2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum

The Development of Parallel Adaptive Sampling Algorithms for Analyzing Biological Networks

Kathryn Dempsey*, Kanimathi Duraisamy[0], Sanjukta Bhowmick[0], Hesham Ali[0]*
[0]College of Information Science and Technology, University of Nebraska at Omaha
*Department of Pathology & Microbiology, University of Nebraska Medical Center
Contact Email: hali@mail.unomaha.edu

***Abstract*- The availability of biological data in massive scales continues to represent unlimited opportunities as well as great challenges in bioinformatics research. Developing innovative data mining techniques and efficient parallel computational methods to implement them will be crucial in extracting useful knowledge from this raw unprocessed data, such as in discovering significant cellular subsystems from gene correlation networks. In this paper, we present a scalable combinatorial sampling technique, based on identifying maximum chordal subgraphs, that reduces noise from biological correlation networks, thereby making it possible to find biologically relevant clusters from the filtered network. We show how selecting the appropriate filter is crucial in maintaining the key structures from the original networks and uncovering new ones after removing noisy relationships. We also conduct one of the first comparisons in two important sensitivity criteria—the perturbation due to the vertex numbers of the network and perturbations due to data distribution. We demonstrate that our chordal-graph based filter is effective across many different vertex permutations, as is our parallel implementation of the sampling algorithm.**

Keywords: chordal graphs, ordering, correlation networks, edge enrichment, cluster overlap

I. INTRODUCTION

Most of the crucial questions in biology relate to understanding the complex interactions between entities, such as genes or proteins. Large-scale networks, where the nodes represent entities and the edges the interactions between them, are used to represent these biological processes. There exist two important challenges in analyzing networks, particularly those arising in data intensive and experiment fields such as biology. First, networks built from high-throughput assays are extremely large, and therefore the analysis requires filtering the network to reduce its size and/or high performance computing resources for lowering the analytic execution time. Second, networks are inevitably associated with some noise due to experimental calibrations or subjective choice of thresholds. This noise should be reduced for correct analysis and identification of causative network structures.

Network sampling is an obvious choice to reduce both the data size as well as the accompanying noise. However, most network sampling methods, such as random walks, focus on maintaining as many of the key properties of the graph as possible. We contend that for a particular objective based analysis this is potentially harmful on noisy networks, since it also effectively captures noise. Instead of such agnostic sampling, we propose an adaptive method that is designed to conform to the objective of the analysis.

In this paper, we focus on identifying genes and gene clusters with biological functionalities based on gene correlation networks created from microarray data. In the network model, each node represents a gene and two nodes are connected if the associated genes exhibit high correlation in their behavior to stimuli. Regions of highly connected subgraphs (such as cliques, or near-cliques) indicate groups of genes with potential common functions. Our sampling algorithm is developed with the goal of retaining all or most of such cliques. In our earlier works [6,7] we developed a parallel sampling algorithm based on finding the maximal chordal subgraph of the correlation network. Chordal graphs are graphs where any cycle larger than four is cut by a chord so that the largest uncut cycle is a triangle. Algorithms for extracting chordal subgraphs therefore, will attempt to eliminate all larger cycles, while maintaining highly connected regions of the original graph, such as the cliques. Chordal graphs are also triangulated; meaning the largest cycle in the graph is a C_3. The C_3 is a motif commonly identified in biological networks as relating to gene co-expression; i.e., if $gene_A$ has a similar expression pattern to $gene_B$ and $gene_A$ also has a similar expression gene pattern as $gene_C$, then $gene_B$ and $gene_C$ will likely also have a correlated expression pattern, thus forming a triangle in a network because they are connected by common relationship. These features of chordal graphs indicate that extracting the maximal chordal subgraph of a network would match the analysis goals of finding highly connected clusters of genes.

While our initial results demonstrated the effectiveness of the chordal sampling technique, several key aspects of sampling correlation networks remain unstudied. In this paper, we address the following key points that are essential in evaluating the effectiveness of network sampling:

1. *Selection of sampling algorithm***.** We demonstrate the importance of choosing the proper sampling filter by

978-0-7695-4676-6/12 $26.00 © 2012 IEEE
DOI 10.1109/IPDPSW.2012.90

comparing the chordal subgraph filter with a control filter represented by a random walk based sampling, a popular method for filtering. We show that though the size of the networks are close, chordal sampling provides much better approach to retaining and uncovering key genes and gene clusters with biological significance.

2. *Effect of data perturbations.* Like all combinatorial optimization methods, the output of maximal chordal graph is affected by the ordering of the vertices. We study how such perturbations affect our analysis. Our observation is that with different orderings even though the chordal graph changes slightly, the functionality results remain more or less the same.

3. *Effect of parallel algorithms***.** Parallel computing is primarily used to allow the analysis of large datasets and to reduce the analysis time. We present an improved communication-free version of our algorithm for parallel graph sampling. We have also conduct one of the first studies on the impact of parallelism on the results of the analysis—that is, how increasing the number of processors affects the quality of the obtained key genes and gene clusters.

4. *Orthogonal validation of results***.** Our cluster detecting methods are primarily combinatorial, and dependent on the connectivity of the network. We provide an additional verification of these analytical results by extensively comparing the resultant clusters with using orthogonal data from literature.

The remainder of this paper is arranged as follows. In Section II we briefly discuss how correlation networks are created and some recent work on graph sampling. In Section III we describe our main filtering algorithm a highly scalable chordal-graph based sampling and introduce our hypothesis as to why this filter conforms to the analysis objective. In Section IV we provide experimental results on networks obtained from hypothamali of murine models that demonstrate that the empirical data indeed supports our hypothesis. We conclude in Section V with a summary of current results and discussion of future research.

II. BACKGROUND

The use of networks as a representation of high throughput biological data is becoming a popular method for identifying mechanisms behind aging and disease. While the network model is powerful in portraying real biological communication and function, the networks created from biological data are often large and noisy, making handling and analysis of large networks extremely difficult for current algorithms. To circumvent this issue, network filtering or sampling is used to reduce network size and density while *retaining the real biological relationships that define the function of the network*. In this section we give a short overview of how correlation networks are formed from microarray data and a brief description of some of the graph sampling methods currently available in biological networks.

