# Local Community Detection in Protein Interaction Networks

Jack G. LeBien, Dept. of Physics, University of New Orleans

*Abstract*—**Protein-protein interactions are important to the biological and medical research communities as their properties can provide insight into cellular and molecular processes. Extensive protein classification and interaction detection records have allowed for the creation of interaction databases and network construction. In these graphs, sub-communities can correspond to functional protein complexes. Attempts to study PPI community structure have often used global clustering algorithms which partition the entire network. However, observation of the local community structure of these networks can provide insight as to how functional modules interact and relate to larger scale processes, as well as reducing computational cost. In the present study, three local community detection algorithms are evaluated as applied to protein interaction network data from varying model organisms and experimental systems.**

***Index Terms*—**Local community detection, protein-protein interaction network, random walk, spectral clustering, seed set expansion

## I. Introduction

COMMunity detection is a ubiquitous problem in graph analysis, as the small-scale structure of dynamical systems generally governs large scale processes. The definition of a community is not absolute. An appropriate metric for community classification depends on the application. A trivial method is to search for fully connected subgraphs (cliques), however this has been shown to be infeasible for large networks: finding the size of the largest clique in a graph is an NP-Complete problem. Thus the search for sub-communities has naturally expanded to consider more generally dense components. Partitioning algorithms have been developed for application to network data which aim to classify node clusters and minimize the number of inter-cluster links. Spectral graph theory addresses the relationship between graph structure and the eigenvectors and eigenvalues of its associated matrices such as the adjacency or Laplacian matrix. Spectral analysis of adjacency matrices has been used to find clique-like components, and the eigenvectors of the graph Laplacian are often used in spectral graph clustering algorithms.

Protein-protein interactions are of significant interest to the biological and medical research communities as their properties govern cellular and molecular processes. Extensive protein classification and interaction detection records have allowed for the creation of interaction databases and network construction. In these graphs, sub-communities can correspond to functional protein complexes. Previously unknown functionalities between protein are often revealed by the identification of communities in PPI networks. Attempts to study PPI community structure have often used global clustering algorithms which partition the entire network. However, observation of the local community structure of these networks can provide insight as to how functional modules interact and relate to larger scale processes. It is therefore of interest to test algorithms that can account for overlapping communities and efficiently find local communities provided a query protein or selection of proteins.

Three local community detection algorithms are evaluated as applied to protein networks. These algorithms are LEMON [1], Nibble [2], and PageRank-Nibble [3]. Each takes an input of a set of query nodes, along with the graph data, and outputs a set of nodes that optimize a cost function. Multiple community membership can be revealed by varying the seed set but maintaining one or several nodes. Each also use random walks to reveal the local structure of the graph, although use differing mining techniques. An overview of the algorithms is given below. For this study, seed sets contained a single protein. The process of identifying a cluster by growing an initial seed set is referred to as seed set expansion.

A widely cited study by Voevodski et al [4] reported that Nibble and PageRank-Nibble return sets of functionally coherent interacting proteins. The functional coherence is derived from a functional distance measure proposed by Yu, Jansen, et al [5]. Data from the Gene Ontology Process classification project was used to derive the distances, in which proteins are leaf nodes of a directed acyclic graph. Consecutive levels of the tree above the terminal nodes correspond to functional annotations of varying levels of specialization. The distance between a pair of proteins is defined as the number of gene pairs that share the least common ancestors of the pair in the classification hierarchy. To calculate the functional coherence of a subset of the protein network, absolute and relative measures were defined. The absolute functional coherence is calculated as the difference between the average pairwise functional distance of the entire network and that of the community. The relative measure takes the difference in average pairwise distances of proteins not within the community, and those within. In [4], it was also shown that both measures of functional coherence had a statistically significant relationship with the conductance of the detected communities. For this reason, the conductance was the chosen metric for evaluating community quality. In order to verify this correlation, they randomly selected vertices, and then chose $k - 1$ of its nearest neighbors to be the evaluated group. Randomly selected groups would likely have very poor conductance and functional coherence scores, and of course they should not be selected by algorithms that minimize conductance.

Network data was obtained from the BioGRID [6] database of protein, chemical, and genetic interactions. The database currently searches over fifty thousand publications for verified interactions, and chemical associations.

