Evaluation of clustering algorithms for protein-protein interaction networks. *BMC Bioinformatics*, vol. 7, pp. 488, 2006.

[6] C.M. Deane, L. Salwinski, I. Xenarios, and D. Eisenberg. Protein interactions: Two methods for assessment of the reliability of high throughput observations. *Mol Cell Proteomics*, vol. 1, no. 5, pp. 349-56, 2002.

[7] A.J. Enright, S. Van Dongen, and C.A. Ouzounis. An efficient algorithm for large-scale detection of protein families. *Nucleic Acids Res*, vol. 30, no. 7, pp. 1575-1584, 2002.

[8] L.C. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, vol. 40, pp. 35-41, 1977.

[9] A.C. Gavin, M. Bosche, R. Krause, P. Grandi, M. Marzioch, A. Bauer, J. Schultz, J.M. Rick, A.M. Michon, C.M. Cruciat et al. Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature*, vol. 415, pp. 141-7, 2002. Available: http://yeast.cellzome.com/index.php

[10] M. Girvan and M.E.J. Newman. Community structure in social and biological networks. *Proc Natl Acad Sci U S A*, vol. 99, no. 12, pp. 7821-6, 2002.

[11] 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 et al. CYGD: the Comprehensive Yeast Genome Database. *Nucleic Acids Res*, vol. 33 (Database issue), pp. D364-8, 2005.

[12] L.H. Hartwell, J.J. Hopfield, S. Leibler, and A.W. Murray. From molecular to modular cell biology. *Nature*, vol. Suppl 402, pp. C47-52, 1999.

[13] Z. Hu, J. Mellor, J. Wu, M. Kanehisa, J.M. Stuart, and C. DeLisi. Towards zoomable multidimensional maps of the cell. *Nature Biotechnol*, vol. 25, no. 5, pp. 547-554, 2007.

[14] H. Hu, X. Yan, Y. Huang, J. Han, and X.J. Zhou. Mining coherent dense subgraphs across massive biological networks for functional discovery. *Bioinformatics*, vol. Suppl 21, pp. i213-21, 2005.

[15] J.A. Lee, R.S. Sinkovits, D. Mock, E.L. Rab, J. Cai, P. Yang, B. Saunders, R.C. Hsueh, S. Choi, S. Subramaniam, and R.H. Scheuermann. Components of the antigen processing and presentation pathway revealed by gene expression microarray analysis following B cell antigen receptor (BCR) stimulation. *BMC Bioinformatics*, vol. 7, pp. 237, 2006. Available: http://pathcuric1.swmed.edu/pathdb/CLASSIFI.html

[16] W. Li, Y. Liu, H.C. Huang, Y. Peng, Y. Lin, W.K. Ng, and K.L. Ong. Dynamical systems for discovering protein complexes and functional modules from biological networks. *IEEE/ACM Trans Comput Biol Bioinform*, Apr-Jun 2007.

[17] F. Luo, Y. Yang, C.F. Chen, R.L. Chang, J.Zhou, and R.H. Scheuermann. Modular organization of protein interaction networks. *Bioinformatics*, vol. 23, no. 2, pp. 207-14, 2007.

[18] H.W. Mewes, D. Frishman, U. Gldener, G. Mannhaupt, K. Mayer, M. Mokrejs, B. Morgenstern, M. Mnsterktter, S. Rudd, and B. Weil. MIPS: a database for genomes and protein sequences. *Nucleic Acids Res*, vol. 30, no. 1, pp. 31-4, 2002. Available: http://mips.gsf.de/genre/proj/yeast

[19] M.E.J. Newman and M. Girvan. Finding and evaluating community structures in networks. *Physical Review E*, vol. 69, pp. 026113, 2004.

[20] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi. Defining and identifying communities in networks. *Proc Natl Acad Sci U S A*, vol. 101, no. 9, pp. 2658-63, 2004.

[21] E. Ravasz, A.L. Somera, D.A. Mongru, Z.N. Oltvai, and A.L. Barabsi. Hierarchical organization of modularity in metabolic networks. *Science*, vol. 297, pp. 1551-5, 2002.

[22] A.W. Rives and T. Galitski. Modular organization of cellular networks. *Proc Natl Acad Sci U S A*, vol. 100, no. 3, pp. 1128-33, 2003.

[23] 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*, vol. 13, pp. 2498-504, 2003.

[24] V. Spirin and L.A. Mirny. Protein complexes and functional modules in molecular networks. *Proc Natl Acad Sci U S A*, vol. 100, no. 21, pp. 12123-8, 2003.

[25] C. Stark, B.J. Breitkreutz, T. Reguly, L. Boucher, A. Breitkreutz, and M. Tyers. BioGRID: A General Repository for Interaction Datasets. *Nucleic Acids Res*, vol. 34, pp. D535-9, 2006. Available: http://www.thebiogrid.org

[26] Texas Advanced Computing Center, The University of Texas at Austin. Available: http://www.tacc.utexas.edu

[27] S. Van Dongen. Graph Clustering by Flow Simulation. Ph.D. thesis, University of Utrecht, 2000. Available: http://www.library.uu.nl/digiarchief/dip/diss/1895620/inhoud.htm

[28] F. Wilcoxon. Individual comparisons by ranking methods. *Biometrics*, vol. 1, pp. 80-83, 1945.

[29] I. Xenarios, L. Salwnski, X.Q. Duan, P. Higney, S.M. Kim, and D. Eisenberg. DIP, the Database of Interacting Proteins: a research tool for studying cellular networks of protein interactions. *Nucleic Acids Res*, vol. 30, no. 1, pp. 303-5, 2002. Available: http://dip.doe-mbi.ucla.edu

[30] H. Xiong, X. He, C. Ding, Y. Zhang, V. Kumar, and S.R. Holbrook. Identification of functional modules in protein complexes via hyperclique pattern discovery. *Pac Symp Biocomput*, vol. 10, pp. 221-32, 2005.

[31] Q. Yang and S. Lonardi. A parallel edge-betweenness clustering tool for protein-protein interaction networks. *International Journal of Data Mining and Bioinformatics*, vol. 1, no. 3, pp. 241-7, 2007.