Correlation Networks. Despite the need to explore the mechanisms of aging and disease from experimental data, there exist few models that can handle the size and complexity of this massive volume of information, especially when it is obtained from multiple sources. Correlation networks are effective models for such data analysis because structures within the network can be directly linked to cellular function and thus users can query the network as necessary depending on individual research interests.

In this study, we build a correlation network by examining levels of co-variance using Pearson's correlation coefficient, in microarray data between pairs of genes in the network, that is between every pair of genes in the original dataset. Correlation scores for a gene pair range from $-1.00 \leq \rho \leq 1.00$, from being inversely proportional to being exactly proportional to all values in-between. Correlation values indicate that the two genes in question have some level of common influence and can be biologically related via function if the relationship is not coincidental. After network construction and low correlations removed via thresholding, the model itself can be analyzed for structural and biological impact. It has been shown in correlation networks and many other types of biological networks that structures can be tied directly to biological subsystems [13]. For instance, nodes with a high degree tend to represent essential genes in protein-protein interaction networks [19], clusters of genes tend to represent complexes or regulatory cohorts[19], and other network structures indicate overall communication network position (signaling proteins). Previous studies have identified high centrality nodes (degree, betweenness, closeness and their combinations) to relate to node essentiality in terms of network robustness and organism survival [20]; further, clusters have been shown to have common functions according to Gene Ontology enrichment [9]. The details of this process as it pertains to our research are given in the experimental design description in Section IV. However, correlation networks are notorious for containing noisy edges (correlation does not imply causation) and thus, these structures are harder to find in larger networks as compared to those created from smaller datasets.

Graph Sampling. Graph sampling is effective in reducing coincidental relationships and computational costs while preserving the accuracy of analysis results. Previous work has focused on sampling the networks for better visualization, such as in maintaining degree distribution and component size distribution as the two most important visual features of the network [1] or compression schemes for visualization that preserve the semantics of the original graph [2]. There also exists research in generating sampling algorithms that enhance the structural diversity of the samples [3].

Many sampling methods for large scale-free networks are based on random sampling, such as random node selection or random walks on the network. Leskovic et al. [4] stated that random walks and 'forest fire' approaches are good at extracting samples from

large networks and are effective as a general sampling method that would retain many of the graph properties. A more recent work [5] analyzes the result of various sampling algorithms using three different measures: degree, clustering and reach and demonstrates that no single sampling algorithm is effective in preserving all these properties.

As can be seen from these examples, most of the research in this area is concerned with constructing samples that match structural properties of the original network and do not take into account functionality of the underlying data. Our goal is to match combinatorial properties with the underlying functional objective and thereby, selects good representative samples that can filter out the noise, while preserving important and relevant characteristics of the network.

III. Chordal Graph Sampling

In this section we present our parallel chordal graph based sampling algorithm and provide hypothesis as to why it is appropriate for finding important gene functionality clusters. Most graph filtering algorithms focus on obtaining a good approximation of the underlying graph to reduce the data size for faster computation. Our goal, in contrast, is to selectively remove noise from the network. The reduction of size that is obtained can be used to estimate the amount of noise in the network. Ideally, if the data is noise free, no reduction should occur.

A. Parallel Algorithms for Graph Sampling

Parallel Chordal Graph Based Sampling. Our sampling algorithm is based on finding the maximal chordal subgraph of the network. A *Maximum* chordal subgraph is the largest (based on the number of edges) chordal subgraph that can be obtained and a *maximal* chordal subgraph where addition of any new edge destroys the chordality. A maximal subgraph is not necessarily a maximum subgraph. Finding the maximum chordal subgraph from a given graph is a NP-hard [8] problem. However, Dearing *et al.* [8] has developed a polynomial time algorithm of complexity ($O(Ed)$), where E is the number of edges and d is the highest degree in the graph. This algorithm follows a variation of graph traversal. Initially a starting vertex and its associated edges are selected, and then successive vertices are added to the subgraph as long as chordality of the subgraph is preserved. The algorithm completes when all vertices have been included in the subgraph. We base our parallel implementation on a multiprocessor distributed memory system based on this sequential algorithm.

In our earlier work [6,7] we developed a parallel algorithm for obtaining maximal chordal subgraphs as follows: we divided the network into P partitions. Within each partition, we obtained the maximal chordal subgraph formed only of edges whose endpoints lie completely within the partition. We then identified the border edges whose endpoints lie across the partitions. Next, we exchanged border edges across processors. For every pair of processors one was designated as the sender (of the mutual border edges) and the other as the receiver (of those mutual border edges). The receiver then computed which border edges could be retained while maintaining the chordality of its subgraph. The inclusion of the border edge could potentially (but rarely) add non-chordal edges in the sender's subgraph, resulting in a cycle. We termed this structure, with a few large cycles across the partitions as quasi-chordal subgraphs (QCS).

The total communication per processor depended on the number of border edges (b) that were exchanged, and the scalability was $O(b^2/d)$. A limitation of this implementation is that the algorithm does not scale well. If the network is too small and number of processors is large, then b increases. If the network is too big and there are fewer processors, b also increases significantly. Additionally, depending on the distribution some processors might have more border edges to analyze as compared to other processors. In the current version of our algorithm our primary goal therefore was to reduce the communication costs and maintain a better balance of the workload.