## II. Methods

The performance of three algorithms was evaluated on the BioGRID protein interaction network. Each uses a random walk to reveal the local structure of the network. At step $k$ of a random walk, the probability distribution spreads to nodes a distance of $k$ hops away. For weighted graphs, the probability typically depends on edge weights, however for an unweighted graph it only depends on node degree. Therefore, the random walk matrix which iterates the spreading of the probability distribution can be defined

$$W_G = D_G^{-1} A_G$$

where $D_G$ and $A_G$ are the degree and adjacency matrices, respectively.

Each algorithm also uses the conductance as an optimization metric. Conductance is a widely used measure of community quality. Given a graph $G = (V, E)$ and a subset of vertices $S \in V$, consider the edge boundary of $S$

$$\delta(S) = \{\{x, y\} \in E \mid x \in S, y \notin S\}$$

That is, the collection of edges with exactly one point in $S$. Let us call the volume of $S$ the sum of the degrees of its nodes:

$$vol(S) = \sum_{x \in S} d(x)$$

The conductance is then defined

$$\Phi(S) = \frac{|\delta(S)|}{\min(\mathrm{vol}(S), \mathrm{vol}(\bar{S}))}$$

By this definition, a cluster can have low conductance without being dense.

### A. LEMON

The LEMON method proposed in [1] uses a local spectral approach. As mentioned, PPI data has largely been analyzed using global clustering methods, in which the entire set is partitioned. Local techniques determine cluster topology near a set of seed or query nodes. LEMON takes inspiration from spectral clustering methods to detect local communities. Spectral methods use eigenvectors and eigenvalues of the graph Laplacian. The Laplacian matrix of an undirected graph $G = (V, E)$ can be defined

$$L_G(u, v) = \left\{ \begin{matrix} d(u) & \text{if } u = v \\ -1 & \text{if } u \neq v \text{ and } (u, v) \in E \\ 0 & \text{otherwise} \end{matrix} \right\}$$

$$= D_G - A_G$$

In traditional spectral clustering methods, the first several $(d)$ singular vectors of the graph Laplacian are found and used to create a $n \times d$ matrix as a latent space (for $G$ with $n$ vertices) in which vertices are clustered using some method such as $k$-means. These methods have been shown to perform very well on clusters that have complications such as concave boundaries in the original feature space.

The LEMON algorithm deviates from this approach by calculating a basis of the matrix formed by the concatenation of the random walk probability distribution at several steps. That is, if $P_n$ is a column vector of the network probability distribution at step $n$ of the random walk, the algorithm calculates an orthonormal basis of the matrix

$$P_{nk} = [P_n, P_{n+s}, \dots, P_{n+s(k-1)}]$$

where $s$ is the random walk step interval. From this basis, community members are chosen by a linear programming problem to select rows of the basis that are most similar to those of the seed members. It was determined by Li et al. that a subspace dimension of three and a random-walk length of three-steps was appropriate for real and synthetic data.

The effect of adjusting several parameters on the performance of LEMON was observed. Results were consistent with those mentioned in the original paper, a random walk step interval of 3, and a subspace dimension of 3 were used.

In order to evaluate their algorithm, the authors in [1] use data with manually verified ground-truth communities. They are thus able to use a more direct quality metric. They adopt the F1-score which can be defined as follows

$$F_1(C, C^*) = \frac{2Precision(C, C^*)Recall(C, C^*)}{Precision(C, C^*) + Recall(C, C^*)}$$

where the precision and recall are defined as

$$Precision(C, C^*) = \frac{|C \cap C^*|}{|C|}$$

$$Recall(C, C^*) = \frac{|C \cap C^*|}{|C^*|}$$

A novel subroutine of the LEMON algorithm to increase its scalability is using a random walk to sample the graph. The method is based on the assumption that community members will likely be near the seed members, typically only a few

