BIOINFORMATICS ORIGINAL PAPER

Vol. 27 no. 3 2011, pages 326–333
doi:10.1093/bioinformatics/btq655

Sequence analysis

Advance Access publication November 29, 2010

Improving the quality of protein similarity network clustering algorithms using the network edge weight distribution

Leonard Apeltsin[1], John H. Morris[1], Patricia C. Babbitt[1,2] and Thomas E. Ferrin[1,2,*]

[1]Department of Pharmaceutical Chemistry and [2]Department of Bioengineering and Therapeutic Sciences, University of California, San Francisco, CA, USA

Associate Editor: Burkhard Rost

ABSTRACT

Motivation: Clustering protein sequence data into functionally specific families is a difficult but important problem in biological research. One useful approach for tackling this problem involves representing the sequence dataset as a protein similarity network, and afterwards clustering the network using advanced graph analysis techniques. Although a multitude of such network clustering algorithms have been developed over the past few years, comparing algorithms is often difficult because performance is affected by the specifics of network construction. We investigate an important aspect of network construction used in analyzing protein superfamilies and present a heuristic approach for improving the performance of several algorithms.

Results: We analyzed how the performance of network clustering algorithms relates to thresholding the network prior to clustering. Our results, over four different datasets, show how for each input dataset there exists an optimal threshold range over which an algorithm generates its most accurate clustering output. Our results further show how the optimal threshold range correlates with the shape of the edge weight distribution for the input similarity network. We used this correlation to develop an automated threshold selection heuristic in order to most optimally filter a similarity network prior to clustering. This heuristic allows researchers to process their protein datasets with runtime efficient network clustering algorithms without sacrificing the clustering accuracy of the final results.

Availability: Python code for implementing the automated threshold selection heuristic, together with the datasets used in our analysis, are available at http://www.rbvi.ucsf.edu/Research/cytoscape/threshold_scripts.zip.

Contact: tef@cgl.ucsf.edu

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

Received on June 14, 2010; revised on November 19, 2010; accepted on November 22, 2010

1 INTRODUCTION

In the last decade, there has been an explosion in the available protein sequence data. Currently, the Uniprot database contains approximately 11 million protein sequences and is growing exponentially (Apweiler *et al.,* 2004); a very large proportion of these proteins have not been experimentally characterized. Computational clustering approaches can provide an important means to deciphering the functions of these uncharacterized proteins in an efficient way. Recent efforts in this area, discussed below, have focused on developing and testing algorithms for clustering proteins by functional similarity based only on sequence data. These algorithms go beyond traditional clustering approaches, such as hierarchical and *k*-means, which require advance knowledge approximating the number of functional groups present in order to either cluster effectively or to interpret clustering output correctly. Rather, these algorithms rely on the network properties of a protein sequence dataset to cluster the data into functional groups without any prior knowledge of the group identities (Schaeffer, 2007).

Network clustering algorithms take as input a protein similarity graph (Noble *et al.,* 2005). Vertices in the graph represent individual proteins, while edges represent the pairwise sequence similarities between the proteins. Often, BLAST (Altschul *et al*., 1997) scores are used as edge weights. Subsequent to input, the similarity graph is processed by the network clustering algorithm to identify distinct groups of nodes in the graph that in many cases correspond to groups of proteins that share the same function.

How the similarity graphs are processed varies with each clustering algorithm. In general, most network clustering approaches may be assigned to one of two categories; geometry-based and flow-based (Frivolt and Pok, 2006). Geometry-based approaches, such as Force (Wittkop *et al*., 2007), Regularized Kernel Estimation (Lu *et al*., 2005), spectral clustering (Paccanaro *et al*., 2006) and TransClust (Wittkop *et al.*, 2010) embed the protein graph into high-dimensional space and then group the nodes into clusters based on spatial proximity. Flow-based approaches, such as the Markov Clustering Algorithm (MCL; Enright *et al*., 2002) and Affinity Propagation (Frey and Dueck, 2007) model the possible flow of information between nodes based on edge weight. How the information congregates across groups of nodes then determines the final output of clusters.

The differences between these two categorizes of algorithms reflect a difference in performance. Geometry-based approaches such as Force rely on non-linear calculations between pairwise elements in the similarity graph, leading to potentially long execution times. Flow-based approaches such as MCL rely on simple matrix and vector multiplication, which leads to relatively short execution times. However, it has been shown that Force outperforms MCL for certain similarity graphs (Wittkop *et al*., 2007), making the hours to seconds difference in run times a worthwhile performance trade-off.

[*]To whom correspondence should be addressed.

While comparative performance of various network clustering algorithms has been examined in great detail in the literature, there remains a property of network clustering that warrants additional investigation. The protein similarity graphs themselves are treated as static objects when used as input to the algorithms. These graphs, however, are not static but rather exhibit a variety of dynamic properties when studied over a series of different edge weights (Atkinson *et al*., 2009). As the threshold for allowing edge weight inclusion is adjusted, the similarity graphs break and regroup into varying representations of protein similarity when visualized using an edge weighted network layout algorithm (Fruchterman and Rheingold, 1991). By viewing the graph behavior over a range of thresholds with a network visualization tool such as Cytoscape (Shannon *et al*., 2003) or BioLayout (Enright and Ouzounis, 2001), a researcher may observe degrees of protein similarity that are not visible in the complete, unthresholded graph. In other words, given a graph, one particular threshold may be more optimal for analysis than another.

It was our goal to examine how dynamic graph thresholding relates to the various network clustering approaches. We set out to answer a number of important questions. Given a protein similarity graph and a network clustering algorithm, does a threshold exist at which network clustering performance is optimal? If so, how does the optimal threshold vary across different graphs and different categories of network clustering algorithms? Given an uncharacterized protein similarity graph, can we estimate the optimal edge weight threshold from the properties of the graph itself, prior to clustering?