In our current algorithm the graph is partitioned as before and chordal edges (edges of the chordal subgraph within each partition) and border edges (across the

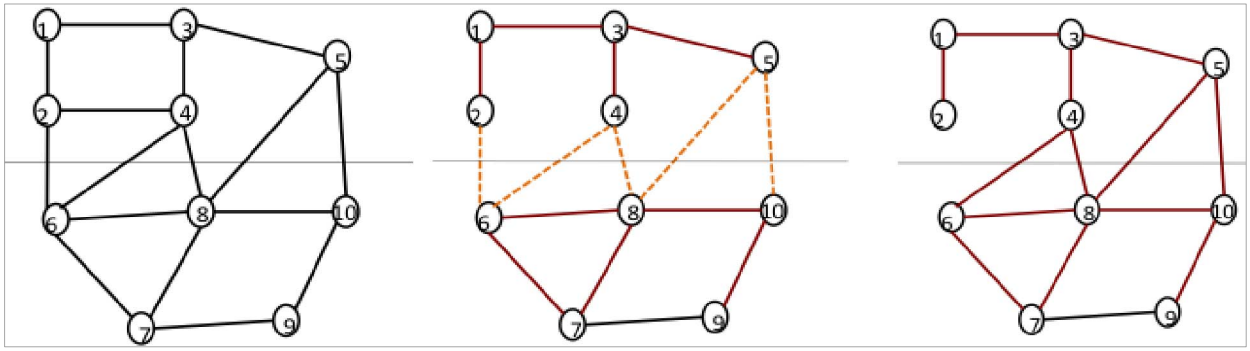


Figure 1: The steps in our chordal graph-based algorithm. Left Figure: *Step 1:* The original graph is partitioned. Middle Figure: *Step 2*: Maximum chordal graph identified for each partition. Edges connecting partitions are classified as border edges. Border edges are not included in the maximal chordal graph identification of Step 2. Right Figure: *Step 3*: Border edges are added to connect partitions.

partitions) are marked (as shown in Figure 1). However, instead of sending the border edges to the receiver, we simply compare them with the local chordal edges. A pair of border edges are included in the subgraph they form a triangle with already marked chordal edge. In Figure 1, edges (2, 6) and (4, 6) will not be included in the top partition because (2, 4) is not a chordal edge. However in the bottom partition (4, 6) and (4, 8) are included since (6, 8) is a chordal edges and so are (5, 8) and (5, 10). As before, some larger cycles such as (3, 4, 5, and 8) are formed.

This implementation requires no communication and provides a more equitable distribution of the workload. It is therefore more scalable than our earlier algorithm. We have also noticed that this method leads to fewer larger cycles, as we insist on the presence of a triangle (rather than just the chordality of the subgraph) to add border edges. Note that, the only border edges can create cycles. Therefore to eliminate cycles, we can copy the subgraph induced by the border edges to a single processor and delete appropriate edges to break the cycle. This however can create cycles within the processors, and we have to check the neighbors of the border edges to detect cycles. Complete elimination of large cycles is challenging because deletion of edges can create newer cycles. However, our experiments have shown that inclusion of some cycles due to parallelization does not deteriorate the results and actually some additional new clusters can be identified in the quasi-chordal graphs as compared to the perfect chordal graphs obtained from the sequential implementation. As with all combinatorial optimization schemes, the size of the resulting chordal graph as well as the edges present therein will depend on the vertex ordering as well as the number of processors used. Our experimental data (see IV. Empirical Results) demonstrates that despite this variability, the biological function of clusters identified via this method is not affected in a negative way.

Note that the elimination of communication comes at a cost. Because multiple processors can work on the same border edge, it is likely that some of the border edges will be represented twice in the final filtered subgraph. During analysis, which is done sequentially, we have to remove these duplications. In the worst case there can be as many as b duplications, where b is the number of border edges.

Parallel Random Walk Based Sampling. In order to compare the effectiveness of our method, we also implemented a parallel random filtering method. The random walk was also designed as a variation on graph traversal. At each vertex of degree d, one of its associated edges was selected with probability $1/d$. The graph traversal was completely random in that we did not maintain a list of which edges or vertices have been visited, and a vertex could be visited multiple times. The rationale for random walk is that tightly connected groups of vertices will have a higher change of being repeatedly selected and therefore cliques and other highly connected regions would be preserved in the filtered graph. The traversal process is continued iteratively until the number of times edges are selection is half the total number of edges in the network.

The parallel random walk algorithm also divides the network across processors and as in the case of the chordal graph based sampling, each processor finds its local random walk based subgraph. However, the addition of the border edges is much simpler. Each border edge is associated with a binary random value, and based on the value the edge is either included in the subgraph (e.g. for value 1) or not (e.g. for value 0). This algorithm is of course perfectly scalable as again no communication is required for the border edges. The random walk filter would also require less execution time than the chordal graph filter, because the choice of the next edge is much simpler—a random choice between d objects as oppose to computing whether chordality is maintained.

Effect of Vertex Ordering. The size of the maximal chordal graph is sensitive to the order in which vertices are accessed. To check whether this affected our analysis of gene functionality, we permuted the original network according to four different vertex orderings as follows: *1. Natural Order*: This is the original order in which the vertices were arranged in the network. This order is generally based on the nomenclature of the genes, such as arranging the genes in alphabetical order. *2. High Degree Order:* The vertices are arranged in descending order of degree. The ones with the higher degree are likely to be processed first. *3. Low Degree Order:* The vertices are arranged in ascending order of degree. The ones with the lowest degree are likely to be processed first. *4. Reverse Cuthill McKee (RCM Order:* The vertices were ordered to reduce the bandwidth of the corresponding adjacency matrix of the graph. In the context of connectivity, this means that closely connected vertices are numbered consecutively.

It is difficult to understand how ordering affects random walk, as the random choices nullifies the effect of vertex ordering. However we have seen that the sizes of the random walk based subgraphs do not change significantly due to different orderings.

B. Hypotheses about Chordal Graph Based Sampling

We now tie in the combinatorial properties of our sampling method with the functional characteristics of our data. Recall that our objective is to identify genes of similar functionality from correlation networks and we require a filter that would: 1) identify key nodes and structures in the correlation networks and 2) also uncover new useful structures (node clusters) that could not be obtained directly from the network due to the presence of noise. We state our hypothesis as follows;

Hypothesis H_0: Given a graph G representing a correlation network obtained from gene expression data, a maximal chordal subgraph G1 of G preserves most of the dense subgraphs of G while excluding edges representing noise-related relationships in the network. The effectiveness of G1 is based on the following

corollary hypothesis that we will empirically prove in Section IV;

H_{0a} – Sampling filters based on finding maximal chordal subgraphs are more effective than standard control filters, such as random walk based filters, in preserving key dense subgraphs and uncovering new ones from the original networks;

H_{0b} – Input parameters such as the order of nodes processed by the filter building algorithms have minimal overall impact on the process of obtaining biologically relevant clusters from networks filtered using maximal chordal subgraphs

H_{0c} – Implementation parameters, such as data distribution and varying number of processors, associated with parallel sampling of the network have minimal impact on the produced clusters. Specifically, by increasing the number of processors, the resulting filtered network has fewer edges but the clusters within remain unaffected.

