Graphlet-based edge clustering reveals pathogen-interacting proteins

R. W. Solava, R. P. Michaels and T. Milenkovic

Department of Computer Science and Engineering, University of Notre Dame, Notre Dame, IN 46556, USA

ABSTRACT

Motivation: Prediction of protein function from protein interaction networks has received attention in the post-genomic era. A popular strategy has been to cluster the network into functionally coherent groups of proteins and assign the entire cluster with a function based on functions of its annotated members. Traditionally, network research has focused on clustering of nodes. However, clustering of edges may be preferred: nodes belong to multiple functional groups, but clustering of nodes typically cannot capture the group overlap, while clustering of edges can. Clustering of adjacent edges that share many neighbors was proposed recently, outperforming different node clustering methods. However, since some biological processes can have characteristic ‘signatures’ throughout the network, not just locally, it may be of interest to consider edges that are not necessarily adjacent.

Results: We design a sensitive measure of the ‘topological similarity’ of edges that can deal with edges that are not necessarily adjacent. We cluster edges that are similar according to our measure in different baker’s yeast protein interaction networks, outperforming existing node and edge clustering approaches. We apply our approach to the human network to predict new pathogen-interacting proteins. This is important, since these proteins represent drug target candidates.

Availability: Software executables are freely available upon request.

Contact: tilen@nd.edu

Supplementary Information: Supplementary data are available at Bioinformatics online.

1 INTRODUCTION

Network research spans many domains. We focus on protein–protein interaction (PPI) networks, with the goal of identifying new pathogen-interacting proteins. In PPI networks, nodes are proteins and undirected edges correspond to physical binding between the proteins. We focus on PPI networks since it is the proteins, gene products, that carry out most biological processes, and they do so by interacting with other proteins. High-throughput screens for interaction detection, such as yeast two-hybrid (Y2H) assays or affinity purification coupled to mass spectrometry (AP/MS), have yielded partial PPI networks for many model organisms and human (Jeong et al., 2000; Simons et al., 2001; Stelzl et al., 2005; Bu et al., 2006) as well as for bacterial and viral pathogens (Milenković et al., 2005; Patrie et al., 2007). Many biological network datasets are now publicly available.

Analogous to genomic sequence research, biological network research will impact our biological understanding. However, it is in its infancy, owing to computational hardness of many graph theoretic problems (Cook, 1971), as well as to incompleteness of the current network data. Importantly, the number of functionally uncharacterized proteins is large even for well-studied model species (Sharan et al., 2007). Computational characterization of protein function could save resources needed for biological experiments. In particular, PPI network analysis could suggest likely candidates for experimental validation, since proteins aggregate to perform a function, and since PPI networks model these aggregations.

Thus, prediction of protein function (Sharan et al., 2007), and the role of proteins in disease (Cook et al., 2009; Rašič et al., 2008; Sharan and Ideker, 2006; Vazquez et al., 2004) from PPI networks have received attention in the post-genomic era. It has been argued that proteins that are close in the network are involved in similar biological processes (Sharan et al., 2007) that ‘topologically central’ proteins correspond to ‘biologically central’ (e.g. lethal, aging- or cancer-related) proteins (Jong et al., 2003; Breitkreutz et al., 2003; Milenković et al., 2006; Sharan and Ideker, 2006) or that proteins with similar topological neighborhoods have similar biological characteristics (Ho et al., 2002; Milenković and Pržulj, 2008).

A particularly popular strategy for functional characterization of proteins has been to cluster the network into functionally ‘coherent’ groups of nodes and assign the entire cluster with a function based on functions of its annotated members (Sharan and Ideker, 2006; Sharan et al., 2007). A variety of clustering approaches exist, each with its own (dis)advantages (Brohee and van Helden, 2006; Fortunato, 2010). Typically, they aim to group nodes that are in a dense connected network region (Fortunato, 2010). In addition, approaches exist that cluster ‘topologically similar’ nodes without the nodes necessarily being connected in the network. This is important, since a biological process can have characteristic topological ‘signatures’ throughout the network, not just locally in close network proximity (Ho et al., 2002; Milenković et al., 2006; Milenković and Pržulj, 2008). For example, we designed a measure that computes the topological similarity of the extended network neighborhoods of two nodes, without the nodes necessarily being close in the network (Milenković and Pržulj, 2008). We found that 96% of known cancer gene pairs that are topologically similar according to our measure are actually not neighbors in the PPI network; instead, they are at the shortest path distance of up to six (Milenković et al., 2011). As such, they may be missed by approaches that focus on connected nodes only. We clustered proteins in the human PPI network that are topologically similar and showed that function of a protein and its network position are closely related (Milenković and Pržulj, 2008) and that the topology around cancer and non-cancer genes is different (Milenković and Pržulj, 2008). We used these observations to predict new cancer genes in melanogenesis-related pathways and our predictions were validated phenotypically (Ho et al., 2011).

