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

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 should be used for interactome functional analyses. Overall our results suggest a re-examination of when and how clustering

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), 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; Arnaud et al., 2005; Ashburner et al., 2000; Brun et al., 2003; Chen and Yuan, 2006; Dunn et al., 2005; Enright et al., 2002; King et al., 2004; Lu et al., 2007; Navlakha et al., 2009; Newman, 2006; Poyatos and Hurst, 2004; Radicchi et al., 2004; Samanta and Liang, 2003; Schliit 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
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 (Broggié 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 \textit{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
\textit{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 \textit{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 \textit{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 11928 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 \textit{S.cerevisiae} protein
complexes from MIPS; this is the same set as used in the study of Broggieé
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 \textit{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
specificity and size.

2.2 Clustering algorithms

We consider six network clustering algorithms: NetworkBlust (Sharan et al.,
2005), Cfinder (Adamiczek et al., 2006), MCL, (Enright et al., 2002),
DPClus (Altaf-Ul-Amin et al., 2006), Moby (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
Blust, 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 ≥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 (Moby; 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$_i$ be the set of proteins within cluster $i$ and let G$_j$ be the set of proteins associated with the j-th group (e.g. in the j-th complex or annotated with j-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 $Jac_C_{ij} = \frac{|C_j \cap G_i|}{|C_j \cup G_i|}$ denote their Jaccard coefficient.

**PR measure**: for sets of proteins corresponding to cluster $j$ and group $i$, let $PR_{ij} = \frac{|C_j \cap G_i|}{|C_j|}$ denote their precision-recall (PR)-based score. The first part $\frac{|C_j \cap G_i|}{|C_j|}$ 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 $\frac{|C_j \cap G_i|}{|G_i|}$ 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 $Ap \subseteq A$. For example, $S$ can be a MIPS complex, $A$ can be the set of clusters obtained after computational analysis of the entire interactome and $Ap$ gives which clusters $p$ belongs to. Alternatively, $S$ can be a cluster of proteins, and $A$ can be the set of groups of proteins with a shared functional annotation; in this case $Ap$ gives the proteins $p$ is part of. Then,

$$density(S,A) = \frac{\sum_{p \in S, p \in Ap} |W_A(p_1,p_2)|}{\sum_{p \in S, p \in Ap} \binom{|Ap|}{2}}$$

where $W_A(p_1,p_2)$, 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 $A$, $W_A(p_1,p_2) = 1$ if $Ap_1 \cap Ap_2 \neq \emptyset$, 0, and undefined otherwise. This weight $W_A(p_1,p_2)$ is similar to considering MIPS complexes as $A$. When GO-derived functional groups are used as $A$, 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 $s(a) = -\log(\frac{|a|}{\sum_{a} |a|})$ be a measure of how specific the annotation is. Then,

$$W_A(p_1,p_2) = \frac{2 \cdot \max_{a \in Ap_1 \cap Ap_2} \{f(a) \cdot s(a)\} \cdot \log(|Ap_1 \cap Ap_2|)}{\max_{a \in Ap_1} \{f(a) \cdot s(a)\} + \max_{a \in Ap_2} \{f(a) \cdot s(a)\}}$$

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 groups 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 $Jaccard_C_i = \max_{G_i \in S} Jac_C_{ij}$ for the Jaccard measure, $PR_C_i = \max_{G_i \in S} PR_{ij}$ for the PR measure and $\text{Density}_C_i = \max_{G_i \in S} \frac{|C_j \cap G_i|}{|G_i|}$ for the density measure where $G_i$ is the set of groups we are considering. If cluster $C_j$ is a singleton cluster, then we define $Jaccard_C_i = PR_C_i = \text{Density}_C_i = 0$. For each measure, we take an average over the clusters, weighted by cluster size, to obtain $\text{Jaccard}_C$, $\text{PR}_C$ and $\text{Density}_C$. That is, $\text{Jaccard}_C = \frac{\sum_{j=1}^{M} Jaccard_C_i}{\sum_{j=1}^{M} |C_j|}$, 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 $\text{Density}_C$ 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 clusters. 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 $\text{Jaccard}_G_i = \max_{C_j \in C} Jac_C_{ij}$ for the Jaccard measure, $\text{PR}_G_i = \max_{C_j \in C} PR_{ij}$ for the PR measure and $\text{Density}_G_i = \max_{C_j \in C} \frac{|C_j \cap G_i|}{|G_i|}$ for the density measure where $C$ is the set of clusters we are considering.
We note that the previously introduced separation, positive predictive value
for each measure, we take an average over the groups, weighted by
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 term is also predicted for protein
(they are always on the same hierarchy). 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:

\[
\text{recall} = \frac{\sum k_i}{\sum m_i} \quad \text{precision} = \frac{\sum k_i}{\sum n_i}.
\]

We note that, as outlined earlier, we do not consider overly general or specific
functions 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 \( Meode \). Part of the performance difference is due to
the number of unclustered proteins: \( Meode \) 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.
Functional analysis of interactome-derived clusters

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 (see Supplementary Material and Fig. S6), 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.

3.2 Predicting protein function

Protein physical interaction data is often utilized to predict protein function. The simplest approach is based on guilt-byassociation (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 H. S. cerevisiae network

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

See text for details.

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 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
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 or 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