IV. Empirical Results

Our empirical results fall into two categories. The first deals with the parallel sampling algorithm, their scalability and effect on analysis of results. The second involves a detailed analysis of the clusters obtained, including comparison with the random walk method and chordal graphs with different permutations of the network.

A. Test Suites and Experimental Design

Datasets GSE5140 and GSE5078 were downloaded from NCBI's GEO database and divided based on age/treatment [17, 21]. GSE5078 was divided into young mice (YNG) and middle-aged (MID) mice data; GSE5140 was divided into untreated middle-aged mice (UNT) and creatine-supplemented middle-aged mice (CRE) data sets. Both datasets were designed to identify age-related changes in brain tissue from mouse models at different ages/states.

The general format of the experimental design is as follows: create correlation networks and filter to extract only important relationships, and identify potential subsystems with network clustering. Resulting clusters are then scored and annotated and ranked according to true biological function. This process is performed on all original and sampled networks. All clusters from original networks are compared to all clusters from sampled networks based on the following metrics: *(i)* node overlap, *(ii)* edge overlap, *(iii)* biological relevance of clusters in the original versus the sampled networks, *(iv)* number of known (found in the original network) and new (not found in the original network) clusters identified. These experiments have been designed and datasets chosen with two datasets at two states to test the hypotheses outlined in Section III.

Network creation & cluster identification. Correlation networks were built for all four datasets using Pearson correlation coefficients ($p \leq 0.0005$) of all gene pairs in each dataset; only high correlations ($0.95 \leq \rho \leq 1.00$) were used in the final network analysis. Networks were clustered using AllegroMCODE version 1.0 [22], which identifies clusters as groups of genes that are more highly inter-connected than they are to the rest of the network. The algorithm was run under default parameters on each network and all clusters with a score of 3.0 or higher were included in the final analysis. (Scores of 2.9 or lower tend to indicate small cliques, or K_3 graphs, which were not of interest in this study).

Cluster annotation and scoring. Clusters were annotated using the edge enrichment technique described by Dempsey *et al*. [7] in 2011 which exploits the parent-child nature of any of the three main Gene Ontology annotation trees (biological process, molecular function, cellular component). Each GO tree is a directed acyclic graph where nodes represent functional descriptive terms and directed edges represent term relationships; a parent-child relationship in the tree indicates that the child term is a more specific function than the parent, thus, the deeper in the tree, the more specialized the terms.

The process of cluster annotation via edge enrichment is as follows: For each edge e connecting nodes n_1 and n_2 in some cluster C, the terms associated with genes represented by nodes n_1 and n_2 are identified and mapped onto the GO biological process tree. Then the deepest common parent/ancestor (DCP) of nodes n_1 and n_2 is identified and used to annotate edge e. Scoring is performed using a measure of DCP depth (distance from the ROOT node to the DCP) and term breadth (length of the shortest path from term 1 and term 2) where the final score of edge e is equal to *DCP depth – term breadth*. Edges that represent true relationships will be deep in the tree and closer to each other, so the higher the edge score, the better. In addition, scores at or below 0 are more likely to represent noise or coincidental relationships. Using this method we annotate and score every edge in the current cluster C. Clusters are scored by taking the *average edge enrichment score* (AEES) over all edges in the cluster and function is annotated using the most common/dominating term(s) within the cluster. The depth of that annotation can also indicate a cluster's relevance – a cluster annotated with "metabolic process" means that some majority of edges within the cluster all have that term as a common ancestor within

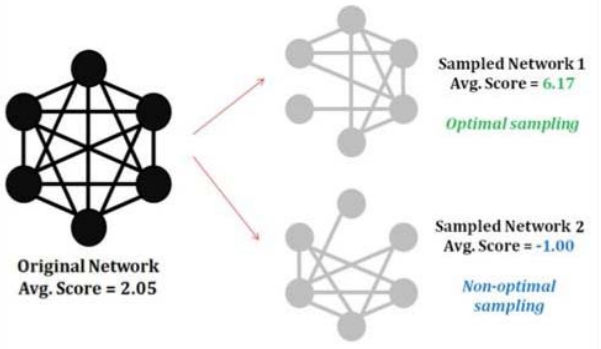


Figure 2: Example of how network sampling can positively or negatively affect the average edge enrichment score of a cluster by removing different sets of edges.

the tree; however the "metabolic process" term is only one step deep in the tree. Using this method of cluster annotation we can compare real function of clusters and sort true biological subsystems from noise as shown in Figure 2.

Cluster overlap. There are four types of clusters that can be identified from comparing original clusters to sampled clusters using the average edge enrichment score and cluster overlap (how many nodes/edges are shared between original and sampled clusters). We use these measures to define sensitivity and specificity of our filters as follows:

- *High AEES, High overlap (True positive):* Clusters that have a high AEES and have a high (>50%) node or edge overlap indicates clusters that were found in the original network and the sampled network, and the cluster has biological meaning.
- *Low AEES, High overlap (False positive):* Clusters that have a low AEES and have a high (>50%) node or edge overlap indicates clusters that were found in the original network and the sampled network, but the cluster likely has no biological meaning. These tend to represent clusters that both original and sampled networks find due to high density or large size but that do not have true biological function.
- *High AEES, Low overlap (False negative):* Clusters that have a high AEES and have a low (<50%) node or edge overlap indicates clusters that were *not* found in the original network but were present in the sampled network, and have biological meaning. These clusters tend to be small and less dense and are only uncovered when noise is removed; hence they are hidden in the original network.
- *Low AEES, Low overlap (True Negative):* Clusters that have a low AEES and have a low (<50%) node or edge overlap indicates clusters that were *not* found in the original network but were present in the sampled network, and likely have no biological meaning.

Using these measures, we can define Sensitivity and Specificity for each filter to identify which (if any) orderings are optimal compared to the others, as shown in Figure 3.

