BIOINFORMATICS ORIGINAL PAPER

Vol. 25 no. 23 2009, pages 3143–3150
doi:10.1093/bioinformatics/btp551

Systems biology

How and when should interactome-derived clusters be used to predict functional modules and protein function?

Jimin Song and Mona Singh*

Department of Computer Science & Lewis-Sigler Institute for Integrative Genomics Princeton University, Princeton, NJ 08544, USA

Received on July 10, 2009; revised on August 28, 2009; accepted on September 14, 2009

Advance Access publication September 21, 2009

Associate Editor: Burkhard Rost

ABSTRACT

Motivation: Clustering of protein–protein interaction networks is one of the most common approaches for predicting functional modules, protein complexes and protein functions. But, how well does clustering perform at these tasks?

Results: We develop a general framework to assess how well computationally derived clusters in physical interactomes overlap functional modules derived via the Gene Ontology (GO). Using this framework, we evaluate six diverse network clustering algorithms using *Saccharomyces cerevisiae* and show that (i) the performances of these algorithms can differ substantially when run on the same network and (ii) their relative performances change depending upon the topological characteristics of the network under consideration. For the specific task of function prediction in *S.cerevisiae*, we demonstrate that, surprisingly, a simple non-clustering guilt-by-association approach outperforms widely used clustering-based approaches that annotate a protein with the overrepresented biological process and cellular component terms in its cluster; this is true over the range of clustering algorithms considered. Further analysis parameterizes performance based on the number of annotated proteins, and suggests when clustering approaches should be used for interactome functional analyses. Overall our results suggest a re-examination of when and how clustering approaches should be applied to physical interactomes, and establishes guidelines by which novel clustering approaches for biological networks should be justified and evaluated with respect to functional analysis.

Contact: msingh@cs.princeton.edu

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

1 INTRODUCTION

Proteome-scale physical interaction data have become available for a large number of organisms, including human and most model organisms. Global analyses of the resulting protein interaction networks provide new opportunities for uncovering cellular organization and revealing protein functions and pathways. Beyond the basic characterization of these interaction networks with respect to their topological features (e.g. Barabási and Oltvai, 2004), arguably the most widespread approach for analyzing biological networks is to cluster or partition them into subcomponents. Clustering of biological networks has revealed a modular organization (Hartwell *et al.*, 1999), with highly connected groups of proteins taking part in the same biological process (BP) or protein complex (Bader and Hogue, 2003; Pereira-Leal *et al.*, 2004; Rives and Galitski, 2003; Spirin and Mirny, 2003). Indeed, dozens of papers for analyzing protein interaction networks have focused on finding clusters within them and novel network clustering methods continue to be developed (e.g. Adamcsek *et al.*, 2006; Altaf-Ul-Amin *et al.*, 2006; Arnau *et al.*, 2005; Asthana *et al.*, 2004; Brun *et al.*, 2003; Chen and Yuan, 2006; Dunn *et al.*, 2005; Enright *et al.*, 2002; King *et al.*, 2004; Luo *et al.*, 2007; Navlakha *et al.*, 2009; Newman, 2006; Poyatos and Hurst, 2004; Radicchi *et al.*, 2004; Samanta and Liang, 2003; Schlitt *et al.*, 2003; von Mering *et al.*, 2003; Wang *et al.*, 2007).

Most frequently, computationally derived clusters within physical interaction networks are used to uncover protein complexes and functional modules, as well as to predict protein function. Typically, a cluster is associated with a known complex or function by determining whether the number of proteins known to be part of the complex or annotated with the function is enriched, as judged by the hypergeometric distribution. Within a cluster, enriched functions, perhaps also required to annotate a suitable fraction of member proteins, can then be transferred to other member proteins. While these types of analysis are commonplace in interactomics, how effective are they for the tasks at hand?

Here, we focus on the task of utilizing network-derived clusters to uncover functional modules and predict protein functions. Evaluating how well clusters correspond to functional modules is a challenging task. Central to this is that while functional modules are commonly defined as groups of proteins that work together to accomplish a BP, there is no widely accepted formal definition of a module; many have been proposed, though typically based on topological features of the network (e.g. Radicchi *et al.*, 2004). We utilize an external measure—the Gene Ontology (GO; Ashburner *et al.*, 2000)—to derive functional modules. That is, for a GO BP or cellular component (CC) functional term, the corresponding module contains all the proteins that are annotated with that term. Since GO relates functions in a hierarchical fashion, the next challenge for evaluating clusters is to deal with this hierarchy. At first glance, it may appear that functions can be chosen at a particular resolution in the hierarchy. For example, it is possible to utilize the high-level GO ‘slim’ functional terms, and then clusters

*To whom correspondence should be addressed.

can be evaluated in how well they recapitulate these terms, using sensitivity and positive predictive value measures, as introduced in an influential quantitative assessment of how well clustering approaches can uncover known protein complexes (Brohée and van Helden, 2006). However, for evaluating functional modules, this approach has the weakness that a clustering that finds many small tightly connected clusters corresponding to very specific BPs would be unfairly penalized.

Our main technical contribution is a series of measures that can be used to compare and evaluate network clustering algorithms with respect to how well they perform in uncovering known, potentially overlapping functional modules. We demonstrate the quality of our measures by using them on random networks, and on clusters derived from the annotations themselves (i.e. these two extremes represent the noisy versus ideal scenarios). With this evaluation framework in hand, in order to make general conclusions about the efficacy of network clustering-based approaches, we experiment with six available clustering algorithms on four different high-throughput derived *Saccharomyces cerevisiae* physical interaction networks. We find that clustering algorithms exhibit a wide range of performances in recapitulating functional modules, derived from either BP or CC GO terms, even when run on the same network, and that the relative performance of clustering algorithms varies depending on the network at hand. In particular, we find that topological features of the network should guide algorithm choice. Given the vast differences we find in how well clustering algorithms recapitulate functional modules, this is an important practical consideration. As a byproduct of our analysis, we can also make conclusions about individual clustering algorithms: overall, though there are some clustering approaches which clearly outperform others, there is no single network clustering approach that dominates the rest in all cases.