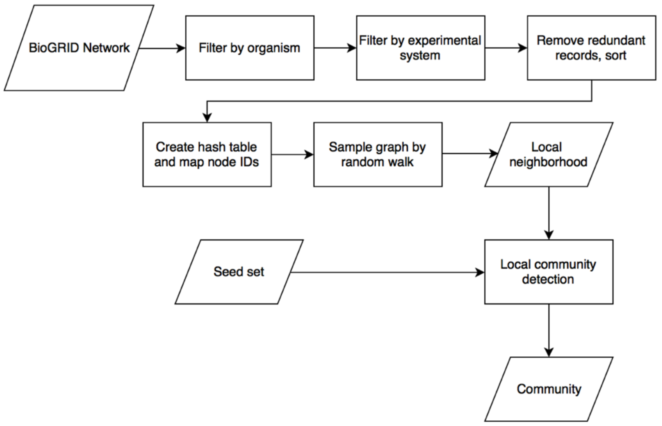
BioGRID Network
Filter by organism
Filter by experimental system
Remove redundant records, sort
Create hash table and map node IDs
Sample graph by random walk
Local neighborhood
Seed set
Local community detection
Community

Fig. 1. The data flow model used for collecting results. At the local community detection steps, the varying algorithms are tested.

steps away. Thus sampling is done by conducting a random walk until the probability distribution has spread to $\alpha|C|_{avg}$ vertices, where $\alpha$ is some constant, and $|C|_{avg}$ is the estimated average community size in the graph. The constant $\alpha$ can be chosen by some evaluation, and should be large enough to avoid under-sampling but small enough to avoid unreasonable computational expenses. If a community does exist for the seed members, as explained in [7] it should serve as a bottleneck for the spread of probability. For this reason, a random walk is utilized for sampling as opposed to a method such as breadth-first search.

In this study, community detection algorithms were applied to graphs sampled by this sampling routine detailed in [1]. IT is shown that the Nibble and PageRank-Nibble succeed in yielding low conductance communities from subgraphs sampled by this method.

### *B. Nibble*

The Nibble algorithm is local clustering algorithm developed by Spielman and Teng [2]. It uses a lazy random walk transition probability matrix. The lazy walk incorporates the possibility of remaining at the current vertex with probability ½. The purpose of this is to support the formation of a steady state of the walk. For each step of the random walk, the algorithm rounds nodes with probability less than a chosen threshold to 0. It then searches for a subset of the nodes with non-zero probability that yields low conductance.

To perform this "sweep" of the probability distribution, the vertices are ordered by degree-normalized probability, and the conductance of the first $j$ vertices is computed. The parameter $j$ is varied from one to the number of non-zero entries of the distribution. The subset with the lowest conductance is returned. More precisely, if $S_j^P$ is the set of the highest $j$ members of the ordered distribution $P$, consider a collection of sets $S^P = \{S_1, \dots, S_j\}$. Then the algorithm will return

$$\Phi(P) = \min_{j\in[1,N]} \Phi(S_j^P)$$

where $\Phi(S)$ is the conductance of the subgraph $S$.

### *C. PageRank-Nibble*

A variation of Spielman and Teng's algorithm was proposed by Andersen, Chung, and Lang [3]. In this algorithm a PageRank vector was used to defined nearness rather than a probability vector. Taking a single starting node, the personalized PageRank vector gives the stationary distribution of a random walk that will return to the starting node. The same sweep technique detailed for Nibble is applied to the calculated PageRank vector to return the cluster of lowest conductance. PageRank-Nibble requires that the output cluster contains the starting vertex, as it will have the highest probability in the computed PageRank vector.

The PageRank algorithm incorporates a constant $\alpha$ in the range $(0,1]$ referred to as the teleportation constant. This dictates the probability of jumping to a random vertex. The personalized PageRank vector refers to the use of a lazy random walk transition matrix. In [3] it is demonstrated that this is form of the algorithm is equivalent to the traditional PageRank up to a change in $\alpha$. A limit on the accuracy is also an input, and for the current study, a limit of 0.5 was put on the minimizing function, which is the conductance.

## III. Data

The Database of Interacting Proteins (DIP) was the first of an increasing number of public PPI databases, including BioGRID, STRING, and BIND. Methods used for data collection can be categorized into three general types. Primary databases collect interaction data verified by small or large scale experimental methods such as yeast two-hybrid screening or affinity capture, whereas meta-databases integrate primary database information. Prediction databases such as STRING incorporate machine learning and pattern recognition methods to identify probable protein pairs. For this proposal the BioGRID repository is of interest. The entire dataset is available in several compressed formats, with file structure options for the intended analysis. Interaction data filed by experimental system is of interest in this study, as it will strongly influence the network topology. Affinity capture experiments tend to yield networks with a stronger community structure as they verify multiple interactions for a single protein per trial. This is done by attracting interactive proteins using a single "bait" protein. Two-hybrid screening experiment datasets tend to be sparser as there are not multiple interactions verified in one trial.