Traditionally, network research has focused on clustering of nodes (Fortunato, 2010). However, a network consists of nodes and edges.
We recently designed a graphlet-based measure of the network position of a node (Pržulj, 2008). This measure generalizes the degree of a node that counts the number of edges that the node touches (where an edge is the only 2-node graphlet) into the node graphlet degree vector (node-GDV) that counts the number of 2 to 5-node graphlets that the node touches (see Section 2). Hence, node-GDV describes the topology of the node’s up to 4-deep neighborhood (Milenkovic and Pržulj, 2008). This is effective: going to distance of four a node captures a large portion of real networks, as they are small-world (Watts and Strogatz, 1998). For this reason, and since the number of graphlets on n nodes increases exponentially with n, using larger graphlets could unnecessarily increase the computational complexity. In addition, we designed node-GDV-similarity measure to compare node-GDVs of two nodes and quantify the topological similarity of their extended network neighborhoods (Milenković and Pržulj, 2008).

Now, we design edge-GDV to count the number of different graphlets that an edge touches (Fig. 2), and we design edge-GDV-similarity to compare edge-GDVs of two edges and quantify the topological similarity of their extended network neighborhoods (see Section 2). Unlike the shared-neighborhood measure (Ahn et al., 2010), edge-GDV-similarity can deal with edges independent on whether they are adjacent. In addition, by counting the shared neighbors of end nodes of two (adjacent) edges, the shared-neighborhood measure actually counts the 3-node paths that the end nodes share (Ahn et al., 2010). Since edge-GDV counts the different up to 5-node graphlets that an edge touches, including 3-node paths, edge-GDV is a more constraining measure of topolgy. See Section 2 for details.

We cluster edge-GDV-similar edges in the human PPI network to predict new pathogen-interacting (PI) proteins from the clusters. But first, we compare our approach to existing clustering methods, as follows. The existing edge clustering method mentioned above, henceforth denoted by edge-shared-neighborhood (edge-SN) (Ahn et al., 2010), was already shown to be superior to different node clustering methods on four baker’s yeast PPI networks. With our approach, we cluster edges in the same way as edge-SN, but we use edge-GDV-similarity instead of the edge-SN’s shared-neighborhood measure as the distance metric, without changing other aspects of the clustering procedure. This way, we can evaluate the contribution of edge-GDV-similarity alone to the quality of clusters. Just as Ahn et al. (2010) (we initially) cluster only adjacent edges, and of all partitions, we choose the one with the maximum density (see Section 2). Just as Ahn et al. (2010) we evaluate such partition with respect to: cluster coverage (the portion of the network ‘covered’ by ‘non-trivial’ clusters), overlap coverage (the amount of node overlap between clusters), cluster quality (enrichment of clusters in Gene Ontology (GO) terms (Ashburner et al., 2000) and overlap quality (the correlation between the number of clusters and the number of GO terms that nodes participate in). When applied to the same yeast networks, our approach in comparable or superior to edge-SN (and hence to the node clustering approaches that were outperformed by edge-SN). Thus, we gain by using a more sensitive measure of topology compared to edge-SN. When we cluster both adjacent and non-adjacent edges, our method in general performs even better. Hence, we gain further by using a measure that can deal with edges that are not necessarily adjacent. We note that we do not propose a new clustering method but a new edge similarity measure that can serve as a distance metric for existing clustering methods.

After we evaluate our approach on yeast, we apply it to human, hypothesizing that if many end nodes of edges in a cluster are PI proteins, then the other end nodes of edges in that cluster are likely PI proteins. Here, instead of comparing different distance metrics within the same clustering method (as above), we aim to compare different clustering methods when using the same (best) distance metric. Hence, we use edge-GDV-similarity as the distance metric for two popular clustering methods: hierarchical and k-means (i.e. k-medoids) clustering (see Section 2). In addition, we aim to evaluate how much we gain by clustering edges compared to clustering nodes. This is partially answered on yeast, since we compare our method to edge-SN, which was already compared to different node clustering methods on the same networks. However, our edge clustering method (and hence edge-SN) is conceptually different than these node clustering methods. For a fair comparison...
of edge and node clustering, we use edge-GDV-similarity and node-
GDV-similarity, conceptually equivalent measures of the network position
of an edge and a node, respectively, as distance metrics for the same clustering method. We do this for both hierarchical and
k-medoids clustering.