*Lost and Found clusters***.** It is also possible to have clusters in both the original and sampled networks that do not share overlaps; clusters that are only found in the original network are termed as *lost* and clusters that are only found in filtered networks are termed as *found* – found clusters tend to represent smaller and less dense subsystems that are hidden by noise in the larger network. Lost clusters tend to represent subnetworks with cycles that are small and sparse enough that removal of 1-2 edges causes the cluster to fall below the threshold for identification

B. Analysis of Clusters Obtained by Filters

We now analyze the quality of the clusters in each network as obtained by the filters. Our experiments

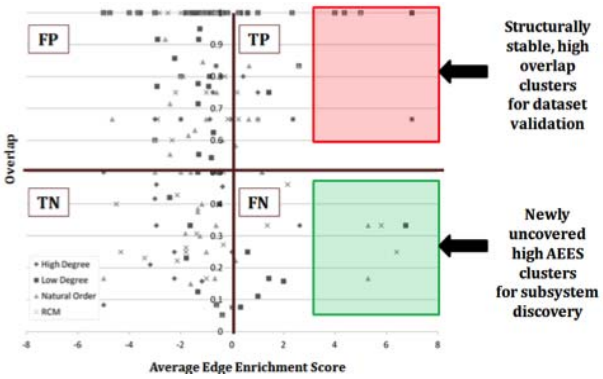


Figure 3: Example of how to identify the likely biologically meaningful clusters. By dividing the graph into equal quadrants, we can identify TP, FP, FN and TN counts. Red box highlights clusters with high AEES scores that were found in both original and clustered networks; the green box highlights clusters with high AEES scores that were found in the original network but were ranked higher in filtered networks.

	Original		High Degree		Low Degree		Natural Order		RCM	
	C#	AvgScore	C#	AvgScore	C#	AvgScore	C#	AvgScore	C#	AvgScore
Young	1	2.824	1	n/a	1	2.581	1	2.759	1	2.333
	2	2.620			2	2.820	2	2.417	2	3.278
	3	2.671			3	3.000	3	3.800	3	2.500
	4	1.700			4	2.582	4	2.250		
	5	2.441			5	1.781	5	2.000		
	6	2.000			6	2.000	6	1.000		
Mid	1	3.724			1	3.681	1	3.375	1	3.667
	2	3.383			2	3.294	2	2.808	2	3.625
	3	2.000			3	3.575	3	1.879	3	2.636
	4	2.067			4	7.000	4	3.347	4	3.400
	5	1.500			5	2.875	5	3.294	5	3.333
	6	3.294			6	2.750			6	1.500
	7	7.000								

Figure 4: Average edge enrichment scores for each cluster in the five orderings for YNG and MID. Higher AEES scores are highlighted with a darker red. C# refers to cluster ID number.

showed that *random walk filtered networks find no clusters at all.* This confirms H_{0a} above; the random walk filter does not identify subsystems/graphs within the network at all, in that there are not enough edges retained using the random walk method to identify very dense groups of nodes. Thus, no clusters are identified via the random walk method.

Preparation of the YNG and MID dataset included using statistical methods to focus on about 33% of the total possible genes, which included only those genes that were differentially expressed between the YNG and MID conditions and thus were thought to be involved in the aging process. This results in a smaller dataset about 25% size of the overall network (compared to UNT and CRE which examine the entire transcriptome). This preprocessing hurts the ability to identify biologically significant clusters, in that only few clusters found had actual biological relevance according to AEES score as shown in Figure 4.

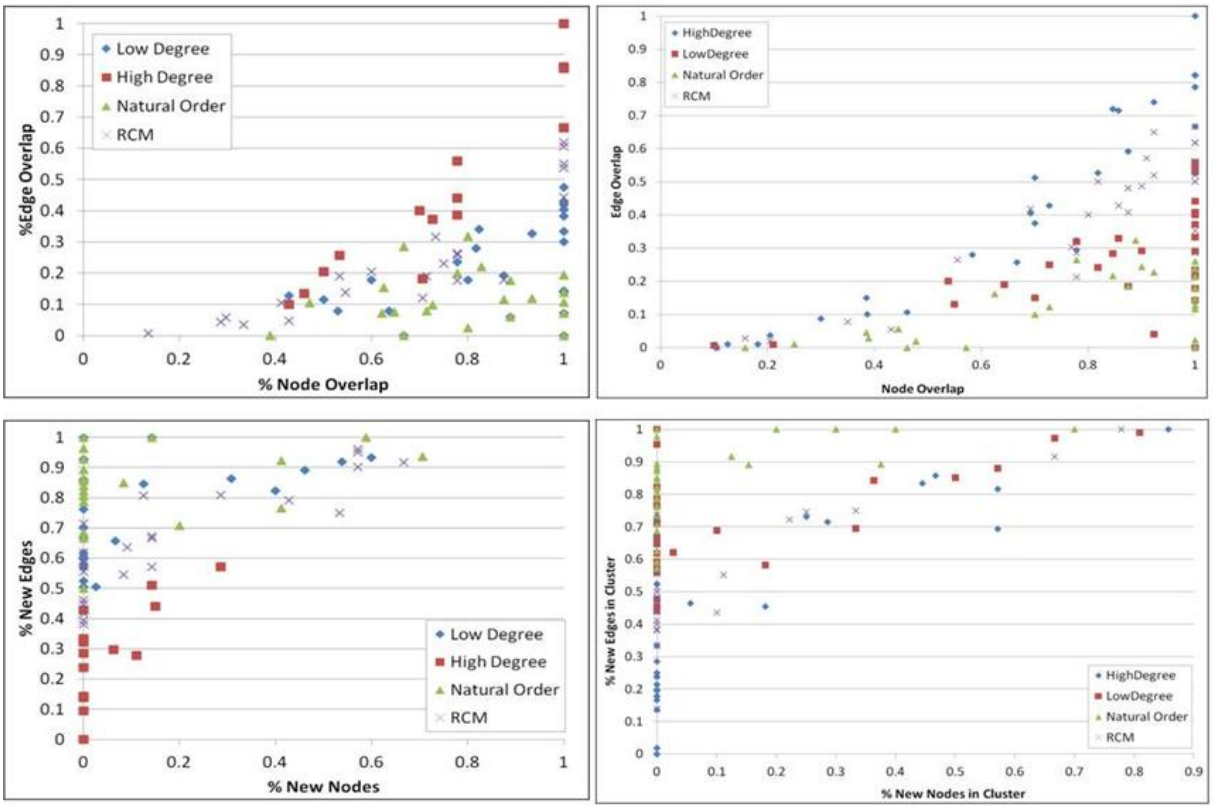


Figure 5: Node and edge overlap for GSE5140 dataset, original vs. sampled networks. Untreated overlap scores (upper left), creatine scores (upper right). Newly discovered nodes and edges for untreated (bottom left) and creatine (bottom right).