For this study, networks generated from BioGRID containing protein interactions within several model species (*Drosophila Melanogaster, Saccharomyces Cerevisiae, Homo Sapiens)* verified by several experimental systems (Affinity Capture-MS, Affinity Capture-Western, and Two-hybrid) are considered. This evaluation can reveal dependencies of the algorithms to varying properties of the data. Figure 1 depicts the flow of data from the full BioGRID network to the subgraph sampled by random walk, and finally the detected community. The algorithms are applied to each local neighborhood generated by a seed.

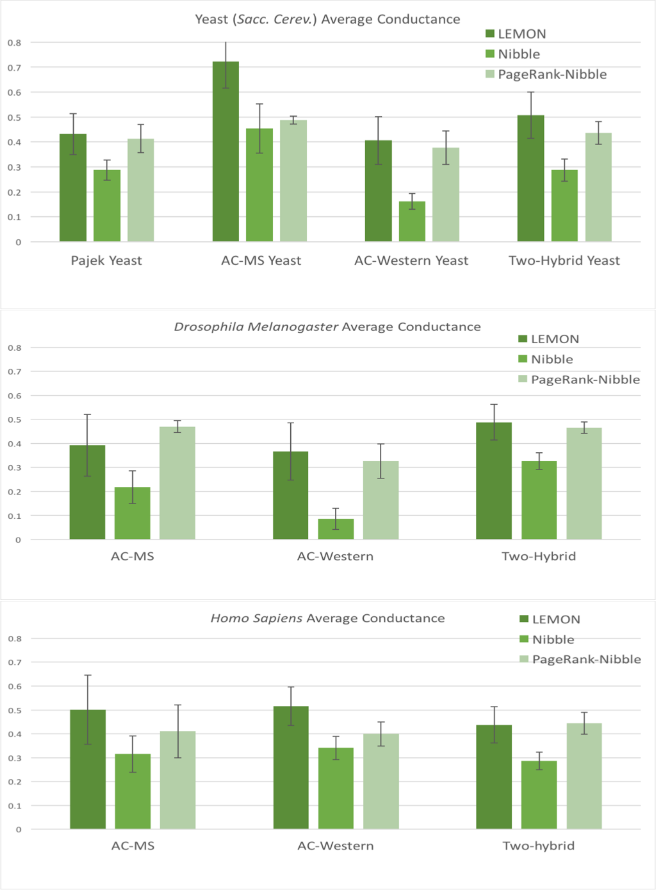

Fig. 2. Conductance results for the tested algorithms. Each cluster of bars represents a generated subgraph from the BioGRID network. Network data for protein interactions in Baker's yeast were made available by Pajek, and were also included the evaluation for that organism. Error bars show the standard deviation of the fifty trials.

## IV. Results and Analysis

To evaluate the performance of each algorithm, fifty nodes were randomly sampled from each generated subgraph (local neighborhood, Fig. 1). Community sizes were bound to with the range of 10 to 100. As mentioned in [4], biologically relevant communities are expected to be within this range. The conductances of the returned communities were compared. The results show that the Nibble algorithm consistently yields the lowest conductance communities. These results support those found in [4] that Nibble is applicable to PPI network research.

As shown in the table in figure two, the *Homo Sapiens* has the greatest number of verified interactions for each network, followed by *Drosophila* save for its Affinity Capture-Western network, which is the smallest network considered. It is apparent that the low node to edge ratio in AC-MS network for Yeast caused a decline in the performance of the algorithms. The highest node to edge ratio of AC-W for *Drosophila* also corresponds to the lowest average conductance of any network-algorithm pair.