We used two representative and well-studied network clustering algorithms for our analysis, Force and MCL. Our results are somewhat surprising. For any of our four datasets (see below) and the two network clustering algorithms, there is a range of thresholds over which algorithm performance will be near optimal. This threshold range does not necessarily include zero, the threshold at which the graph remains completely unfiltered. More importantly, our research shows that the optimal threshold range for any given similarity graph is dependent on the edge weight distribution across that graph. These findings allowed us to test a heuristic for estimating thresholds within the optimal range using network properties pertaining to the edge weight distribution. Applying our automated threshold selection heuristic prior to clustering improves performance for both Force and MCL. In addition, automated threshold selection bridges the gap between Force and MCL in comparative accuracy analysis. We also tested the threshold heuristic on three other clustering algorithms. The use of a threshold yielded improvement, but MCL continued to outperform all other algorithms after a threshold was applied. As a result, we believe researchers may now more confidently use the time-efficient MCL clustering technique for most of their protein sequence analysis needs.

2 METHODS

2.1 Dataset selection

Protein sequence datasets from four well-studied superfamilies were used in our study. Each superfamily is composed of individual families categorized by a distinct set of functions. This allowed us to test cluster performance based on how well individual clusters overlap with functionally characterized protein families. Two of the superfamilies, Enolase (Gerlt *et al.,* 2005) and Amidohydrolase (Seibert and Raushel, 2005), represent enzymes that perform catalytic functions. These superfamilies are available as a ‘gold standard’ set of well-characterized mechanistically diverse enzyme superfamilies (Brown *et al.,* 2006) in the Structure-Function Linkage Database (Pegg *et al.,* 2006). A third dataset was composed of sequences from a recent study on the solute-carrier transferase (SLC) superfamily (Schlessinger *et al.,* 2010). The final dataset contained sequences from the extensively studied Kinase superfamily (Manning *et al.,* 2002). A total of 1174 amidohydrolase sequences, 1308 enolase sequences, 696 SLC sequences and 527 kinase sequences were used in our study.

Of course, this data represents just a small sampling of each superfamily. For amidohydrolase alone, there are over 20 000 known members. Nonetheless, the families in each dataset represent a diverse sampling of sequence–structure–function relationships which are not trivial to distinguish from one another. Certain superfamily members in the dataset share nearly identical sequences, with a few amino acids accounting for the different functions they perform (Seffernick *et al.,* 2001). More divergent families often share similar structural elements in which at least the active site residues associated with the superfamily-common partial reaction are conserved despite sharing a low level of sequence identity (Glasner *et al.,* 2006). Thus, our sampling of superfamily data provides good test cases for measuring algorithm performance.

2.2 Computing the similarity network

For all four datasets, we carried out an all-by-all BLAST search using a custom database built from all sequences in the dataset. Four such runs were executed for each of the four families. The BLAST expectation value (E-value) cutoff for each search was set to one. Next, each protein was treated as a node in the similarity network. Whenever a BLAST alignment was returned between two proteins in the dataset, we connected these proteins with an edge. Each edge was given a weight equivalent to the $-$log of the BLAST E-value. Of course, using such a relaxed cutoff value produces a dense network where virtually every node is connected to every other node.

2.3 Evaluating Clustering performance across an edge weight threshold range

Each superfamily similarity network was filtered across a consecutive series of edge weight thresholds, ranging from zero to 100. At each threshold, all edges below the threshold were removed from the network. The filtered similarity network was then clustered using both Force and MCL. Clustering performance for each algorithm was quantified through F-measure, an evaluation criterion previously used both to study and compare multiple protein clustering techniques (Paccanaro *et al*., 2006; Wittkop *et al*., 2007) as well as in other areas of research where clustering is used (Chim and Dang, 2007). F-measure, ranging in value from zero to one, allowed us to compare the performance of both algorithms over the entire threshold range.

To compute the F-measure, we characterized all pairs of proteins classified as belonging to the same functional family as true positives and all pairs of proteins classified as belonging to different families as true negatives. Each clustering run estimated the identities of the families, with respect to the family assignments in each dataset, leading to a count of true positives, false positives, true negatives and false negatives in the clustered data. These four values were then used to compute precision (P) and recall (R), which were then used to generate the F-measure using the formula $2{*}P{*}R/(P+R)$. An F-measure of 0.5 or less indicates clustering performance that was no better than random. An F-measure of 0.9 or more indicates very accurate clustering performance, because both high precision and high recall are desirable, and the F-measure reflects both these values as their arithmetic mean (Rodriguez-Esteban *et al.,* 2009).

2.4 Analyzing the edge weight distribution

In order to test how threshold selection relates to the similarity network edge weight distribution, we computed the edge weight histogram for each of the four superfamily networks. The number of edges in each network at each threshold value was counted and plotted. For binning purposes, we rounded the –log of the edge weights to the nearest integer. The edge weight histogram plot could then be overlaid with the thresholded clustering performance data to test for a relationship between performance and the shape of the distribution.

To better overlay distribution shape and clustering performance, we normalized each edge weight distribution. Normalization was carried out by first selecting the edge weight bin containing the greatest edge count. Next, the edge count in each edge weight bin was divided by this maximum value. This resulted in a distribution whose value at each edge weight ranged from zero to one. This range also corresponds to the range of F-measure, allowing us to view clustering performance and edge weight distribution shape using a single axis in our plots.

2.5 Designing and testing an automated edge weight threshold selection heuristic

Network-based clustering algorithms group protein sequences into clusters that ideally correspond to functional families by estimating the edges that most likely connect proteins belonging to the same cluster based on network topology. These techniques arise directly from graph theory, and therefore do not consider certain biological properties relevant to our networks of interest. In particular, a purely topological analysis does not explicitly take into account that proteins with very low sequence identity are less likely to perform the same function as proteins with greater similarity (Ponting, 2001). With this assumption in mind, we aimed to design a simple threshold selection heuristic for automatically prefiltering a protein similarity network prior to clustering.