Since module finding in biological networks is often motivated by the task of function prediction, we also perform a comprehensive evaluation in this scenario. Surprisingly, we find that for *S.cerevisiae*, the common practice of annotating a protein with the overrepresented BP or CC terms in its cluster is less accurate than simple guilt-by-association approaches based on considering just the annotations of direct interaction partners. This is true regardless of which underlying clustering approach is used. Additionally, as annotations are removed from the network, the relative performance of clustering-based function prediction improves in comparison to the simple scheme that just considers the annotations of interacting proteins. This suggests that clustering-based methods are most useful in networks obtained for genomes with fewer protein annotations.

In addition to characterizing the utility of network-derived clusters in uncovering functional modules and predicting protein functions, a major contribution of our work is a framework that can be used in the future for evaluating how well a new clustering approach performs for these tasks. Importantly, our testing suggests that while clustering of networks is often motivated by the goal of predicting protein function, if new clustering approaches are evaluated with respect to function prediction, it is important to demonstrate how much, or in which circumstances, improvement is obtained over guilt-by-association approaches. Overall, we hope that our testing framework as well as our findings about the utility of interactome-based clustering will inform future methodological advances in clustering biological networks.

2 MATERIALS AND METHODS

2.1 Interaction and functional module datasets

We use *S.cerevisiae* protein interaction data from BioGRID (Stark *et al.*, 2006), release 2.0.20, and generate four different networks in order to analyze how the underlying characteristics of the networks affect the performance of clustering algorithms. The first network contains all *S.cerevisiae* genetic and physical interactions in BioGRID. The second network contains all physical interactions. The third network consists of high-throughput physical (HTP) interactions from large datasets, in case the small-scale experiments in BioGRID overlap the protein complexes used for evaluation, and the last network consists of physical interactions derived via three large-scale experiments utilizing the yeast two-hybrid (Y2H) technique (see Supplementary Material for details). For each network, we filter the data to remove proteins which interact with more than 50 other proteins. Furthermore, self-interactions are ignored. The resulting four networks have different topological properties (Supplementary Table S1), as judged by the average number of interactions per protein, and the average node clustering coefficient. While we utilize all of these networks in our analyses, in the main body of this article, we focus on the third and fourth networks, which we will refer to as the HTP network and the Y2H network. The HTP network has 4160 proteins with 11 928 interactions, and the Y2H network has 2828 proteins with 3170 interactions.

We derive our gold standard groups from MIPS complexes (Mewes *et al.*, 2004) and GO (Ashburner *et al.*, 2000). We utilize 220 *S.cerevisiae* protein complexes from MIPS; this is the same set as used in the study of (Brohée and van Helden, 2006). For each of the networks described above, we remove from consideration any complex that has two or fewer proteins in the network. This leaves between 107 and 133 protein complexes for each network. In GO, there are 1963 BPs and 551 CC terms. We remove GO annotations with evidence codes IEA, RCA and IPI. For each network, we remove BP and CC terms that annotate more than 100 proteins or fewer than three proteins. This leaves from 954 to 1090 BP terms and from 324 to 357 CC terms for each *S.cerevisiae* network. For the HTP network, 66% of the proteins are annotated with one of these BP terms, and 41% are annotated with one of these CC terms. For the Y2H network, these numbers are 70% and 45%, respectively. For each BP and CC term under consideration, we define a functional module consisting of the proteins in the organism annotated with it. This gives us sets of potentially nested functional modules that range in specificity and size.

2.2 Clustering algorithms

We consider six network clustering algorithms: *NetworkBlast* (Sharan *et al.*, 2005), *CFinder* (Adamcsek *et al.*, 2006), *MCL* (Enright *et al.*, 2002), *DPClus* (Altaf-Ul-Amin *et al.*, 2006), *Mcode* (Bader and Hogue, 2003) and a spectral approach based on modularity (Newman, 2006), which we refer to as *SpectralMod*. We briefly highlight the main features of these algorithms; parameter settings used are given in the Supplementary Material. *Network Blast*, designed for comparing multiple protein networks but applicable for clustering a single protein network, greedily builds small 'dense' clusters. Highly overlapping clusters are filtered by the program. *Clique Finder (CFinder)* finds a set of k-clique percolation clusters, each of which consists of a maximal connected component of adjacent cliques of size k where two cliques are adjacent if they share $k-1$ nodes. We run *CFinder* with a range of k and keep all clusters of size $\leq$500; each protein may thus be in multiple clusters. *Markov clustering (MCL)* is a global clustering approach based on modified random walks on networks. *Density-periphery-based clustering (DPClus)* is a greedy approach that grows clusters based on adding nodes that are well connected to other nodes in the cluster and that maintain cluster density. *Molecular Complex Detection* (*Mcode*; Bader and Hogue, 2003) is one of the first approaches for clustering interactomes, and greedily grows clusters from a seed node. *Modularity-based spectral clustering* (*SpectralMod*) is a global procedure that iteratively cuts the network so that there are more than the expected number of edges within clusters (Newman, 2006). For each of these algorithms except *SpectralMod*, we download the software made available by the authors. For *SpectralMod*, we use software obtained from the author. For a baseline comparison, we also include a trivial algorithm, *OneCluster*, which always outputs a single cluster that includes all proteins in the network.

2.3 Evaluation measures for clustering

We evaluate clustering algorithms by judging how well the clusters correspond to groups of proteins as specified by MIPS complexes or functional modules as derived from either GO BP or GO CC annotations. Throughout the article, we refer to the output of the clustering algorithms as 'clusters' and the proteins comprising complexes or functional modules as 'groups'. Though cluster validation approaches are well-developed (e.g. see Handl *et al.*, 2005), much of this work has focused on either internal measures (i.e. the quality of the clusters are judged without desired groups in mind) or external measures where groups partition the data (i.e. the groups are non-overlapping). Since our groups are overlapping, these external measures are not directly applicable. For each of the three tasks we are considering (uncovering complexes, BP functional modules and CC functional modules), we utilize several measures to ascertain (i) how well each cluster maps to a known group and (ii) how well each group maps to a cluster. Depending on what we want to test, we utilize either one direction of these mappings (e.g. clusters to groups) or both directions (see Supplementary Fig. S1.) When we consider clustering in order to uncover protein complexes, there should be a one-to-one mapping of clusters and protein complexes. Thus, both directions of mappings are utilized. On the other hand, GO annotations are organized in a hierarchical fashion with respect to each other. So, even for a high-quality clustering where each cluster corresponds to a functional module, there may be functional modules to which no clusters correspond. Also, while proteins interacting with each other tend to have the same GO term and thus highly connected regions or clusters are likely to be enriched with GO terms, it may be less likely that all proteins with the same GO term are together in the same cluster. Therefore, in the case of functional modules, we evaluate a clustering only by mapping clusters to groups. It is important to note that when mapping a cluster to a GO term, each of the overlap measures we introduce below considers the total number of proteins annotated with that term (i.e. if that term annotates many proteins that are not part of the cluster, then the score for mapping that cluster to the term will be lower).