The detected communities by LEMON and Nibble have a typical size of 50 nodes. This is slightly higher than the expected range of 10 – 40 mentioned in [4]. PageRank-Nibble returned communities of sizes with a higher variance.

| | LEMON | Nibble | PR-Nibble |
|---|---|---|---|
| SC – Pajek | 36 | 58 | 57 |
| SC – AC-MS | 66 | 71 | 99 |
| SC – AC-W | 33 | 66 | 18 |
| SC – 2H | 41 | 58 | 42 |
| DM – AC-MS | 51 | 56 | 99 |
| DM – AC-W | 52 | 56 | 20 |
| DM – 2H | 42 | 49 | 24 |
| HS – AC-MS | 68 | 70 | 42 |
| HS – AC-W | 46 | 63 | 37 |
| HS – 2H | 40 | 60 | 31 |

| | Nodes | Edges | Ratio (N/E) |
|---|---|---|---|
| SC – Pajek | 2284 | 6646 | 0.343665363 |
| SC – AC-MS | 4656 | 50653 | 0.091919531 |
| SC – AC-W | 3172 | 11814 | 0.268495006 |
| SC – 2H | 3653 | 13151 | 0.277773553 |
| DM – AC-MS | 2982 | 13103 | 0.22758147 |
| DM – AC-W | 681 | 917 | 0.74263904 |
| DM – 2H | 7239 | 23516 | 0.307832965 |
| HS – AC-MS | 13956 | 119169 | 0.117110994 |
| HS – AC-W | 8187 | 41093 | 0.199231013 |
| HS – 2H | 11089 | 45569 | 0.243345257 |

Fig. 3. Average community sizes (above) of the algorithms detected from each network, and network sizes before sampling by random walk (below), with node-to-edge ratios.

Without the sampling subroutine, Nibble and PageRank-Nibble detected communities in one or two minutes. On the sampled graphs, communities were returned in 10-20 seconds. A thorough comparison of performance with and without the sampling routine was not made. However, significantly low conductance communities are shown to be revealed consistently by the algorithms on the sampled networks.

As shown in the Figure 2, the average conductance for all algorithms is below 0.5 for each network tested, save for the AC-MS Yeast network with the lowest node-to-edge ratio. As expected, there is a general decrease in performance between the AC-MS and Two-hybrid networks of *Drosophila*, despite a similar node-to-edge ratio, likely due to the nature of experimental system.

LEMON outperformed PageRank-Nibble for the Two-hybrid network of *Homo Sapiens* as well as the AC-MS *Drosophila* network, however it otherwise had the poorest performance. This suggests that sweeping method of Nibble and PageRank-Nibble is more highly applicable to these types of networks than the optimization by linear programming performed on the subspace calculated in LEMON.

Degree distributions of the networks were also observed for trends. The distributions generally follow a power-law, with varying levels of deviation at the lower end of the degree range. An increase in variance and average value in the tails of the distributions between the *Drosophila*, and *Homo sapiens* networks seems seem to correspond with an increase of average conductance yielded by LEMON and Nibble. This distribution property implies there to be a small number of nodes with unusually high degree, which could disrupt the convergence of a random walk if encountered, by spreading the probability into a functionally unrelated region. An observation of between-ness centrality of these nodes could reveal if they are links between components of differing functionality. The same property of a wide distribution tail is seen for the AC-MS Yeast network which yielded the poorest conductance.

## V. Conclusion

The optimization methods of Nibble and PageRank-Nibble appear to be more appropriate to the application of protein interaction networks than that used in LEMON. However, the applicability of LEMON to co-purchase and social networks has been demonstrated in [1], suggesting that the structure of the graphs generated by PPI data affects the performance. Spectral methods may not be necessary. There appears to be a positive correlation between the node-to-edge ratio and the detection performance. The stability of a random walk would be expected to decline as this ratio decreases, because local neighborhoods are obscured and there are more opportunities to hop to a functionally distant vertex. Nibble performed better than the other algorithms in every case.