We apply our clustering strategies to human to predict new
PI proteins: hierarchical node clustering (node-HIE), hierarchical
edge clustering (edge-HIE), k-medoids node clustering (node-
KM) and k-medoids edge clustering (edge-KM). We evaluate their
prediction accuracies in systematic leave-one-out cross-validation
and precision-recall settings (see Section 2). Interestingly, edge-
HIE is superior, followed by node-KM, edge-KM and node-HIE.

Since all clustering strategies except node-HIE produce non-
random results, for each of them, we identify clusters that are
subsequently significantly enriched in known PI proteins, and predict
novel PI proteins from the clusters. This way, we complement
HIE is superior, followed by node-KM, edge-KM and node-HIE.

We compare our method to three popular node clustering methods: clique
multiple strategies. We validate 44% of our predictions in the
overlap and identify 'high-confidence' predictions produced by
novel PI proteins from the clusters. This way, we complement
HIE is superior, followed by node-KM, edge-KM and node-HIE.

Therefore, we apply four clustering strategies to human to predict new
PI proteins from the clusters. This way, we complement
HIE is superior, followed by node-KM, edge-KM and node-HIE.

2 METHODS

2.1 Datasets

Yeast PPI networks. We cluster the same four baker’s yeast PPI networks
that edge-SN was evaluated on in Ahn et al. (2010), i.e. i) Y2H
network, obtained by Y2H, with 1647 proteins and 2518 PPIs; ii) AP/MS
network, obtained by AP/MS, with 1004 proteins and 8319 PPIs; iii) LC
network, obtained by literature curation, with 1215 proteins and 2556 PPIs
and iv) ALL network, representing the union of Y2H, AP/MS and LC, with
2729 proteins and 12 174 PPIs. Using these different networks ensures that
our method is robust to different types of experiments for PPI detection.

The human PPI network. The human PPIs were obtained in July 2011
from BioGRID (version 3.1.79) (Breitkreutz et al. 2004). We take the union of the two networks (using UniProt protein
IDs) to increase the coverage of the cellular space, resulting in 12 111 unique proteins and 59 191 unique PPIs.

Pathogen-interacting proteins. 9884 human–pathogen PPIs were obtained from VirusMint (Chatr-aryamontri et al. 2009) and Dyer
et al. (2010). The PPIs involve 1338 human proteins, of which 1113 are present in the human
PPI network, and 706 pathogen proteins from 193 pathogens.

2.2 Related work

We compare our method to three popular node clustering methods: clique
percolation (Palla et al. 2005), greedy modularity optimization (Newman
2004 and Infomap (Rosvall and Bergstrom 2008). In addition, we compare it to the existing edge clustering algorithm, edge-SN (Ahn et al. 2010).

Briefly, clique percolation is the most prominent overlapping node clustering
algorithm, greedy modularity optimization is the most popular modularity-
based technique, and Infomap is often considered the most accurate method available (Ahn et al. 2010). Edge-SN hierarchically groups adjacent edges
whose non-common end nodes share many neighbors (see below). We did not
run these algorithms on the yeast networks ourselves. Instead, we use
the results reported by Ahn et al. (2010), who ran the algorithms on the same networks we used and implemented, see Ahn et al. (2010). We do explain how Ahn et al. (2010) implemented edge-SN, as we implement our method in the same way (except that we use a different distance metric).