2.3.1 *Overlap measures*

We utilize three measures for evaluating clusters that are based on overlaps between clusters and known groups of proteins. Each measure gives a value in the range of 0–1, where higher numbers correspond to better overlaps. Before describing these measures, we give some preliminaries. Let M be the number of clusters given by a particular clustering, and N be the number of groups against which we are evaluating. Let C_j be the set of proteins within cluster j and let G_i be the set of proteins associated with the i-th group (e.g. in the i-th complex or annotated with i-th function). Our measures are as follows:

***Jaccard** measure*: given two sets, the Jaccard similarity coefficient is defined as the size of the intersection over the size of the union. For sets of proteins corresponding to cluster j and group i, let $\mathrm{Jac}_{ij}=\frac{|G_i \cap C_j|}{|G_i \cup C_j|}$ denote their Jaccard coefficient.

***PR** measure*: for sets of proteins corresponding to cluster j and group i, let $\mathrm{PR}_{ij}=\frac{|G_i \cap C_j|}{|C_j|}\cdot\frac{|G_i \cap C_j|}{|G_i|}$ denote their precision–recall (PR)-based score. The first part $\frac{|G_i \cap C_j|}{|C_j|}$ measures what fraction of the proteins in the cluster correspond to the grouping at hand (i.e. precision with respect to group i). The second part $\frac{|G_i \cap C_j|}{|G_i|}$ measures how much of group i is recovered by cluster j (recall). We note that our **PR**-based measures are similar to the F-measure (see, e.g. Handl *et al.*, 2005).

Semantic density measure: the density of a set of vertices in a network is typically defined as the number of edges among them divided by the maximum number of possible edges. We generalize this notion for protein interaction networks as follows to better recapitulate characteristics of the groups being compared to. For a set of proteins S, each protein $p \in S$ may be associated with labels $A(p) \subseteq \mathcal{A}$. For example, S can be a MIPS complex, $\mathcal{A}$ can be the set of clusters obtained after computational analysis of the entire interactome and $A(p)$ gives which clusters p belongs to. Alternatively, S can be a cluster of proteins, and $\mathcal{A}$ can be the set of groups of proteins with a shared functional annotation; in this case $A(p)$ gives the groups p is part of. Then,

$$\mathrm{density}(S,\mathcal{A})=\frac{\sum_{(p_1,p_2)\in S} W_{\mathcal{A}}(p_1,p_2)}{\sum_{(p_1,p_2)\in S}(1)}$$

where $W_{\mathcal{A}}(p_1,p_2)$, defined next, is the weight given to a pair of proteins p_1,p_2 and is in the range of 0–1. When considering clusters as $\mathcal{A}$, $W_{\mathcal{A}}(p_1,p_2)=1$ if $A(p_1)\cap A(p_2)\neq\emptyset$, and 0 otherwise. This weight function is also used when considering MIPS complexes as $\mathcal{A}$. When GO-derived functional groups are used as $\mathcal{A}$, the weight function is defined using a standard semantic similarity measure (Lord *et al.*, 2003). In particular, let $f(a)$ for functional group a be defined as the fraction of the total number of proteins in the considered network that have annotation a, and let $s(a)=-\log(f(a))$ be a measure of how specific the annotation is. Then,

$$W_{\mathcal{A}}(p_1,p_2)=\frac{2\cdot\max_{a\in A(p_1)\cap A(p_2)} s(a)}{\max_{a\in A(p_1)} s(a)+\max_{a\in A(p_2)} s(a)}$$

2.3.2 *Mapping scores*

Before describing our mapping scores, we briefly highlight some of our choices in computing these. First, some clustering approaches attempt to cluster all proteins (e.g. MCL and spectral clustering), whereas others leave many proteins unclustered. We chose to consider the unclustered proteins as singleton clusters, instead of ignoring them in the evaluation. Second, we remove from consideration all proteins in the complex and functional module groups that are not included in the network at hand. Third, when mapping a cluster to a group, we did not consider proteins that do not have any annotations from the grouping at hand. This means that clusters are filtered so as to remove any unannotated proteins, thereby potentially changing the size of the cluster. In practice, clusters consisting of mostly unannotated proteins could be considered as putative protein complexes or functional modules in further analysis. Fourth, when mapping a group to a cluster (performed only in the MIPS analysis), all proteins in the clusters, including unannotated ones, are considered. Fifth, when combining measures, we take a weighted average over clusters (or groups); other alternatives include, for example, taking unweighted averages or mapping the cluster measures to the proteins within them and averaging these values over proteins instead.

For each overlap measure described above, we utilize three 'scores'. First, we define scores for a clustering that measure how well clusters map to known groupings of proteins. For each cluster C_j, we find the group G_i that maximizes the overlap between it and cluster C_j. That is, we define $\mathbf{Jaccard}C_j=\max_i \mathrm{Jac}_{ij}$ for the **Jaccard** measure, $\mathbf{PR}C_j=\max_i \mathrm{PR}_{ij}$ for the **PR** measure and $\mathbf{sDensity}C_j=\mathrm{density}(C_j,\mathcal{G})$ for the **sDensity** measure where $\mathcal{G}$ is the set of groupings we are considering. If cluster C_j is a singleton cluster, then we define $\mathbf{Jaccard}C_j=\mathbf{PR}C_j=\mathbf{sDensity}C_j=0$. For each measure, we take an average over the clusters, weighted by cluster size, to obtain $\mathbf{Jaccard}C$, $\mathbf{PR}C$ and $\mathbf{sDensity}C$. That is, $\mathbf{Jaccard}C=\frac{\sum_{j=1}^{M}|C_j|\cdot\mathbf{Jac}C_j}{\sum_{j=1}^{M}|C_j|}$, and the other two measures are defined analogously. Since we are averaging over clusters, depending on the clustering approach and the evaluation task at hand, it may be preferable to filter highly overlapping clusters before calculating these measures. We note that $\mathbf{sDensity}C$ is similar to the biological homogeneity measure utilized previously to evaluate gene expression clusters (Datta and Datta, 2006).