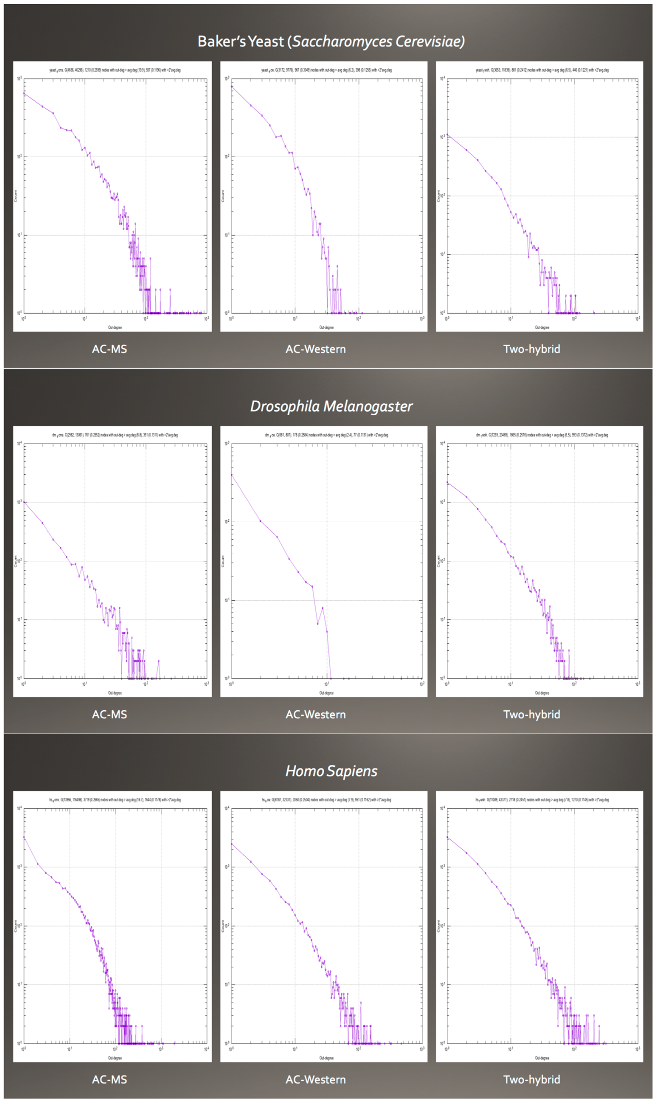

Fig. 4. Degree distributions for each network tested, save the Pajek Yeast network.

The sampling method proposed in [1] appears to be applicable to the other algorithms. A thorough investigation of performance with and without the sampling routine still remains to be undertaken. Another consideration for future study is the effect of seed set property on the results, using seed sets of size greater than one.

## References

[1] Y. Li, K. He, D. Bindel, and J. Hopcroft. Uncovering the small community structure in large networks: A local spectral approach. In *WWW*, pages 658–668, 2015.

[2] Daniel A. Spielman and Shang-Hua Teng. A local clustering algorithm for massive graphs and its application to nearly-linear time graph partitioning. CoRR, abs/0809.3232, 2008. Available at http://arxiv.org/abs/0809.3232. Submitted to SICOMP.

[3] R. Andersen, F. Chung, and K. Lang. Local graph partitioning using PageRank vectors. In *FOCS '06: Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, pages 475–486, 2006.

[4] Konstantin Voevodski, Shang-Hua Teng, and Yu Xia. Finding local communities in protein networks. BMC Bioinformatics, 10(1):297, 2009.

[5] Yu H, Jansen R, Gerstein M: Developing a similarity measure in biological function space. Bioinformatics 2007, 23: 2163–2173. 10.1093/bioinformatics/btm291

[6] Stark C, Breitkreutz BJ, Reguly T, Boucher L, Breitkreutz A, Tyers M: BioGRID: a general repository for interaction datasets. Nucleic Acids Res 2006, 34: D535–9. 10.1093/nar/gkj109

[7] R. Andersen and K. J. Lang. Communities from seed sets. In WWW, pages 223–232. ACM, 2006.

[8] Marcus T. Dittrich, Gunnar W. Klau, Andreas Rosenwald, Thomas Dandekar, Tobias Müller; "Identifying functional modules in protein–protein interaction networks: an integrated exact approach." *Bioinformatics* 2008

[9] J. Leskovec, K. Lang, and M. Mahoney. Empirical compari- son of algorithms for network community detection. In *WWW '10*, 2010.

[10] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney. Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *Internet Mathematics*, 6(1):29–123, 2009.

[11] M. Girvan and M. Newman. Community structure in social and biological networks. *PNAS*, 99(12):7821–7826, 2002.

[12] J. Yang and J. Leskovec. Defining and Evaluating Network Communities based on Ground-truth. Extended version, 2012.