Edge-SN algorithm works as follows. If the set of node i and its neighbors is
denoted as n(i), the similarity between adjacent edges eij
and eik with common node k in S(eij, eik) = |n(i) ∩ n(j)|/|n(i) ∪ n(j)|. This
shared-neighborhood measure is used as a distance metric for single-linkage
hierarchical clustering. With this method, a tree or dendrogram, is created.
Leaves of the tree are edges of the network and an interior node in the
tree represents a cluster made up of all children of the node. The tree is
constructed by assigning each edge to its own cluster and iteratively merging
the most similar pair of clusters. The tree has to be cut in order to create a
partition of K clusters. To determine where to cut the tree, edge-SN uses an
objective function called partition density, computed as follows. For a
network with M edges, {P1, ..., PM} is a partition of the edges into K clusters.
Cluster C has nc = |C| edges and ne = |E(C)|/|C| nodes. C’s density is
DC = |nc - (nc - 1)/2| / |nc - (nc - 1)/2 - (nc - 1)/2|. For details, see Ahn
et al. (2010). Edge-SN cuts the tree at different levels and chooses a partition
with the maximum value of D. However, meaningful structure may also exist
above and below the level corresponding to maximum D (Ahn et al. 2010).

2.3 New measures of network topology: edge graphlet
degree vector (edge-GDV) and edge-GDV-similarity

An edge is an induced subgraph of graph X that contains all edges of
X connecting its nodes (Fig. 8). We generalized the degree of node v
to the number of edges that v touches (where an edge is the only 2-node
graph). Go in Fig. 8 into node graphlet degree vector (node-GDV) of v
that counts the number of 2–5-node graphlets (Go, G2, ..., Gp) in Fig. 8 that
v touches (Milenkovski and Przulj 2008). We need to distinguish between v
touching, e.g. a G2 at an end node or at the middle node, since G2 admits
an automorphism that maps its end nodes to each other and the middle
node to itself. To understand this, recall the following. An isomorphism
f from graph X to graph Y is a bijection of nodes of X to nodes of Y
such that v is an edge of X if and only if f(v)(f(v)) is an edge of Y. An
automorphism is an isomorphism from X to itself. The automorphisms of
X form the automorphism group, Aut(X). If x is a node of X, then
the automorphism node orbit of x is in Orb(x) = {x ∈ V(X) | f(x) for some
f ∈ Aut(X)}, where V(X) is the set of nodes of X. Thus, end nodes of A can
belong to one node orbit, while its middle node belongs to another one.
There are 73 node orbits for 2–5-node graphlets. Hence, node-GDV of v
has 73 elements counting how many node orbits of each type touch v’s (v’s degree)
is the first element). It captures v’s up to 4-deep neighborhood and thus
a large portion of real networks, as they are small-world (Watts and Strogatz
1998).

Since a graphlet contains nodes and edges, we propose a new graphlet-
based measure of the network position of an edge. We define edge-GDV

Fig. 2. All to 5-node graphlets. They contain 73 topologically unique ‘node orbits.’ In a graphlet, nodes in the same node orbit are of the same shade,
and 706 pathogen proteins from 193 pathogens.

Downloaded from http://bioinformatics.oxfordjournals.org/ by guest on July 8, 2015
to count the number of graphlets that an edge touches at a given "edge orbit" (Fig 4). We define edge orbits as follows. Given the automorphism group of graph X, Aut(X), if xy is an edge of X, then the edge orbit of xy is Orb(xy) = {zw | E(X) f (z) = f (w) for some f ∈ Aut(X)}, where E(X) is the set of edges of X. Alternatively, we can define edge orbits as follows. An edge-automorphism from graph X to graph Y is a bijection g of edges of X to edges of Y such that two edges xy and zw share a node in X if and only if edges g(xy) and g(zw) share a node in Y. An edge-automorphism is an edge-isomorphism from X to itself. The edge-automorphisms of X form the edge-automorphism group, Aut(X). If xy is an edge of X, then the edge orbit of xy is Orb(xy) = {zw | E(X) f (z) = f (w) for some f ∈ Aut(X)}, Independent of which of the two definitions we choose, the resulting edge orbits are the same. For example, in Fig 4a in a G1, both edges are in edge orbit 1. In a G2, all three edges are in edge orbit 2. In a G3, the two "outer" edges are in edge orbit 3, whereas the "middle" edge is in edge orbit 4, and so on. There are 68 edge orbits for 3 to 5-node graphlets. (We intentionally exclude edge orbit 0 in the only 2-node graphlet, G0, as each edge touches exactly one G0, i.e., itself.) Comparing edge-GDVs of two edges gives a sensitive measure of their topological similarity, since their extended network neighborhoods are compared. Using some existing measure, e.g., Euclidean distance, to compare edge-GDVs might be inappropriate, as some edge orbits are not independent. Instead, we design edge-GDV-similarity measure as follows. For an edge e, c, e is the kth element of its edge-GDV. The distance between the kth edge orbits of edges f and g is Dc(e,f) = | log(c, f) − log(c, g)| + | log(c, f) − log(c, g)|. This is applied to all edge orbits; the smaller the number of orbits that affect orbit i (including itself), c, the higher its weight wi, where wi = 1/| log(c, i) − log(68)|. Clearly, wi is in [0,1] and the highest weight is 1 assigned to orbit 1 with c = 68. The log is used in the formula for Dk, because the kth elements of two edge-GDVs can differ by several orders of magnitude and we do not want the distance between edge-GDVs to be dominated by large values; in addition, we want to account for the relative difference between c, e and f, and that is why we divide by the value of the denominator, which also scales Dk to [0,1]. The constants are added to Dk to be finite. The total distance is Dc(e,f) = E c, e Dc(e,f). Finally, edge-GDV-similarity is Sc(e,f) = 1 − Dc(e,f). It is in [0,1]. The higher the edge-GDV-similarity, the higher the topological similarity of edges' extended network neighborhoods. We design edge-GDV-similarity as described above because we already designed node-GDV-similarity, which compares node-GDVs, in a similar way (Milenkovic and Przulj 2004 and Milenkovic et al. 2007). Because we showed in different contexts that node-GDV-similarity successfully extracts function from network topology (Kuchaiev et al. 2009; Milenkovic et al. 2007), we expect edge-GDV-similarity to successfully extract function from topology as well.