Next, we define scores for a grouping that measure how well the known groups of proteins correspond to clusterings. Here, for each group G_i, we try to find cluster C_j that maximizes the overlap between it and the group G_i. That is, we define $\mathbf{Jaccard}G_i=\max_j \mathrm{Jac}_{ij}$ for the **Jaccard** measure, $\mathbf{PR}G_i=\max_j \mathrm{PR}_{ij}$ for the **PR** measure and $\mathbf{sDensity}G_i=\mathrm{density}(G_i,\mathcal{C})$ for the **sDensity** measure where $\mathcal{C}$ is the set of clusters we are considering.

For each measure, we take an average over the groups, weighted by group size, to obtain **Jaccard**G, **PR**G and **sDensity**G. That is, **Jaccard**$G = \frac{\sum_{i=1}^{N}|G_i|\cdot \mathbf{Jac}G_i}{\sum_{i=1}^{N}|G_i|}$, and the other two measures are defined analogously.

Finally, we define **Jaccard** as the harmonic mean of **Jaccard**C and **Jaccard**G; **PR** as the harmonic mean of **PR**C and **PR**G; and **sDensity** as the harmonic mean of **sDensity**C and **sDensity**G.

2.4 Quality of performance metrics

We demonstrate the utility of our performance metrics by using them to evaluate clusters found in random networks versus real networks. We generated 10 random networks for each of the four networks under consideration using a degree-preserving stub-rewiring algorithm (Newman *et al.*, 2001). We ran the six clustering algorithms on the four original networks as well as their randomized versions using the parameters given above. We find that the performance, as judged by our introduced measures, of each clustering approach is better in real networks than the corresponding randomized networks (Supplementary Table S2). For example, when considering *CFinder* on the HTP network and using either MIPS, BP or CC as the desired set of groupings, each of the three measures is >8.2 times larger on the real network than its average over 10 random networks; these ratios are >2.3 for *SpectralMod*, >1.6 for *DPClus*, >19.2 for *Mcode*, >1.5 for *MCL* and >1.5 for *NetworkBlast* (We note that this analysis also provides some information about the quality of the underlying clustering algorithms; for example, the measures always stay the same for the trivial *OneCluster* algorithm).

The **sDensity** measure seems to be the best with respect to its ratio in real versus random networks. For example, when considering *CFinder* on the HTP network, and using either MIPS, BP or CC as the desired set of groupings, **sDensity** is >47 times larger on the real network than its average over the randomized networks (Supplementary Table S2). Overall, our performance evaluations metrics are typically much higher when evaluating clusters derived from real networks as compared with those derived from random networks, demonstrating the strength of our measures. Of the 72 evaluations we performed (4 networks, 6 algorithms and 3 groupings), the only exception to this is *NetworkBlast*'s performance in recapitulating BP and CC modules from the Y2H network; in this case, **Jaccard** is on average better in the randomized networks than the actual network (data not shown). We note that the previously introduced separation, positive predictive value and accuracy measures for evaluating interactome-derived clusters (Brohée and van Helden, 2006) were often similar in value on clusters derived from networks corresponding to single high-throughput datasets as they were on the corresponding randomized networks.

As a sanity check, we also constructed three ideal clusterings, each of which corresponds exactly to the groupings we are trying to recover (i.e. protein complexes or functional modules), and evaluated each of those clusterings with respect to all three groupings to compute the maximum performance based on the evaluation framework. We see that performances of ideal clusterings are excellent when compared with the appropriate grouping, as expected (Supplementary Fig. S2). While ideal clusterings obtain **Jaccard** and **PR** values of 1.0, we note that the performances of ideal clusterings for GO terms evaluated by **sDensity** are lower; this is because of the characteristics of the weight function used.

2.5 Protein function prediction

2.5.1 Protein function prediction based on clustering Given a set of clusters, each protein i is scored with respect to each function f in the following way. For protein i in a cluster, we compute the P-value of all other member proteins in the same cluster having function f based on the hypergeometric distribution (i.e. with parameters as the number of proteins in the entire network, the number of proteins in the cluster, the number of proteins annotated with f in the network and the number of proteins annotated with f in the cluster). If protein i belongs to multiple clusters, the score for function f is taken to be the minimum P-value computed for this function over all clusters to which it belongs.

2.5.2 Protein function prediction via the neighborhood algorithm The *Neighborhood* algorithm scores each protein i with respect to function f using the hypergeometric distribution to compute the P-value of protein i's direct interactions having function f.

2.5.3 Evaluation of algorithms for protein function prediction Since there are parent–child relationships between terms in the BP and CC GO ontologies, for each protein, we update the predictions to deal with such a hierarchy. In particular, for each protein, we update P-values for the functions so that the P-value of a parent functional term is set to be less than or equal to the P-value of any of its children. Thus, given a threshold, if a term is predicted for protein i, then its parent terms are always predicted for protein i (the rationale being that a protein cannot have a more specific functional annotation without having more general terms as well). We utilize a PR curve, as suggested by (Deng *et al.*, 2003), where we vary the P-value threshold from 0 and 1. For protein i, let m_i be its functional annotations, n_i be a set of predicted functions for i based on the P-value threshold and k_i be the overlap between m_i and n_i. Then, recall and precision are defined as:

$$\text{recall} = \frac{\sum_i |k_i|}{\sum_i |m_i|},\ \text{precision} = \frac{\sum_i |k_i|}{\sum_i |n_i|}.$$

We note that, as outlined earlier, we do not consider overly general or specific functional terms within the ontology. Moreover, proteins within the network that are not annotated by any of these terms are ignored in computing the precision and recall.