In order to do so, we first needed to study the properties of the similarity network edge weigh distribution, and how these properties overlap with Force and MCL clustering performance. These observations, discussed in Section 3, led to the direct development of a threshold selection heuristic. The details and logic behind our heuristic are discussed in Section 3.4.

Suspecting our heuristic would arise from specific details pertaining to the Force and MCL clustering algorithms, we expanded the scope of our evaluation beyond these two clustering approaches by choosing three additional biological network clustering algorithms for testing. Using each algorithm, we clustered all four networks, both in their unthresholded state as well as at the threshold determined using our heuristic. We used this comparison to evaluate whether the automatically selected threshold generally leads to better clustering performance.

The three additional algorithms selected for testing our heuristic were TransClust, Spectral Clustering of Protein Sequences (SCPS) (Paccanaro *et al.*, 2006) and Affinity Propagation. The first two of these were designed to cluster protein similarity networks and the third is a general purpose clustering algorithm. TransClust is a geometrical layout-based clustering algorithm similar to Force, designed to cluster proteins into families directly. SCPS is a variation of spectral clustering. Unlike most spectral clustering algorithms, SCPS does not require the number of clusters to be known in advance. SCPS was designed with the purpose of clustering protein sequences into superfamilies, but its capacity to cluster proteins into families has not yet been explored. Affinity Propagation has been suggested as an alternative to MCL for protein interaction networks (Vlasblom *et al.*, 2009). The diversity of purpose behind these approaches gave us additional reason to measure their ability to cluster proteins into families, relative to Force, MCL, and each other, under both thresholded and unthresholded conditions.

3 RESULTS

3.1 Edge weight distribution shape

Figure 1 shows the clustering performance for all four datasets over the threshold range. Clustering performance has been overlaid with the normalized edge weight distribution associated with each dataset.

The edge weight distributions for all four datasets share similar characteristics. The maximum point in each distribution is located at a very low edge weight value, at or near zero. As the edge weight value increases, the normalized edge count begins to decay. It descends from the maximum value of one towards a small value between 0.1 and zero. In three of the four distributions (Fig. 1A, B and D), a second, much smaller peak is also present. The smaller local maximum is located further along each distribution, at a larger edge weight value. Eventually, as the edge weight increases, each edge weight distribution drops to a value of zero and does not rise again.

The four edge weight distributions may be further subdivided into two broad categories based on the rate with which each distribution descends from the maximum toward the local minimum. In the Amidohydrolase and SLC distributions (Fig. 1A and B, respectively), the descent is immediate, occurring over a range of less than five edge weight bins. In the Enolase and Kinase distributions (Fig. 1C and D, respectively), the descent is more gradual, occurring over a range of 20 or more edge weight bins. We refer to the former as *rapid-descent* histograms, and the later as *gradual-descent* histograms.

3.2 MCL performance over threshold range

MCL algorithm performance follows the same general trend for all four datasets. Initially, the unthresholded MCL clustering results produce a low-performance F-measure. For three of the four datasets (Fig. 1A, C and D), the initial MCL F-measure is below 0.5. The MCL inflation parameter is known to influence the granularity of the clustering. We therefore explored the impact of alternate inflation parameters on the initial MCL F-measures for these superfamilies (Supplementary Table S1). These alternate inflation values did not yield significant improvements.

As the edge weight threshold increases and the edge weigh distribution begins to decrease from the maximum, the MCL performance increases in quality. When the edge weight distribution approaches the local minimum, the MCL performance measure finally plateaus at its maximum value. For three of the four datasets (Fig. 1A, B and D), the maximum performance plateau is above 0.9. Finally, as the edge weight distribution decays completely to zero, MCL performance also drops significantly. Since MCL performance is greatly dependent on the shape of the edge weight distribution, it is not surprising that performance improves at a greater rate and plateaus at a lower threshold in the rapid-descent histograms than it does in the gradual-descent histograms.

3.3 Force performance over threshold range

Force algorithm performance diverges across the two categories of edge weight distributions. In the two rapid-descent histograms, Force performs well even when no initial threshold is present. Force performance for both the rapid-descent histograms lies between 0.8 and 0.9 at edge weight zero. Performance then rises to approximately 0.9 as additional thresholds are considered. Eventually, when the threshold becomes too great, Force performance quickly decays in a manner similar to MCL performance. These results indicate that thresholding provides the Force clustering algorithm with little additional benefits when a rapid-descent distribution is present.