2.4 Our clustering strategies
Clustering of the yeast PPI networks. We cluster the yeast PPI networks in the same manner as edge-SN, except that we use edge-GDV-similarity as the distance metrics instead of using the shared-neighborhood measure. Initially, for a fair comparison with edge-SN, we cluster adjacent edges only, to test if and how much we gain by using our more sensitive measure of edge similarity. Later on, we cluster all edges, to test if and how much we gain by taking into account edges that are not necessarily adjacent. Some further information is given below, after defining measures of partition quality.

Clustering of the human PPI network. We cluster the human PPI network to identify novel PI proteins from the clusters. To test if and how much we gain by clustering edges instead of nodes, we use edge-GDV-similarity as the distance metric to cluster edges, and we use node-GDV-similarity as the distance metric to cluster nodes. We do this for two popular clustering methods: hierarchical clustering and k-medoids clustering. With all four clustering strategies (hierarchical node clustering, hierarchical edge clustering, k-medoids node clustering and k-medoids edge clustering), we cluster nodes/edges independent on whether they are adjacent. When node- or edge-GDV-similarities are computed, all proteins and PPIs in the human network are considered. However, in the clustering process and in subsequent analyses of the clusters, we consider only nodes with more than three interacting partners in the network; consequently, we consider only edges with both end nodes having degree of more than three. We do this since poorly connected proteins are more likely to be involved in noisy PPIs. Similar was done previously (Yuan et al. 2004; Ide et al. 2004; Milenkovic et al. 2005).

In the network, there are 6121 proteins with degrees higher than three (of which 848 are PI proteins) and there are 47735 edges between these proteins. We form hierarchical tree as described above, clustering both adjacent and non-adjacent nodes/edges. We cluster non-adjacent nodes/edges because we hypothesize that function is encoded throughout the network, not just in dense connected local network regions. For this reason, choosing a partition with the maximum partition density might be inappropriate. Instead, we choose a partition as follows. We already clustered the human PPI network into hierarchical node clusters and tested many values for the desired number of clusters, KM = 100, 250, 500, 750, 1000, 1250, 1500, 1750, 2000, 2250 and 2500 (Milenkovic et al. 2007). KM of 1250 resulted in the best overall precision-recall (see Milenkovic et al. 2007 for details). Hence, we use KM = 1250 in this study for hierarchical node clustering. For a fair comparison of node and edge clustering, we use the same KM for hierarchical edge clustering. Using the same KM for node and edge clustering will tell us if we gain by allowing for the group overlap with edge clustering.

We form k-medoids (KM) nongnode clusters as follows. KM is a modification of the k-means algorithm that chooses actual data points as centers. We choose the value for the desired number of k-medoids clusters, KM, as follows. We already clustered the human PPI network into k-medoids node clusters and tested many values for KM = 100, 250, 500, 750, 1000, 1250, 1500, 1750, 2000, 2250 and 2500 (Milenkovic et al. 2005). The algorithm could not converge for KM of 1500 or higher, and it produced inconsistent clusters over multiple runs for KM of 750 or lower. Of the remaining two, KM = 1000 resulted in the best overall precision-recall (see Milenkovic et al. 2007 for details). Hence, we use KM = 1000 in this study for k-medoids node clustering. For a fair comparison of node and edge clustering, we use the same KM for k-medoids edge clustering.