3 RESULTS

3.1 Recapitulating protein complexes and functional modules

We give our performance metrics measuring how well the uncovered clusters correspond to protein complexes, BP functional modules and CC functional modules (Fig. 1) using the six studied algorithms applied to the HTP and Y2H networks. Results for all four networks are given in Supplementary Figures S3–S5. The run-times of the clustering algorithms on the HTP network are given in Supplementary Table S3. The clustering algorithms vary in the number of clusters they find in each network, as well as the number of singleton proteins left after clustering (Supplementary Table S4). On the HTP network, the algorithms find between 40 and 913 clusters of size >1 covering between 631 and 4160 proteins, and on the Y2H network, the algorithms find between 34 and 815 clusters of size >1 covering between 133 and 2828 proteins.

3.1.1 Stark performance differences in clustering algorithms We find significant differences in how well the clustering algorithms perform in recapitulating functional modules and protein complexes. For instance, on the HTP network, the quality of *CFinder*'s and *NetworkBlast*'s clusters, with respect to recapitulating complexes as well as BP and CC functional modules, are 1.6–5.0 times better than that of *SpectralMod* (Fig. 1 and Supplementary Figs S3–S5), according to our measures. These two approaches also significantly outperform *Mcode*. Part of the performance difference is due to the number of unclustered proteins: *Mcode* only clusters 631 of the proteins in this interaction network, whereas *NetworkBlast* and *CFinder* cluster 1371 and 1335, respectively. The significant differences in the performances of these algorithms on the various networks confirm that algorithm choice plays an important role in interactome analysis.

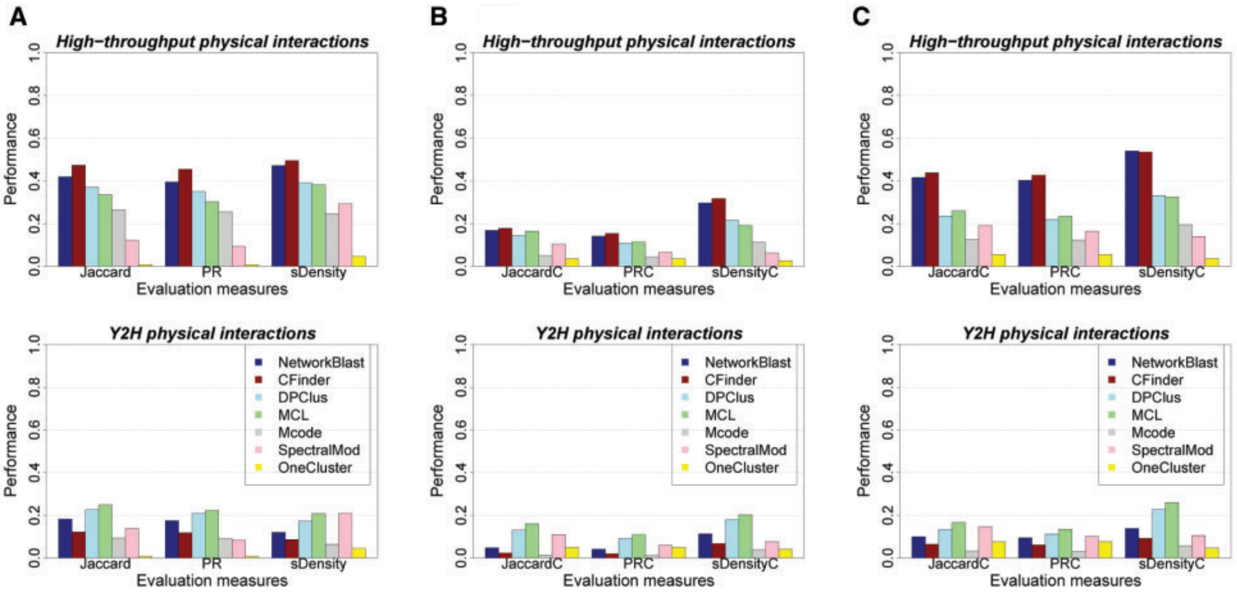

Fig. 1. Performance as judged via three measures (**Jaccard**, **PR** and **sDensity**) of six clustering algorithms and *OneCluster* in how well they recapitulate (**A**) MIPS complexes, (**B**) BP modules and (**C**) CC modules from the HTP (top) and Y2H (bottom) *S.cerevisiae* networks.

3.1.2 No single clustering approach performs best on all networks Different algorithms perform better on the Y2H network in recovering protein complexes and functional modules than those that perform best on the other networks studied. In particular, on the Y2H network, *MCL* performs better than the other approaches, with *DPClus* and *SpectralMod* also demonstrating good performance, whereas *NetworkBlast* and *CFinder* output higher quality clusters than the other algorithms on the other networks. The Y2H network is significantly different from the other three; for example, its average node degree and average node clustering coefficient are significantly lower (Supplementary Table S1). Relative changes in the performances of the clustering methods are also evident in networks obtained from subsampling from the original HTP network (see Supplementary Material and Fig. S6), and these changes vary depending on the network at hand. In the subsampled networks whose degree distribution and node clustering coefficients most closely match the Y2H network (Supplementary Table S5), the clustering methods' relative performances are similar to those seen in the Y2H network (Supplementary Fig. S6).

Overall, the relative change in performance of the clustering algorithms on different networks suggests that there is no clearly superior algorithm in all cases, but that instead algorithm choice should depend on network characteristics. In particular, for dense, well-studied interactomes, *NetworkBlast* and *CFinder* may uncover higher quality clusters as compared with the other methods, but for less studied organisms with sparser experimentally determined interaction networks, *MCL* may be a better choice.

Specific algorithmic properties of the approaches give hints to the situations to which they are well suited, and can be used to guide algorithm choice. For instance, *SpectralMod* tends to output large clusters as compared with other clustering algorithms when the network is dense. Thus, it appears to be more suitable for finding large functional modules which correspond to general GO terms as opposed to uncovering more specific functional modules. If a network is very sparse, *SpectralMod* will divide it into many more clusters and will have relatively better performance in our framework. Indeed, *SpectralMod* works comparatively better in the Y2H network than in the other networks. On the other hand, *CFinder* is based on finding 'dense' regions in the network; it detects a fewer number of clusters in the sparse Y2H network as compared with the other networks and leaves ∼90% of the proteins as singletons. This is a major contributing factor as to why its performances deteriorates in this network.