Many of the clusters found in the CRE and UNT network have little biological relevance (scores around 3 or lower). However, the clusters with biological relevance are easily maintained (based on high edge number) and identifiable across filters. Figure 5 (top) depicts the overlap of filtered clusters with original clusters in terms of percentage of node overlap and percentage of edge overlap. Each point represents a cluster found for a particular filter that had some overlap with a cluster in the original network. Points at lying near the right and the top have higher overlap. Although the filtering method removes edges, we still found some filters to leave complete clusters (100% edge and node overlap) from the original. Figure 5 (bottom) depicts clusters that were not found in original network. Points lying near the left and the bottom have less overlap. While these figures note the density of discovered clusters, it remained to be seen whether these newly found clusters were actually biologically relevant.

We observe that many points on the graph lie on the same coordinates indicating that the despite different orderings chordal-based filters retain many important clusters. This result confirms out hypothesis $H_{0b.}$ Among the orderings we see that *high and low degree orderings retain the maximum number of clusters from the original networks and natural order seems to be the best identifier of new clusters, followed by RCM.*

Figures 6 and 7 show the relevance of the clusters found in the original network. by examining node (and edge) overlap versus AEES. Node overlap seems to better identify known clusters with relevance (of which there are few) when looking at original vs. filtered overlap. The edge overlap measure seems to be a better indicator of noisy clusters (of which there are many). This is counterintuitive because the chordal method actually removes edges and we will explore this phenomena further exploring in future work.

Next, we examine the sensitivity and specificity of our ordering methods. By using our method of identifying TP, FP, FN, and TN we are able to identify rates of sensitivity and specificity for our methods of node and edge overlap. We see in Figure 8 that identifying clusters by percentage of node overlap returns a high sensitivity and low specificity, that is we find many meaningful clusters but also find many non-meaningful clusters. Edge overlap shows the opposite; specifically that using edge overlap to define a cluster match from original to filter allows us to find clusters that are likely to be noise, although the reasoning behind this is not clear. In the future we hope to use these results to better identify meaningful clusters and perhaps use this method of assessment as a secondary filter or sampling.

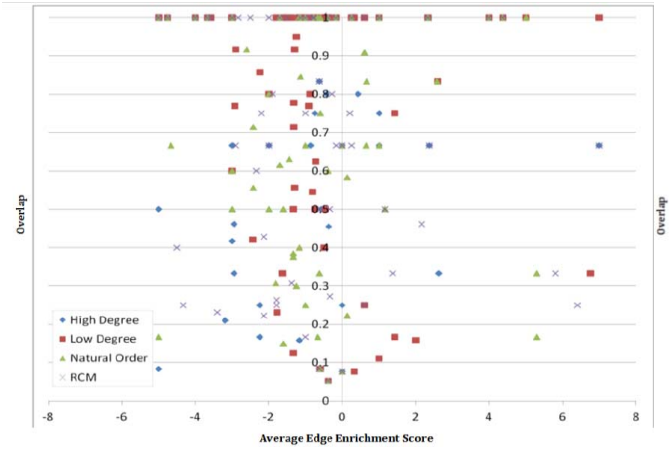


Figure 6: Node Overlap results for all networks. Each dot on the graph represents a cluster from one of the four network filters (HD, LD, NO, RCM) and the node overlap from 0.00 to 1.00 from clusters in the original networks. Lost and found clusters not included in this graph. Y-axis represents node overlap, x-axis represents average edge enrichment score for the filtered cluster.

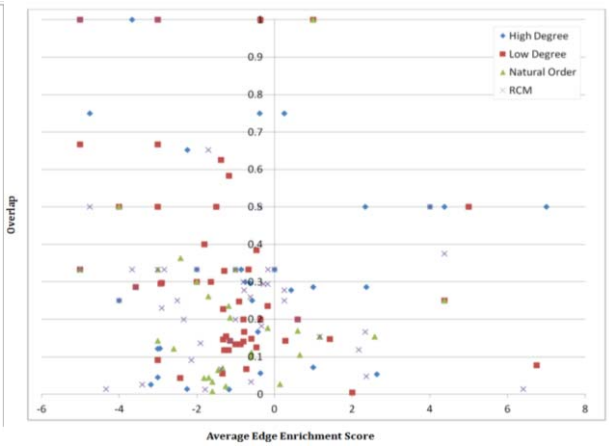


Figure 7: Edge Overlap results for all networks. Each dot on the graph represents a cluster from one of the four network filters (HD, LD, NO, RCM) and the node overlap from 0.00 to 1.00 from clusters in the original networks. Lost and found clusters not included in this graph. Y-axis represents edge overlap; x-axis represents average edge enrichment score for the filtered cluster.

Finally, we see that filters can *improve on AEE score of original clusters and allow the true function to stand out* (Figure 9). This original cluster did not stand out in the ranked list but stood out in all 4 filtered networks as a high AEE scored cluster with high overlap (66.7% node overlap, 28% edge overlap) to original and was found to be involved in regulation of apoptosis in the UNT network. Apoptosis is a critical process for normally functioning cells; when apoptosis is not regulated appropriately it can result in uncontrolled cell growth (cancer) or too much cell death (necrosis).

C. Parallel Results

In the context of parallel results we look at two factors—(i) whether the results are scalable over large number of processors and (ii) whether the data distribution affects the analysis of the results.