2.5 Quality of partitions
Yeast. We evaluate a partition with respect to the same measures that were used by edge-SN: cluster coverage (CC), overlap coverage (OC), cluster quality (CQ) and overlap quality (OQ). CC is the fraction of nodes that belong to at least one ‘non-trivial’ cluster of three or more nodes. OC is the average number of non-trivial clusters that nodes belong to. CQ is the ratio of the average GO similarity over all node pairs that are in the same cluster and the average GO similarity over all node pairs in the network. OQ is the mutual information between the number of GO terms and the number of non-trivial clusters that proteins belong to. Raw values for the four measures do not necessarily fall in [0,1]. Hence, just as Yuan et al. (2005) we normalize each measure such that the best method has a value of one. Then, the overall partition quality is the sum of these four normalized measures, such that the maximum achievable score is four.

We can now note the following. To mimic the same measures, we would report the partition with maximum partition density D. However, we find that CC is strongly negatively correlated with CQ and OQ, and sometimes with OC, over all of our partitions (Supplementary Fig. S1). Thus, choosing the partition with low CC would result in high CQ and OQ (and sometimes OC), hence artificially increasing the overall partition quality. Since in three out of four yeast networks CC is lower for edge-SN than for the node clustering methods, it might not be surprising that edge-SN’s overall partition quality is the highest. Analogously, since edge-SN’s partitions with maximum D have lower CC than our partitions with maximum D, our partitions may
have lower overall partition quality simply because of the strong negative correlation between CC and other measures. Hence, we find the partition with maximum D among all partitions that have CC less than or equal to CC of edge-SN’s partition with maximum D. Then, we report either the partition obtained in this way or the partition with maximum D (independent of its CC), whichever has better overall partition quality. When we cluster both adjacent and non-adjacent edges, selecting the partition based on its density, as just described, might be inappropriate (see above). Thus, in this case, we also report the partition with the best overall partition quality.

Human. For each cluster, for each pathogen, we measure the enrichment of the cluster in proteins that interact with the pathogen. If the enrichment is statistically significant (see below), for each protein in the cluster, we use leave-one-out cross-validation (Sharan et al. 2003) by ‘hiding’ whether the protein is known to interact with the pathogen and predicting it as interacting with the pathogen if the enrichment of the cluster is above a given threshold k. We vary k from 0% to 100%, in increments of 1%. For each k, we evaluate the prediction accuracy through precision (a measure of exactness) and recall (a measure of completeness), combined into F-score. Given our predictions, precision is the number of true positives out of both true positives and false positives, recall is the number of true positives out of both true positives and false negatives, and F-score = 2 Precision × Recall / Precision + Recall. We compute F-scores for all four clustering methods over the entire range for k. As a combination of precision and recall, F-score makes method comparison easier.

Statistical significance. When we compute enrichments, we consider as PI proteins all human proteins (with degrees above three; see above) that interact with a pathogen that interacts with at least one other human protein (with degree above three). Hence, we consider only pathogens that interact with at least two human proteins. If we considered a pathogen that interacts with only one protein, we could never predict other proteins to interact with that pathogen, since we can measure enrichment only when at least two proteins in the cluster interact with the pathogen; but this one protein would be lowering recall as a false positive.) There are 936 such PI proteins.

We compute p-value of a given enrichment as follows. There are |N| PI proteins (as just defined) in the network, |P| out of |N| PI proteins interact with a given pathogen, |C| out of |N| PI proteins are in a given cluster, |P| out of |C| proteins in the cluster interact with the given pathogen. The enrichment is |P|/|C|. The p-value is obtained by the hypergeometric model for sampling without replacement: p-value = 1 − \sum_{i=0}^{k} \binom{\text{PI}}{i} \binom{\text{non-PI}}{\text{C}-i} / \binom{\text{PI}+\text{non-PI}}{\text{C}}.

An enrichment with p-value below 0.05 is statistically significant. We assess the statistical significance of observing F-scores computed from data clusters by comparing them with F-scores computed from random clusters. We create random clusters by randomly assigning nodes to clusters of the same size as the data clusters. We compute F-scores on such randomized clusters as described above, averaged over 100 randomization runs.