3.1.3 Advantages of using more complete networks in uncovering complexes and modules In general, clusters obtained using the full network consisting of all physical and genetic interactions (Supplementary Figs S3–S5A) better recapitulate functional modules and protein complexes than those obtained using the other networks. An interesting exception is that CC functional modules are somewhat better recapitulated (Supplementary Fig. S5A and B) when using physical interactions only. Genetic interactions are found between (related) pathways (Kelley and Ideker, 2005), though are also found within essential complexes (Boone *et al.*, 2007). Depending on the task at hand, it may be advantageous to treat these physical and genetic interactions separately (Brady *et al.*, 2009). Importantly, clusters obtained using just the Y2H network are significantly worse using all measures in recapitulating functional modules and protein complexes. Additionally, clusters obtained from the HTP network better recapitulate modules and complexes than those obtained from networks subsampled from the HTP network (Supplementary Fig. S6).

3.2 Predicting protein function

Protein physical interaction data is often utilized to predict protein function. The simplest approach is based on guilt-by-association (Schwikowski *et al.*, 2000), where a protein is assigned

Table 1. PR AUC for BP and CC predictions of six clustering algorithms and *Neighborhood* in the HTP *S.cerevisiae* network

	Spectral Mod	*DPClus*	*Mcode*	*MCL*	*CFinder*	*Network Blast*	*Neighborhood*
(A) Function prediction							
BP	0.0411	0.1557	0.0975	0.1213	0.1276	0.1082	0.1784
CC	0.1337	0.3309	0.1890	0.3003	0.2789	0.2506	0.3743
(B) Function prediction when factoring out singleton clusters							
BP	0.0411	0.1593	0.1625	0.1251	0.1498	0.1432	0.1784
CC	0.1337	0.3467	0.3512	0.3084	0.3140	0.2985	0.3743
(C) Function prediction when factoring out singleton clusters and large clusters							
BP	0.1320	0.1593	0.1625	0.1251	0.1491	0.1432	0.1784
CC	0.3189	0.3467	0.3512	0.3084	0.3253	0.2985	0.3743
(D) Function prediction when factoring out singleton clusters and poorly annotated clusters							
BP	0.1715	0.1755	0.1677	0.1654	0.1787	0.1444	0.1784
CC	0.3331	0.3604	0.3577	0.3420	0.3577	0.3054	0.3743
(E) Function prediction when local topology is considered for clustering algorithms							
BP	0.1716	0.1679	0.1710	0.1546	0.1827	0.1815	0.1784
CC	0.3535	0.3521	0.3646	0.3496	0.3772	0.3676	0.3743

See text for details.

a function based on those that are found frequently amongst its interacting proteins. Alternatively, to better utilize global information, a physical interactome can be clustered first, and then a protein is assigned the functions that are found to be overrepresented in its clusters. This more sophisticated cluster-based approach is widely used to obtain hints about protein function. We utilize leave-one-out cross-validation to compare clustering-based methods with a variant of a neighbor majority algorithm, *Neighborhood*, which makes a prediction for a protein based on the overrepresented functions found amongst its interacting proteins.

3.2.1 Local approaches outperform clustering in predicting protein function Table 1 (panel A) shows the area under the PR curve (AUC) for each algorithm in the HTP interaction network. Surprisingly, the simple *Neighborhood* approach has a higher AUC than all clustering algorithms in predicting either BP or CC terms. These results are consistent with earlier work showing that a 'neighborhood majority' approach based on total counts performs as well or better than several sophisticated global network approaches (Nabieva *et al.*, 2005), as well as earlier work suggesting that *Neighborhood* performs better than *Mcode* for the task of function prediction (Sharan *et al.*, 2007).

In order to assess whether the lower accuracies of clustering algorithms come from their inability to predict function for proteins in singleton clusters, we also considered function prediction when factoring out singleton clusters. We note that it is necessary to have the same test set when comparing different clustering approaches with PR AUCs, as baseline performance varies with different test sets. Since each of the clustering approaches leaves a different number of proteins in singleton clusters, for all approaches, we use *Neighborhood* to predict the functions of these proteins, so that we are only considering the performance of transferring function within larger clusters [Table 1 (panel B)]. The *Neighborhood* approach still outperforms the clustering-based approaches, though clustering algorithms such as *Mcode* which leave many proteins unclustered see a clear boost in performance.

In order to assess whether the lower accuracies of the clustering algorithms come from transferring functions within large clusters, we next additionally exclude large-sized clusters. For clusters with size greater than 50, as well as those with size 1, we again use *Neighborhood* for proteins within those clusters. For the remaining clusters, we transfer functions according to hypergeometric distribution [Table 1 (panel C)]. We still observe that *Neighborhood* has a higher AUC than the clustering-based approaches.

In order to assess whether the lower accuracies of the clustering algorithms come from transferring functions that are infrequent, we next exclude poorly annotated clusters. That is, we filter out clusters where there are no functions annotating more than 50% of the member proteins, and use *Neighborhood* for proteins within those clusters as well as within singleton clusters. For the remaining non-singleton and well-annotated clusters, we transfer functions according to hypergeometric distribution but require that they annotate at least half the proteins in the cluster [Table 1 (panel D)]. We still observe that *Neighborhood* has a higher AUC than the clustering-based approaches.

3.2.2 Combining local and clustering approaches for function prediction One hypothesis for why cluster-based approaches do not work as well as the local approaches for function prediction is that a cluster may be composed of several functional modules, and while functions may be statistically enriched within them, they should not be transferred to all the members of the cluster. Accordingly, we next combine clustering information with neighbor annotation information. That is, for each protein within a cluster, we use the *Neighborhood* approach but only consider its interacting proteins within the same cluster while ignoring other proteins within its cluster as well as interacting proteins that are in different clusters. For proteins that are not clustered, the *Neighborhood* approach is used while considering all its interactions. In this case, clustering approaches such as *NetworkBlast* or *CFinder* have slightly higher AUCs than *Neighborhood* [Table 1 (panel E)]. The drastic improvement of *SpectralMod*'s PR-AUC also supports the idea that *SpectralMod*'s large clusters consist of several smaller functional modules.