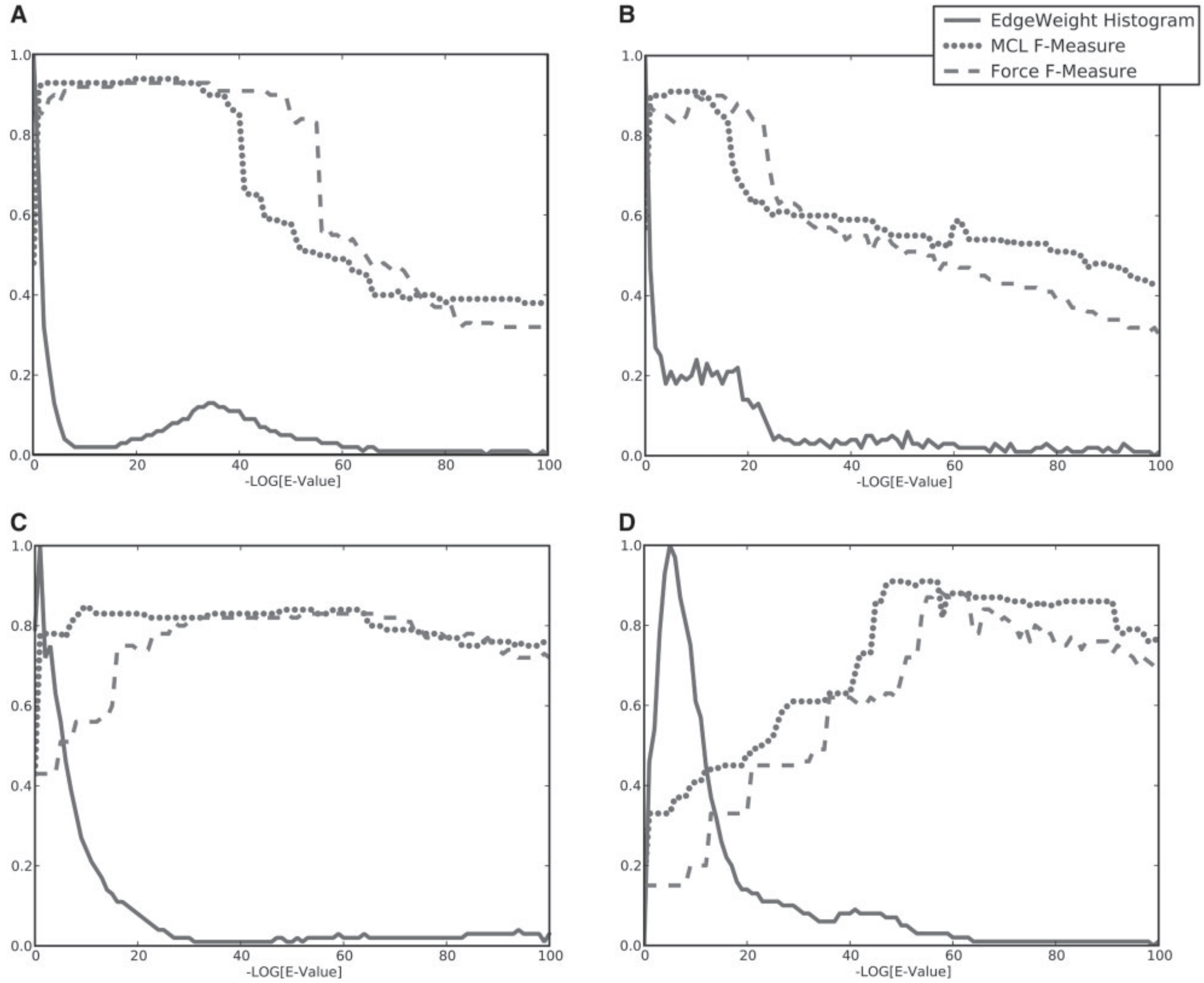

Fig. 1. Clustering performance and edge weight distributions. Each plot shows the F-measure clustering performance metric for both the Force and MCL clustering algorithms over a range of binned $-\log(E\text{-value})$ thresholds, together with a normalized edge weight distribution. (**A**) Amidohydrolase; edge weight distribution is rapid-descent. (**B**) SLC; edge weight distribution is rapid-descent. (**C**) Enolase; edge weight distribution is gradual-descent. (**D**) Kinase; edge weight distribution is gradual-descent.

Furthermore, unthresholded Force clustering will outperform unthresholded MCL clustering in a rapid-descent histogram.

For the two gradual-descent histograms, the opposite holds true. Initially, unthresholded Force performance is poor, falling below 0.5. As the threshold rises from zero, performance increases. This increase, however, is more gradual than the increase in MCL performance over the same threshold range. Eventually, Force performance plateaus at a maximum value equal to the best MCL performance. However, because of the gradual performance increase, Force reaches its maximum value at a greater threshold than MCL. Eventually, the threshold becomes too large, and algorithm performance decays.

3.4 Developing an automated threshold selection heuristic from edge weight distributions

As the edge weight distribution drops rapidly, we observe an increase in clustering quality. From these observations, we may assume that at low-level thresholds, most of the edges removed exist between protein families. The presence of these intercluster edges is effectively noise, which impacts the overall clustering results. Eventually, when the threshold is high enough, a boundary is reached at which most intercluster edges have already been removed. The boundary represents the optimal threshold separating intercluster edges from intracluster edges. At this boundary, the protein family components may begin to be affected by the filtration process, and certain loosely connected nodes may break off from the main network. However, the final increase in clustering precision overcomes any decrease in clustering recall, and the overall clustering quality noticeably improves. Thus, we would like to use this as our threshold in order to maximize the filtration of intercluster edges while minimizing the disruption of the protein family clusters.

Our goal was to estimate this boundary automatically, without knowing in advance the identity of the proteins in the network. We heuristically approximated this boundary b using two available network properties. The first property, Nn(*Th*), is the number of nodes connected by one or more edges at threshold *Th*. The second property is SE(*Th*), the number edges remaining after threshold *Th* is applied. We combined these two properties into a single network summary value, Nsv, where Nsv(*Th*) = SE(*Th*)/Nn(*Th*). Conceptually, Nsv(*Th*) is equivalent to the average weighted node degree at threshold *Th*.

We chose to examine Nsv because its derivative with respect to the threshold, dNsv(*Th*)/d*Th*, could potentially reveal interesting behaviors pertaining to the network. At low value thresholds, most of the filtered edges are between families, while the individual family components remain strongly connected. At these thresholds, the value of SE decreases while the value of Nn(*Th*) remains stable. Thus, as we begin to increase the threshold and filter out the noisy intercluster edges, we expect the value of dNsv(*Th*)/d*Th* to be negative.

When *Th* = *b*, we expect most intercluster edges to have been already removed. We also expect a few poorly connected outlier nodes to disconnect completely from the network, leading to a decrease in Nn(*Th*). SE(*Th*) will also continue to decrease, but at a lesser rate then at lower thresholds, due to a slowdown in the initial rapid decay observed in the edge weight distribution. If the decrease Nn(*Th*) is great enough, and the decrease in SE(*Th*) is low enough, then the value Nsv may actually increase. In this case, dNsv(*Th*)/d*Th* will take on a positive value at a threshold proximate to the boundary, but not at lower thresholds. This leads to the following threshold estimation heuristic: *b* is approximate to the minimum threshold *Th* at which dNsv(*Th*)/d*Th* > 0. If Nsv does not increase at any point in the distribution, then no threshold is returned.