3 RESULTS AND DISCUSSION

We cluster edge-GDV-similar edges in PPI networks. We compare our approach with other methods (Section 3.1). Then, we use it to identify new pathogen-interacting (PI) proteins (Section 3.2).

3.1 Comparison with other methods on yeast

We evaluate three existing node clustering methods and one existing edge clustering method against three versions of our method (Table 1) on four yeast PPI networks (Y2H, AP/MS, LC and ALL), with respect to four partition quality measures (cluster coverage – CC, overlap coverage – OC, cluster quality – CQ and overlap quality – OQ) that are combined into the normalized overall partition quality; see Section 2. Results are shown in Fig. 1 for AP/MS network and in Supplementary Fig. S2 for Y2H, LC and ALL networks.

Fig. 3. Method comparison for AP/MS yeast PPI network.

We gain by using edge-GDV-similarity for clustering: eGDV-A-D outperforms all node clustering approaches on all networks. (This includes node clustering by using node-GDV-similarity, as shown in Supplementary Fig. S1). Also, it outperforms edge-SN on Y2H and AP/MS. Although edge-SN is slightly better than and comparable to eGDV-A-D on LC and ALL networks, respectively, eGDV-NA-D outperforms edge-SN on these two networks, as well as on AP/MS. Hence, we gain further by clustering non-adjacent edges in addition to adjacent ones. The only exception is Y2H, for which edge-SN is slightly better than eGDV-NA-D. However, as already noted, eGDV-A-D outperforms edge-SN on Y2H network. Hence, we are always superior, with either eGDV-A-D or eGDV-NA-D or both eGDV-A-D and eGDV-NA-D. With eGDV-NA-B, we further demonstrate our superiority over all other methods on all networks.

3.2 Prediction of pathogen-interacting human proteins

Motivation. Of all pairs of PI human proteins that both interact with the same pathogen, 96.8% are not adjacent in the network: 3.2%, 25.8%, 55.8%, 14.8% and 0.4% of them are at the shortest path distance of 1, 2, 3, 4 and 5, respectively. In addition, topologies around PI proteins that interact with the same pathogen are different than topologies around PI proteins that interact with different pathogens, as well as than topologies around non-PI proteins. That is, PI proteins that interact with the same pathogen are more node-GDV-similar to each other than to PI proteins interacting with different pathogens or to non-PI proteins (Supplementary Section S1). Thus, we cluster topologically similar but not necessarily adjacent network regions to predict new PI proteins from clusters enriched in known PI proteins. To study how the choice of the clustering method affects the predictions, we cluster both edge-GDV-similar edges and node-GDV-similar nodes, with both hierarchical and k-medoids clustering. We denote hierarchical node clustering as node-HIE, k-medoids node clustering as node-KM, hierarchical edge clustering as edge-HIE and k-medoids edge clustering as edge-KM.

Cluster properties. Cluster sizes, numbers of connected components, and average shortest path distances are shown in Supplementary Fig. S3. With node-HIE, cluster sizes follow a ‘power-law’: many nodes are in small clusters (e.g., 1144 (18.7%) of the nodes are in trivial clusters of size 1 or 2), but there exist some large clusters (e.g. there are clusters with more than 200 nodes). On the other hand, with node-KM, only 121 (2%) of the nodes are in trivial clusters, the majority of clusters have size 3–20, only two have size 21–50, and there are no larger clusters. With edge-HIE and
Table 1. Different clustering approaches evaluated in this study on the yeast PPI networks

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CliqPerc</td>
<td>Cliques percolation (Palla et al., 2005)</td>
</tr>
<tr>
<td>GreedMod</td>
<td>Greedy modularity optimization (Newman, 2004)</td>
</tr>
<tr>
<td>Infomap</td>
<td>Infomap (Rosvall and Bergstrom, 2008)</td>
</tr>
<tr>
<td>Edge-SN</td>
<td>Edge - shared neighborhood (Alm et al., 2010)</td>
</tr>
<tr>
<td>cGDV-NA-D</td>
<td>Our method when clustering adjacent and non-adjacent edges and reporting the partition with the maximum density</td>
</tr>
<tr>
<td>eGDV-NA-B</td>
<td>Our method when clustering adjacent and non-adjacent edges and reporting the partition with the best overall quality</td>
</tr>
</tbody>
</table>

CliqPerc, GreedMod and Infomap are existing node clustering approaches. Edge-SN is an existing edge clustering approach. See Section 2 for details.