3.2.3 Characterizing cluster-based function prediction based on the number of annotations While the *Neighborhood* approach has better performance than cluster-based methods in predicting protein functions for *S.cerevisiae*, clustering approaches have other advantages. In particular, they can uncover structure in networks with no additional information and can make predictions for proteins that interact only with proteins of unknown function. Thus, we expect that for proteomes with larger numbers of unannotated

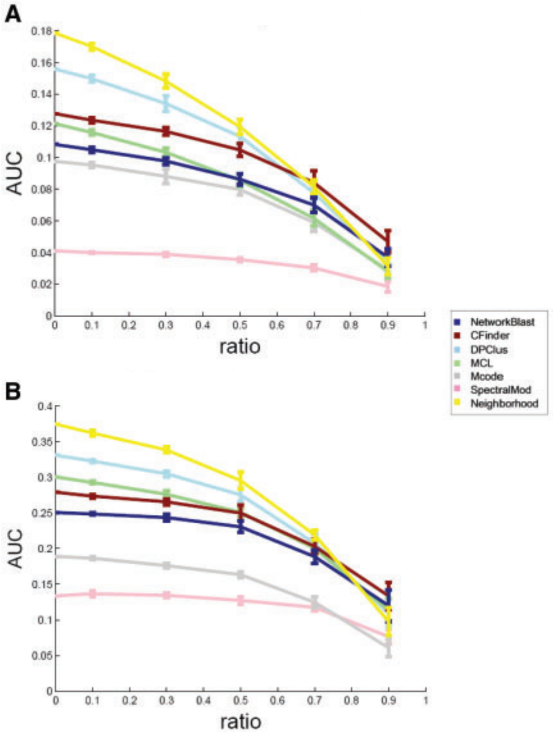

Fig. 2. Function prediction performance as protein annotations are removed. As BP (**A**) or CC (**B**) annotations are removed for 10%, 30%, 50%, 70% and 90% of the proteins in the HTP interaction network, the PR AUC of *Neighborhood* deteriorates more rapidly than that of any of the six clustering algorithms. The average PR AUC over 10 networks is plotted, with each error bar showing ±1SD from the average.

proteins, the performance of the *Neighborhood* approach should decrease faster than that of clustering-based approaches. In order to systematically test this, we analyze how the algorithms perform as we remove annotations from the proteins in the network. That is, we selected 10%, 30%, 50%, 70% and 90% of the proteins in the HTP network at random and removed all of their annotations to make an artificial network with fewer annotations. Figure 2 show the PR AUC as a function of the fraction of proteins whose annotations are removed, with the values at 0% annotation removed corresponding to those in Table 1 (panel A).

As a large fraction of the yeast proteins' annotations are removed, the clustering-based approaches begin to outperform *Neighborhood*. For example, when nearly 70% of the annotations are removed, *CFinder* outperforms *Neighborhood* for BP prediction and *DPClus* outperforms *Neighborhood* for CC prediction. Most clustering algorithms predict better than *Neighborhood* once 90% of the annotations are deleted. We find the same trends when running this procedure on the other yeast networks. See Supplementary Figure S7 for our results on the Y2H network, where *Neighborhood* still outperforms the clustering approaches in function prediction on the original network and on networks with a considerable fraction of annotations removed. On the human physical interaction network from BioGRID, where a somewhat smaller fraction of proteins are annotated (51% with BP terms and 31% with CC terms), *Neighborhood* still outperforms the clustering approaches, though *MCL* is competitive with it (Supplementary Fig. S8) and performs better than it when ∼10% of the annotations are removed. If predictions on unannotated proteins are pessimistically counted as false positives (instead of ignored), then the *Neighborhood* method outperforms the other approaches until 50% of the annotations in the human network are removed (Supplementary Fig. S9). In all networks studied, the relative performances of clustering methods in function prediction as compared with the *Neighborhood* method improve as annotations are removed.

4 DISCUSSION AND CONCLUSIONS

While clustering has become a standard first-line tool in the analysis of physical interactomes, no previous study has systematically assessed how well such an approach performs in predicting protein function and functional modules. Our research establishes guidelines on how and when clustering should be utilized for analyzing physical interaction networks.

Perhaps most importantly, we find that the common practice of looking for enriched functions within clusters is not the best approach for predicting protein function, at least for the yeast proteome. Instead we find that, overall, it is better to use a simple local method such as *Neighborhood* or to use clustering algorithms combined with *Neighborhood* in networks with sufficient annotations. From a computational perspective, this also suggests that clustering algorithms should not be judged solely based on the number of functionally enriched clusters they find, as this may not be the best way to do interactome-derived function prediction for the proteome at hand. The strength of clustering is that it uncovers structure within biological networks, even when nothing is known about individual proteins. Thus, for less annotated proteomes, or even BPs that have not been well-studied, the advantages of clustering over local methods are more likely to be apparent. Indeed, our simulations show that the relative performance of clustering approaches as compared with a simple neighborhood functional annotation scheme improves with fewer annotations. In the future, it would be desirable to characterize which method should be used for function prediction at the per-protein level; this could depend on, for example, the number of annotated interacting proteins, local measures of network topology, the density and size of the clusters it is found within and the particular functions being predicted.

We also find that the topological features of networks can vastly affect the performance of some clustering algorithms in recapitulating functional modules; in particular, some of the best performing algorithms on the more dense HTP network are among the poorest performing in the Y2H network as well as in networks subsampled from the HTP network to resemble the Y2H network with respect to network topological features (Supplementary Fig. S6). This suggests that network characteristics should guide algorithm choice, and there is no one algorithm that always outperforms others in predicting functional modules. It is possible that for some clustering approaches, more fine-tuned parameter choices may lead to better results; however, for approaches such as *CFinder* and *SpectralMod*, which have one and zero parameters, respectively, and whose relative performances swap between the HTP and Y2H networks, this is not the case.

Looking forward, we believe that greater efforts should be made in the future to evaluate biological clustering algorithms. We hope that our evaluation framework provides a good starting point for gauging how well future methodological advances in clustering translate to better detection of functional modules and protein complexes from interactomes.