We applied this heuristic to all four networks, generating thresholds of 1.0 for SLC, 1.0 for Amidohydrolase, 20.0 for Enolase and 69.0 for Kinase. Plots of dNsv(*Th*)/d*Th* for all four networks are available in the Supplementary Figure S1. The heuristically determined thresholds lead to a performance increase in three of the four datasets (Fig. 1A, C and D) for both the Force and MCL algorithms, relative to the unthresholded performance of these same algorithms. For the fourth dataset (Fig. 1B), thresholding leads to an increase in MCL performance and a slight decrease of .01 in Force performance. For all four datasets, MCL performance at the heuristically determined threshold is greater than Force performance at that same threshold.

We qualitatively confirmed the improvement in clustering quality by visualizing the generated clusters prior to and after filtering with the heuristically determined thresholds described above (Figs 2 and 3) using Cytoscape (Shannon *et al.,* 2003), a biological network visualization and analysis tool with both MCL and Force clustering capabilities, and through the use of its ClusterMaker plugin (http://www.cgl.ucsf.edu/cytoscape/cluster/clusterMaker.html). Visualization was carried out by removing all edges from each similarity network that did not correspond to pairs of proteins within the same cluster. Afterwards, the clusters within each network were made visible through Cytoscape's Organic layout, which is a force-directed layout algorithm similar to Fruchterman-Reingold (Fruchterman and Rheingold, 1991). Nodes in the visualized network were colored by known protein family assignment to allow for visual assessment of clustering quality.

Figure 2A shows the unthresholded MCL clustering output for the SLC superfamily, where many false negatives are present. Multiple proteins belonging to the same family are grouped across distinct clusters, instead of being grouped together. In Figure 2B, the heuristically selected threshold has been applied and the number of false negatives has correspondingly decreased. Eliminating certain redundant edges between subsets of proteins within families prevents these subsets from clustering into distinct groups. As a result, more proteins within the same family are clustered together.

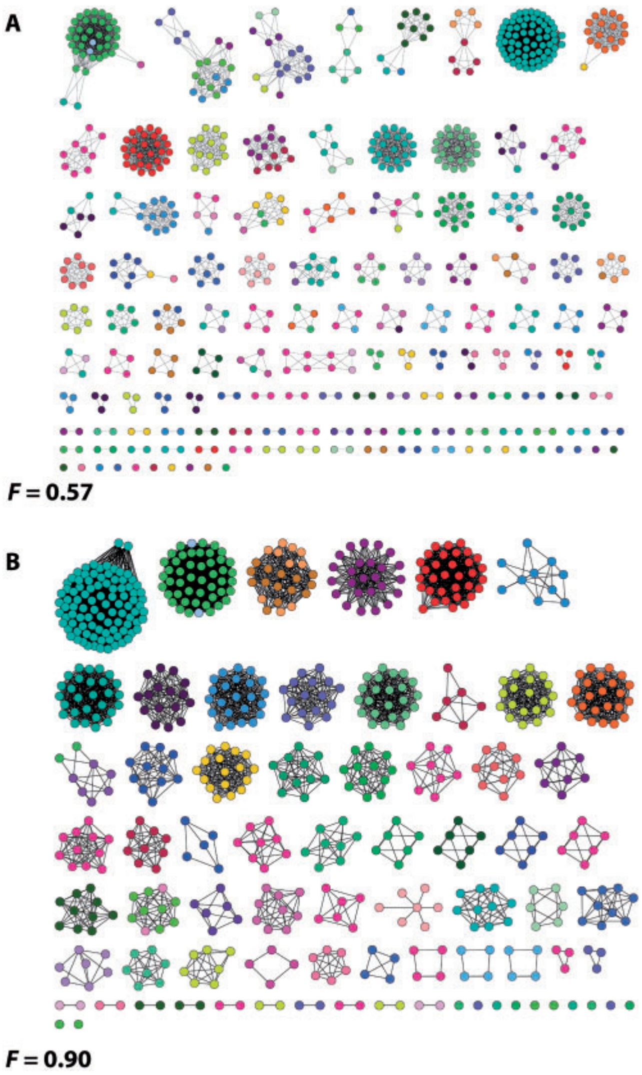

Fig. 2. Visualizing MCL Clusters for the SLC Superfamily. Each set of clustering results has been visualized in Cytoscape using the Force-directed layout algorithm. Each node represents a protein, colored by the currently best available family assignments. Edges between nodes that are not in the same cluster have been removed from the similarity network prior to visualization. The unthresholded clustering results are shown in (**A**) and the thresholded clustering results are shown in (**B**). The same thresholded network is shown unclustered in Supplementary Figure 4b. The mapping of node colors to family assignments is shown in Supplementary Figure 5a.

Figure 3A shows the unthresholded MCL clustering output for the kinase superfamily. The two resulting clusters provide little discrimination among sequences that are known to belong to different functional families. Figure 3B shows the corresponding change in clustering output after applying a heuristically selected threshold. Many of the families that have previously clustered