*Scalability***.** We demonstrate the scalability of our parallel chordal-graph based sampling algorithm. Our experiments were performed on the Firefly Cluster at the Holland Computing Center. Firefly is a Linux-based system comprising of AMD quad- and dual-core processors. Our implementation was based on a distributed memory approach using MPI. We compared the scalability of the following three sampling algorithms: (i) chordal-graph based sampling using communication, (ii) chordal graph based sampling without communication, and (iii) random walk.

Figure 10 shows the execution time for sampling two representative gene correlation networks. The smaller network is the YNG dataset with 5,348 vertices and 7,277 edges. The larger network is the CRE dataset. It is significantly larger and has 27,896 vertices and 30,296 edges. As expected the random walk filter is the most scalable of all and also the fastest. Chordal sampling without communication is also very scalable and takes less time than the version with communication. For the smaller network YNG, the scalability curve for chordal graph with communication rises sharply at 32 processors. Although the same algorithm maintains perfect scalability for the larger graph CRE, it requires more computation time (about two times as much in the case of two processors) as compared to the newer version that does not require any communication.

We compare the results of the original networks to two different types of the new chordal based filter: sequential (1P) and multiple processors (64P). To show that parallel implementation of our method does not negatively affect cluster identification, we present the node/edge overlap of clusters at the CRE Natural Order(NO) ordering at 1P and 64P in Figure 11 (left) and also the top clusters (AEES score > 3.0) in Figure 11 (right). We see that in Figure 11 (left) the method at 64P is comparable to the method at 1P, although the clusters found at 64P have better node overlap (no clusters have less that 40% node overlap) and moderate edge overlap (no better than 50% edge overlap with

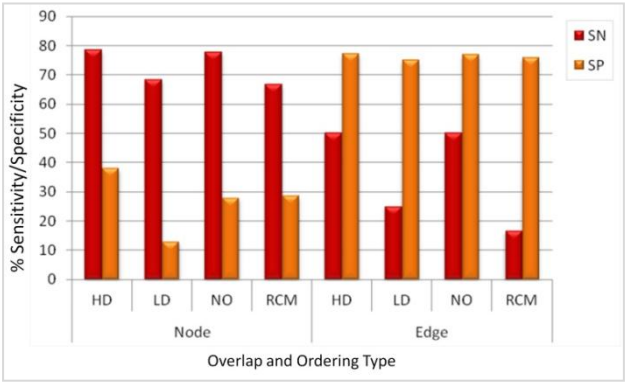


Figure 8: % Sensitivity and specificity (y-axis) of filters for node and edge overlap based on TP, FP, FN, and TN counts using node and edge overlaps (x-axis) in clusters with overlap in original networks.

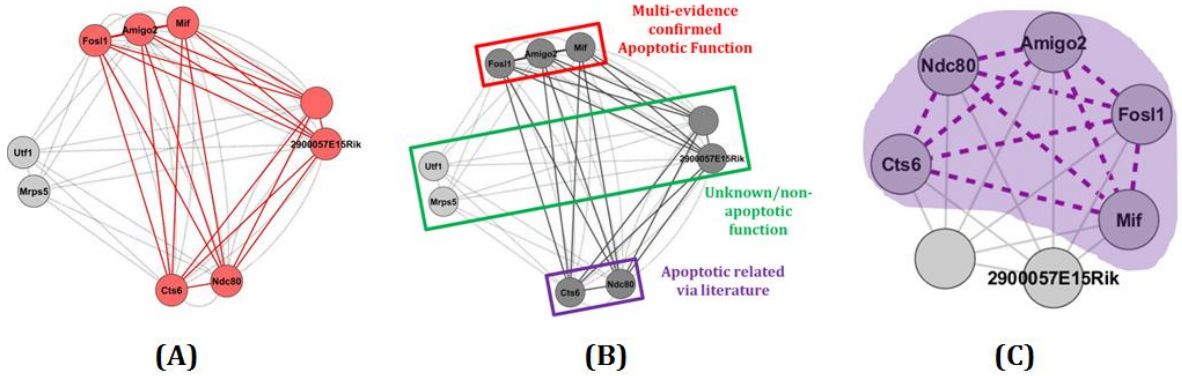


Figure 9: Example of how filtering impacts a cluster. (A) Entire cluster represents cluster 18 of original UNT network, AEES score of 2.33. Red nodes and edges represent the sampled UNT High Degree cluster #10 with AEES score of 4.17, an improvement of almost 2.00 enrichment points on average. (B) The resulting filtered cluster was annotated involvement in apoptotic function; three nodes have been confirmed as having roles in apoptosis via multiple sources (MGI, NCBI, GO, etc.), two nodes have been confirmed in the GO tree and in literature, and two remaining in the filtered network (and additional two in the original network) have not previously been identified as having apoptotic function. By filtering the sample, two nodes with no apoptotic function are removed and the cluster's true function is revealed. (C) The UNT HD cluster #10 with edges enriched in apoptosis as the DCP highlighted in purple dashed lines.