ACKNOWLEDGEMENTS

The authors thank S. Brohee and J. van Helden for providing us with their filtered MIPS dataset, and M. Newman for providing us with the code for his modularity-based spectral clustering method. The authors thank E. Banks and E. Nabieva for providing some of the software utilized for this study, and all members of the Singh group for helpful discussions and comments on the manuscript. The authors also thank the anonymous reviewers for their helpful suggestions.

Funding: National Science Foundation (NSF CCF-0542187, NSF IIS-0612231, in part), National Institute of Health (NIH GM076275, in part); NIH Center of Excellence (grant P50 GM071508, in part). Samsung Foundation of Culture (Samsung Scholarship to J.S.).

Conflict of Interest: none declared.

REFERENCES

Adamcsek,B. *et al.* (2006) Cfinder: locating cliques and overlapping modules in biological networks. *Bioinformatics*, **22**, 1021.

Altaf-Ul-Amin,M. *et al.* (2006) Development and implementation of an algorithm for detection of protein complexes in large interaction networks. *BMC Bioinformatics*, **7**, 207.

Arnau,V. *et al.* (2005) Iterative cluster analysis of protein interaction data. *Bioinformatics*, **21**, 364–378.

Ashburner,M. *et al.* (2000) Gene Ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat. Genet.*, **25**, 25–29.

Asthana,S. *et al.* (2004) Predicting protein complex membership using probabilistic network reliability. *Genome Res.*, **14**, 1170–1175.

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

Barabási,A.L. and Oltvai,Z.N. (2004) Network biology: understanding the cell's functional organization. *Nat. Rev. Genet.*, **5**, 101.

Boone,C. *et al.* (2007) Exploring genetic interactions and networks with yeast. *Nat. Rev. Genet.*, **8**, 437–449.

Brady,A. *et al.* (2009) Fault tolerance in protein interaction networks: stable bipartite subgraphs and redundant pathways. *PLos One*, **4**, e5364.

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

Brun,C. *et al.* (2003) Functional classification of proteins for the prediction of cellular function from a protein-protein interaction network. *Genome Biol.*, **5**, R6.

Chen,J. and Yuan,B. (2006) Detecting functional modules in the yeast protein-protein interaction network. *Bioinformatics*, **22**, 2283–2290.

Datta,S. and Datta,S. (2006) Methods for evaluating clustering algorithms for gene expression data using a reference set of functional classes. *BMC Bioinformatics*, **7**, 397.

Deng,M. *et al.* (2003) Prediction of protein function using protein-protein interaction data. *J. Comput. Biol.*, **10**, 947.

Dunn,R. *et al.* (2005) The use of edge-betweenness clustering to investigate biological function in protein interaction networks. *BMC Bioinformatics*, **6**, 39.

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

Hartwell,L.H. *et al.* (1999) From molecular to modular cell biology. *Nature*, **402** (Suppl. 6761).

Handl,J. *et al.* (2005) Computational cluster validation in post-genomic data analysis. *Bioinformatics*, **21**, 3201–3212.

Kelley,R. and Ideker,T. (2005) Systematic interpretation of genetic interactions using protein networks. *Nat. Biotechnol.*, **23**, 561–566.

King,A. *et al.* (2004) Protein complex prediction via cost-based clustering. *Bioinformatics*, **20**, 3013–3020.

Lord,P.W. *et al.* (2003) Semantic similarity measures as tools for exploring the gene ontology. *Pac. Symp. Biocomput.*, **8**, 601.

Luo,F. *et al.* (2007) Modular organization of protein interaction networks. *Bioinformatics*, **23**, 207–214.

Mewes,H.W. *et al.* (2004) MIPS: analysis and annotation of proteins from whole genomes. *Nucleic Acids Res.*, **32**, D41–D44.

Nabieva,E. *et al.* (2005) Whole-proteome prediction of protein function via graph-theoretic analysis of interaction maps. *Bioinformatics*, **21** (Suppl. 1), i302–i310.

Navlakha,S. *et al.* (2009) Revealing biological modules via graph summarization. *J. Comput. Biol.*, **16**, 253–264.

Newman,M.E.J. (2006) Modularity and community structure in networks. *Proc. Natl Acad. Sci. USA*, **103**, 8577.

Newman,M.E.J. *et al.* (2001) Random graphs with arbitrary degree distributions and their applications. *Phys. Rev. E*, **64**, 026118.

Pereira-Leal,J. *et al.* (2004) Detection of functional modules from protein interaction networks. *Proteins*, **54**, 49–57.

Poyatos,J. and Hurst,L. (2004) How biologically relevant are interaction-based modules in protein networks? *Genome Biol.*, **5**, R93.

Radicchi,F. *et al.* (2004) Defining and identifying communities in networks. *Proc. Natl Acad. Sci. USA*, **101**, 2658–2663.

Rives,A.W. and Galitski,T. (2003) Modular organization of cellular networks. *Proc. Natl Acad. Sci. USA*, **100**, 1128–1133.

Samanta,M. and Liang,S. (2003) Predicting protein functions from redundancies in large-scale protein interaction networks. *Proc. Natl Acad. Sci. USA*, **100**, 12579–12583.

Schlitt,T. *et al.* (2003) From gene networks to gene function. *Genome Res.*, **13**, 2568–2576.

Schwikowski,B. *et al.* (2000) A network of protein-protein interactions in yeast. *Nat. Biotechnol.*, **18**, 1257–1261.

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

Sharan,R. *et al.* (2007) Network-based prediction of protein function. *Mol. Syst. Biol.*, **3**, 88.

Spirin,V. and Mirny,L.A. (2003) Protein complexes and functional modules in molecular networks. *Proc. Natl Acad. Sci. USA*, **100**, 12123.

Stark,C. *et al.* (2006) BioGRID: a general repository for interaction datasets. *Nucleic Acids Res.*, **34**, D535–D539.

von Mering,C. *et al.* (2003) Genome evolution reveals biochemical networks and functional modules. *Proc. Natl Acad. Sci. USA*, **100**, 15428–15433.

Wang,C. *et al.* (2007) Consistent dissection of the protein interaction network by combining global and local metrics. *Genome Biol.*, **8**, R271.