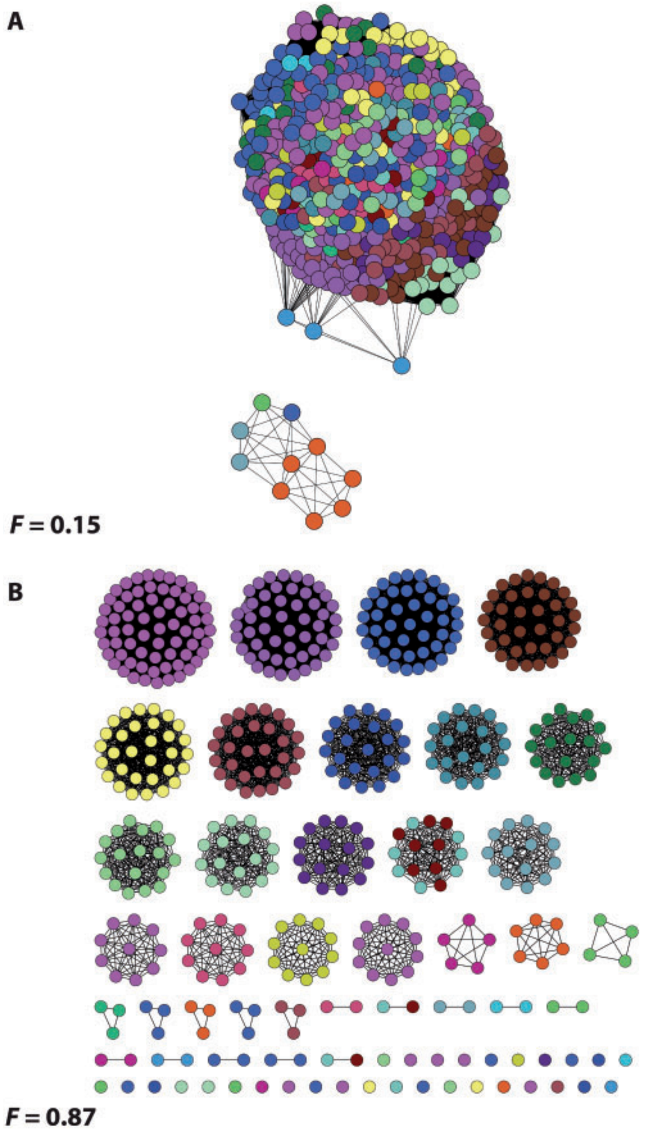

Fig. 3. Visualizing MCL Clusters for the Kinase Superfamily. Each set of clustering results has been visualized in Cytoscape using the Force-directed layout algorithm. Each node represents a protein, colored by the currently best available family assignments. Edges between nodes that are not in the same cluster have been removed from the similarity network prior to visualization. The unthresholded clustering results are shown in (**A**) and the thresholded clustering results are shown in (**B**). The same thresholded network is shown as unclustered in Supplementary Figure 4d. The mapping of node colors to family assignments is shown in Supplementary Figure 5b.

together now separate out into their own distinct clusters. This same pattern as seen in the kinase family also holds for Amidohydrolase (Supplementary Fig. S2) and Enolase (Supplementary Fig. S3) superfamilies.

Visualization of the clustering of well-annotated protein families reaffirms that the increase in F-measure values after automated thresholding corresponds to a genuine increase in clustering quality. Thresholding eliminates false positives in some networks, and eliminates false negatives in others, leading to a relevant improvement in the accuracy of the final clustered results. Visualizing the thresholded networks prior to clustering (Supplementary Fig. S4) emphasizes the role of thresholding in the improvement of clustering quality. When the thresholded SLC, Amidohydrolase and Enolase networks are displayed using a force-directed layout, the boundaries between individual families clearly become visible, even though edges continue to connect the families. In the thresholded Kinase network, clusters of individual families separate out completely, resulting in high-quality input for any protein family-specific clustering algorithm.

To investigate further why families separate out from the thresholded Kinase network, but not from the other three datasets, we hypothesized that this network consisted of tightly connected Kinase families with unusually high edge weights. To confirm this, we calculated the average edge weight within families, and also between families, for each of the four datasets (Supplementary Table S2). We found that while the average intrafamily edge weight for Kinase ranked highly at 99.7, the average intrafamily edge weight for Enolase ranked even higher at 114.9. The key to understanding what made Kinase distinct lay in the average edge weight *between* families. The average interfamily edge family edge weight for Kinase was very low, at 13.8, relative to its high intrafamily edge weight. Furthermore, the Kinase superfamily was the only dataset assigned a threshold greater than its average interfamily edge weight. In the other three datasets, our heuristic assigned a threshold that filters some but not all of the intercluster edges. From these results, we draw the conclusion that our threshold heuristic will filter out some network noise without completely disrupting network connectively, except in those cases when there is a significant difference between interfamily and intrafamily edge weights.

Additionally, our results showed that the average intrafamily edge weights for SLC and Amidohydrolase were 38.7 and 51.6, respectively. This leads us to hypothesize that the difference in the shape of gradual-descending and rapid-descending distributions is related to the connectivity between families. In the two rapid-descending datasets, connectivity exists at lower edge weights, resulting in a more rapid transition between interfamily edges and intrafamily edges. We believe this more rapid transition results in a more rapid edge weight decay, as observed in our plots.

3.5 Automated threshold selection performance on additional clustering algorithms

Table 1 lists the performance of the MCL, Force, TransClust, SPCS and Affinity Propagation algorithms for both the unthresholded networks, as well as the networks filtered using our automated threshold selection heuristic. MCL outperforms the other four algorithms, but only when the automated threshold is applied.

Force ranks second in overall performance. It produces results with F-measure greater than 0.80 for two of the unthresholded networks and three of the thresholded networks. The fourth thresholded network, Enolase, scores an F-measure of 0.74 under Force. This is a great improvement over the unthresholded F-measure of 0.43, but is still less than the 0.83 F-measure associated with the MCL clustering of the thresholded Enolase network.

TransClust ranks third in clustering performance. Both the thresholded and unthresholded SLC networks score an F-measure of 0.87. The thresholded Kinase network scores an F-measure of 0.82,

Table 1. F-measure scores of clustering algorithms for thresholded and unthresholded superfamilies

	Amidohydrolase		SLC		Enolase		Kinase	
	U	T	U	T	U	T	U	T
MCL	0.48	0.92	0.57	0.90	0.43	0.83	0.15	0.87
Force	0.87	0.86	0.86	0.87	0.43	0.74	0.15	0.84
TransClust	0.49	0.59	0.87	0.87	0.46	0.66	0.15	0.82
SCPS	0.30	0.37	0.12	0.65	0.40	0.65	0.15	0.88
SCPS $\epsilon = 1.1$	0.33	0.37	0.70	0.80	0.40	0.72	0.15	0.88
AP	0.16	0.16	0.14	0.15	0.14	0.17	0.16	0.16

The left-most full column of the table lists the clustering algorithms tested. The next four full columns represent the F-measure scores for clustering results across each of the four superfamilies. Each full superfamily column subdivides into two subcolumns; U and T. U represents the F-measure for the clustered, unthresholded superfamily networks. T represents the F-measure for the clustered, thresholded superfamily networks.

improving significantly from the unthresholded F-measure of 0.15. Thresholding the Enolase and Amidohydrolase networks also improves the TransClust output, but not to a significant extent. Both these thresholded networks score an F-measure of less than 0.70.