Edge clusters reveal pathogen-interacting proteins

Since each clustering method has its (dis)advantages, the choice of the most appropriate method is application-dependent. Hence, instead of producing predictions only with the most accurate method (edge-HIE), we produce predictions with each of the methods, except node-HIE, which shows random-like behavior. Since we predict a protein to interact with a pathogen if the enrichment of its cluster is above a given k, the choice of k is crucial. We decide to use as high k as possible while decreasing F-score as little as possible. We choose k of 66%, since for all three clustering methods, F-scores increase or remain relatively "uniform" up to k = 66% but then start decreasing. At this k, F-scores are 23%, 17% and 16% for edge-HIE, node-KM and edge-KM, respectively. These translate into precision of 18% at recall of 30% for edge-HIE, precision of 36% at recall of 11% for node-KM and precision of 16% at recall of 16% for edge-KM. Hence, precision is higher for node-KM than for edge-HIE and edge-KM. However, edge-HIE and edge-KM have higher recall, which might not be surprising, given that they allow for the node overlap between clusters.

Relatively lower F-scores should not be alarming for the following reasons. Since the number of known PI proteins will increase in the future, precision, recall and F-scores are likely to increase as well. Our predictions are already statistically significant; since we removed clusters with random enrichment in PI proteins, the number of possible predictions is automatically decreased, thus decreasing the F-scores. F-scores are higher for data clusters than for randomized clusters; for k = 66% at which we make predictions, data F-scores of 23%, 17% and 16% for edge-HIE, node-KM and edge-KM, respectively. Hence, edge-HIE and edge-KM are higher than those of 17%, 6% and 12% for their randomized counterparts, respectively. We demonstrate the superiority of our approach over others (Section 2). Finally, our clusters are enriched in biological pathways, unlike their random counterparts, which further validates our approach (Supplementary Section S2).
4 CONCLUSION

We introduce edge-GDV-similarity, a sensitive topological measure of edge similarity. When we hierarchically cluster edge-GDV-similar edges in yeast networks, we outperform existing node and edge clustering methods. When we cluster the human network, edge-HIE is better than node-HIE, while edge-KM and node-KM are comparable. Thus, we gain by clustering edges compared to edge clustering methods. When we cluster the human network, of edge similarity. When we hierarchically cluster edge-GDV-search, which confirms the correctness of our approach.

Funding: This work was supported by the National Science Foundation CCF-1243295 grant.

REFERENCES


Supplementary Table S1 lists protein–pathogen associations predicted by edge-HIE, edge-KM and node-KM. Supplementary Fig. S7 shows the overlap of their predictions. Together, they predict 1677 out of 3728 known protein–pathogen associations, resulting in combined recall of 45%. They make the total of 10 190 predictions. Precision is 14% for predictions produced by a single method, 41% for predictions produced by any two methods and 50% for predictions produced by all three methods. Since precision is higher for predictions produced by multiple methods, they could be considered of higher confidence. A small overlap of all three methods (54 predictions) confirms that there is no single ‘best’ method and justifies our decision to use all three methods to produce predictions.

Literature validation. We perform literature search for ‘high-scoring’ predictions selected as follows. Of all new predictions, we exclude HIV-related predictions, which would, due to their large number, make literature search difficult. We also exclude predictions involving proteins that could not be found in any article. Of the remaining predictions, we focus on those with cluster enrichment of 100%, resulting in 18 predictions. We validate 44% of the predictions. We link TGFBR2 with vaccinia virus. TGFBR2 transduces TGFβ1 signal from the cell surface to the cytoplasm, thus regulating genome replication of influenza virus (PMID: 17932485). In addition, IL-21 protein, which is related to upregulation of BCL-2 molecules, is critical for response to vaccinia viral infection (PMID: 21257966). We link MCM complex components MCM6, MCM7 and MCM8 with influenza A virus. MCM complex has a role in regulating genome replication of influenza virus (PMID: 17932485). We link NCOA6 with herpes simplex virus. It harbors a potent N-terminal activation domain, which is as active as the herpes simplex terminal activation domain, which is as active as the herpes simplex


Copied by: MANUSCRIPT CATEGORY: ECCB

R.W.Solava et al.

[5:30 7/9/2015 Bioinformatics-bits76.tex] Page: i486 i480–i486