SCPS performs exceedingly poorly when its primary parameter values remain unchanged. Although clustering improvements are observed in all four networks after thresholding is applied, F-measure values score below 0.70 for three of our four datasets. In an effort to improve these, we attempted to adjust the SPCS epsilon parameter, which is responsible for tighter clustering at higher values. By sampling the epsilon parameter along increments of 0.01, we determined that an epsilon value of 1.1 leads to better clustering than the primary epsilon value of 1.0. At epsilon 1.1, SCPS clustering of the thresholded SLC and Kinase networks results in F-measure scores that are equal to or greater than 0.80. The thresholded F-measure for Enolase is 0.72, which is a significant improvement over the unthresholded F-measure of 0.40. Unfortunately, the Amidohydrolase F-measure remains exceedingly poor, scoring at less than 0.40, even after thresholding.

Affinity Propagation is unable to cluster the four protein networks into functionally meaningful families. All F-measures fall below 0.20. Sampling alternate Affinity Propagation parameters did not yield the improvement.

In order to confidently reconfirm these quantitative observations, we recalculated the clustering table using the Geometric Separation statistic (Brohee and van Helden, 2006) initially developed to compare the quality of protein interaction network clustering approaches. The Geometric Separation results (Supplementary Table S3) confirm the conclusions drawn from the F-measure table. The use of an automatically selected threshold improves the Separation statistic, and the thresholded MCL Separation scores rank the highest relative to the other clustering algorithms in the table.

4 DISCUSSION

The results indicate that the shape of a protein similarity network edge weight distribution correlates with how well the network clusters over a range of thresholds. It is this relationship between the distribution and clustering potential that allows our simple threshold selection heuristic to improve the quality of clustering results in the variety of networks we studied. This is in contrast to the more complicated approach taken by Harlow *et al.* in which they performed single linkage hierarchical clustering on MCL results (Harlow *et al*., 2004). Although these observations are limited to superfamily-based sequence similarity networks of medium size, they nonetheless represent a valuable step in solving the difficult problem of clustering proteins into family groups that may be informative of their different functions. Researchers interested in clustering larger, more diverse datasets may now efficiently group the data into superfamilies using algorithms like SCPS, and afterwards clustering each superfamily into families with the aid of our threshold selection heuristic.

Our results also show that MCL outperforms other common algorithms in the task of clustering proteins into families, after the appropriate threshold is applied. The Force algorithm ranks second. This is in contrast to previous research (Wittkop *et al*., 2007), which showed Force outperforming MCL, as indicated by F-measure. Previous performance comparisons have all been carried out on unthresholded networks. The conclusion that Force outperforms MCL is to be expected when network thresholding is not taken into account. Based on our results, when a threshold is not provided, Force outperforms MCL in a network containing a rapid-descent edge weight distribution and performs just as poorly as MCL in a network with a gradual-descent edge weight distribution. However, as we have demonstrated, an appropriate threshold is easy to extract from an input edge weight distribution. Once that threshold is applied, MCL performs as well as or better than Force. By extending both algorithms to include a preliminary automated threshold selection step, the performance difference between the two approaches can be minimized.

Eliminating the performance gap between Force and MCL is an important development because of the large difference in execution times of the two algorithms. As the size of the network increases, the execution time required for running Force goes up significantly (Wittkop *et al*., 2007). On a modern desktop computer, the Amidohydrolase network takes 5 h to cluster with Force, while MCL clusters the same network in less than 2 min under the same conditions. Given this difference in runtime, and our results that show MCL clustering quality is equal to or better than Force after a heuristically selected threshold is applied, we argue that MCL should be the algorithm of choice. This choice can be especially important when processing large high-throughput protein similarity datasets. For example, the Amidohydrolase superfamily has more than 20 000 members. Using the current implementation of Force would not be feasible for such a superfamily. Applying heuristically selected thresholding to such a massive dataset allows us to cluster the proteins using the faster-performing MCL algorithm without fear of sacrificing accuracy for the sake of speed.

Finally, our general comparison of biological network clustering approaches illustrates the importance of properly distinguishing between categories of networks prior to selecting an appropriate clustering algorithm. Not all biological networks are equal, and not all network-related problems are equal. Although most of the algorithms we tested showed improvement after thresholding, not all algorithms improved equally. This is because some algorithms are more adept for certain types of problems than for others. SCPS, which was designed to group large sequence sets into superfamilies, clustered reasonably well but did not score as high as the more family-specific MCL and Force algorithms. Affinity Propagation, a general purpose algorithm for clustering nodes in networks, had difficulty in processing protein similarity networks of the scale used here. Thus, it is vital for researchers to proceed with caution before selecting a clustering algorithm appropriate for the problem at hand.

Ultimately, application of sequence similarity networks for functional inference requires clustering results that correspond, to the extent possible, with functionally relevant relationships. A critical step in achieving this goal is automated clustering of sequence similarities without benefit of knowledge about their functional properties. As illustrated here, our approach provides a useful heuristic to improve network clustering in this regard. More research will be required, however, to better understand the relationship of functional divergence to clustering of sequences using similarity networks.

5 CONCLUSIONS

We have examined the role that edge weight distribution plays in network clustering and shown how it may be used to improve the performance of several popular network clustering algorithms. Our automated threshold selection heuristic provides a simple approach for determining an appropriate threshold for network clustering. This threshold may then be employed to eliminate the current gap in clustering quality between MCL and other algorithms, thus alleviating the need to incur the heavily penalty in execution time needed with alternate algorithms such as Force. In addition our results, as shown in Table 1, suggest that thresholding generally improves clustering quality for four out of the five tested clustering algorithms. In the context of using protein similarity networks for functional inference, the significant improvement in clustering quality for these algorithms suggests that any future algorithms designed for this application include a threshold heuristic.

More importantly, our research demonstrates that the predictive potential of the similarity network edge weight distribution is an area of study worth exploring in more detail. Future examination of edge weight distributions may help produce better threshold selection approaches, as well as possibly leading to the development of more accurate network clustering algorithms. Furthermore, additional study of edge weight distribution shape could also provide a deeper understanding of protein similarity networks as a whole.

Funding: National Institutes of Health (grant P41 RR01081 to T.E.F. and R01 GM60595 to P.C.B.).

Conflict of Interest: none declared.

REFERENCES

Altschul,S.F. *et al.* (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.,* **25**, 3389–3402.

Apweiler,R. *et al.* (2004) UniProt: the Universal Protein knowledgebase. *Nucleic Acids Res.,* **32**, D115–D119.

Atkinson,H.J. *et al.* (2009) Using sequence similarity networks for visualization of relationships across diverse protein superfamilies. *PLoS ONE,* **4**, e43–e45.

Brohée,S. and van Helden,J. (2006) Evaluation of clustering algorithms for protein-protein interaction networks. *BMC Bioinformatics,* **7**, 488–506.

Brown,S.D. *et al.* (2006) A gold standard set of mechanistically diverse enzyme superfamilies. *Genome Biol.*, **7**, R8.

Chim,H. and Deng,X. (2007) A new suffix tree similarity measure for document clustering. In *Proceedings of the 16th International Conference on World Wide Web*. Association for Computing Machinery, Alberta, Canada, pp. 121–130.

Enright,A.J. and Ouzounis,C.A. (2001) BioLayout—an automatic graph layout algorithm for similarity visualization. *Bioinformatics,* **17**, 853–854.

Enright,A.J. *et al.* (2002) An efficient algorithm for large-scale detection of protein families. *Nucleic Acids Res.*, **30**, 1575–1584.

Frey,B.J. and Dueck,D. (2007) Clustering by passing messages between data points. *Science,* **315**, 972–976.

Frivolt,G. and Pok,O. (2006) Comparison of graph clustering approaches. In *Proceedings in IIT.SRC*. Slovak University of Technology, Veliko Turnovo, Bulgaria, pp. 168–175.

Fruchterman,T.M. and Rheingold,E.M. (1991) Graph drawing by force directed placement. *Softw. Exp. Pract.,* **21**, 1129–1164.

Gerlt,J.A. *et al.* (2005) Divergent evolution in the enolase superfamily: the interplay of mechanism and specificity. *Arch. Biochem. Biophys.,* **433**, 59–70.

Glasner,M.E. *et al.* (2006) Evolution of structure and function in the o-succinylbenzoate sythase/N-acylamino acid racemase family of the enolase superfamily. *J. Mol. Biol.,* **360**, 228–250.

Harlow,T.J. *et al.* (2004) A hybrid clustering approach to recognition of protein families in 114 microbial genomes. *BMC Bioinformatics,* **5**, 45–58.

Lu,F. *et al.* (2005) Framework for kernel regularization with application to protein clustering. *Proc. Natl Acad. Sci. USA,* **10**, 12332–12337.

Manning,G. *et al.* (2002) Evolution of protein kinase signaling from yeast to man. *Trends Biochem. Sci.,* **27**, 514–520.

Noble,W.S. *et al.* (2005) Identifying remote protein homologs by network propagation. *FEBS J.*, **272**, 5119–5128.

Paccanaro,A. *et al.* (2006) Spectral clustering of protein sequences. *Nucleic Acids Res.,* **34**, 1571–1580.

Pegg,S.C.H. *et al.* (2006) Leveraging enzyme structure-function relationships for functional inference and experimental design: the Structure-Function Linkage Database. *Biochemistry,* **45**, 2545–2555.

Ponting,C.P. (2001) Issues in predicting protein function from sequence. *Brief. Bioinformatics*, **2**, 19–29.

Rahmann,S. *et al.* (2007) Exact and heuristic algorithms for weighted cluster editing. *Comput. Syst. Bioinformatics Conf.*, **6**, 391–401.

Rodriguez-Esteban,R. (2009) Biomedical text mining and its applications. *PLoS Comput. Biol.*, **5**, e1000597.

Schaeffer,S.E. (2007) Graph clustering. Comp. *Sci. Review*, **1**, 27–64.

Schlessinger,A. *et al.* (2010) Comparison of human solute carriers. *Protein Sci.*, **19**, 412–428.

Seffernick,J.L. *et al.* (2001) Melamine deaminase and atrazine chlorohydrolase: 98 percent identical but functionally different. *J. Bacteriol.,* **183**, 2405–2410.

Seibert,C.M. and Raushel,F.M. (2005) Structural and catalytic diversity within the amidohydrolase superfamily. *Biochemistry*, **44**, 6383–6391.

Shannon,P. *et al.* (2003) Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome. Res.*, **13**, 2498–2504.

Vlasblom,J. and Wodak,S.J. (2009) Markov clustering versus affinity propagation for the partitioning of protein interaction graphs. *BMC Bioinformatics,* **10**, 99–112.

Wittkop,T. *et al.* (2007) Large scale clustering of protein sequences with FORCE - a layout based heuristic for weighted cluster editing. *BMC Bioinformatics*, **8**, 396–407.

Wittkop,T. *et al.* (2010) Partitioning biological data with transitivity clustering. *Nat. Methods*, **7**, 419–420.