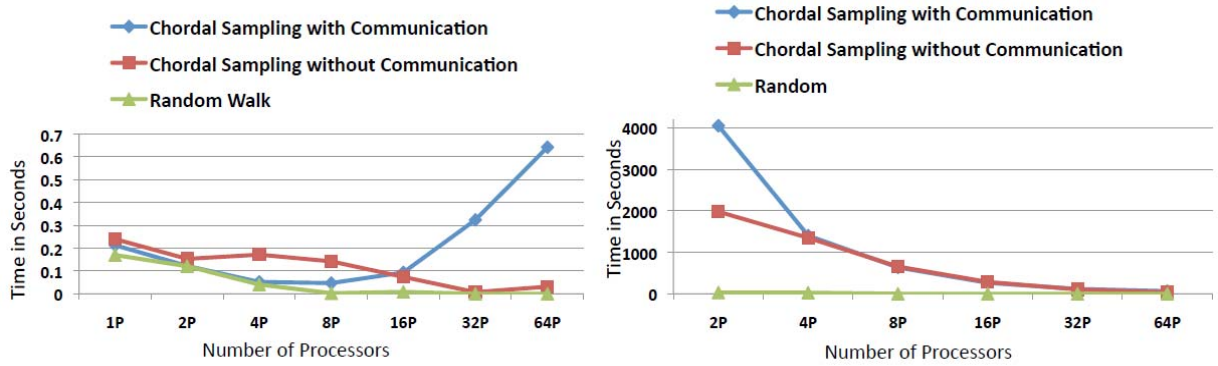


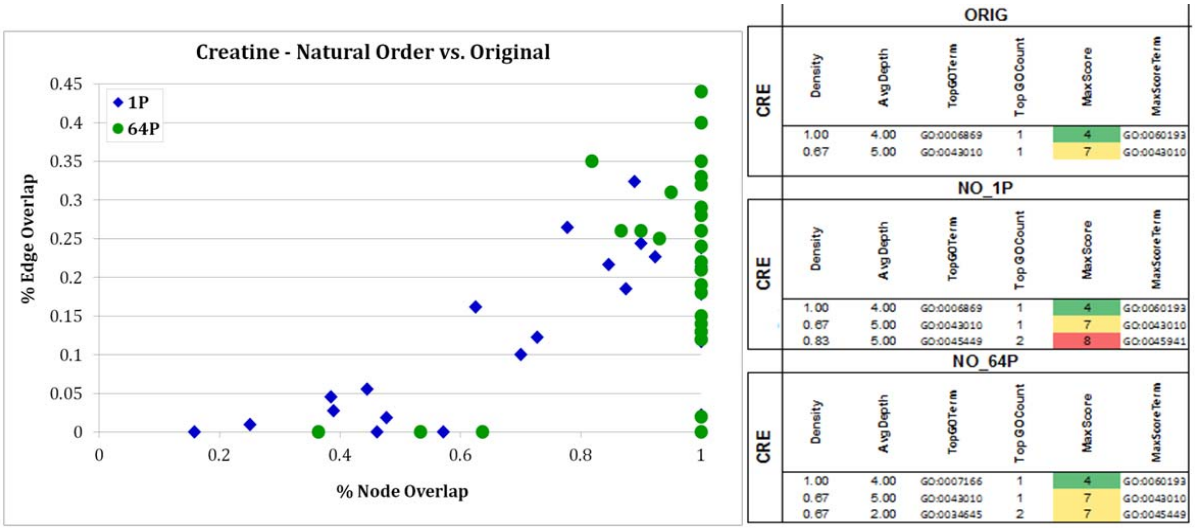
Figure 10: Scalability of sampling algorithms. Random walk sampling is the fastest and very scalable, as is chordal sampling without communication. Scalability for chordal sampling with communication deteriorates for small graphs. For large graphs the time taken can be up to twice that required for the algorithm without communication. The Y-axis gives the time in seconds and the X-axis the number of processors.

original clusters). In Figure 11 (right) we compare the top clusters for each example (original CRE, CRE NO 1P, CRE NO 64P) and find that the original clusters are maintained and both methods at 1P and 64P identify a new cluster; in this case the new cluster identified is consistent among the different processors. These results combined with the scalability of our improved method confirm our hypothesis H_{0C} .

We originally stated in H_0 that given a graph G representing a correlation network obtained from gene expression data, a maximal chordal subgraph G1 of G would preserve most of the highly dense subgraphs of G while excluding edges representing noise-related relationships in the network. We show that our method performs in this way by highlighting the properties stated here: We have shown that filters based on identifying the maximal chordal subgraph performs better than standard control filters in preserving key dense subgraphs in our studs – random walk sampling identified *no* clusters in filtered networks and thus maintained no subgraphs of interest (H_{0a}). We show that while there are some differences in the performance of the High Degree, Low Degree, Natural Order and RCM orderings, the overall impact on identification of biologically relevant clusters was that we were able to consistently identify meaningful subgraphs (H_{0b}) and furthermore, we could identify new clusters. Finally, we address H_{0c} and note in our final set of results that parallel implementation of our filtering method does not negatively impact our results and consistency in clusters by varying the number of processors is maintained.

V. Conclusions

Networks represent a class of models with striking potential and ease of use for identifying biological functionality by modeling *relationships* and allowing for inspection of biological mechanisms at the systems level. In this work we propose a novel method for filtering data using graph theoretic strategies that not only maintains structures from original network models but also reduces complexity by removing noise. We show that the maximal chordal subgraph filter



ORIG

	Density	Avg Depth	TopGOTerm	Top GOCount	MaxScore	MaxScoreTerm
CRE	1.00	4.00	GO:0006869	1	4	GO:0060193
	0.67	5.00	GO:0043010	1	7	GO:0043010

NO_1P

	Density	Avg Depth	TopGOTerm	Top GOCount	MaxScore	MaxScoreTerm
CRE	1.00	4.00	GO:0006869	1	4	GO:0060193
	0.67	5.00	GO:0043010	1	7	GO:0043010
	0.83	5.00	GO:0045449	2	8	GO:0045941

NO_64P

	Density	Avg Depth	TopGOTerm	Top GOCount	MaxScore	MaxScoreTerm
CRE	1.00	4.00	GO:0007166	1	4	GO:0060193
	0.67	5.00	GO:0043010	1	7	GO:0043010
	0.67	2.00	GO:0034645	2	7	GO:0045449

Figure 11: Left: %Edge/Node overlap comparison of clusters at 1P and 64P. Each point represents a cluster found in the ORIG network and its %E/N overlap with a filtered network cluster. Right: Clusters with AEES scores >3.0 found in ORIG, 1P and 64P. Average depth is the AEES score; Max Score is the depth of the deepest term in the cluster.

outperforms the random walk control, and furthermore, our chordal graph method removes noise such that new structures hidden in the original networks are revealed. Reported results also show that our parallel implementation is scalable and the analysis results are not significantly affected by data distribution. This approach highlights another step in gaining ability to analyze complex large-scale biological data using network modeling. This work also emphasizes the need for innovative integration of high-performance computing in the domain of bioinformatics research.

ACKNOWLEDGEMENT

This publication was made possible by the College of Information Science and Technology, University of Nebraska at Omaha and Grant P20 RR16469 from the NCRR, a component of the National Institutes of Health.

REFERENCES

[1] Rafiei, D. (2005) Effectively visualizing large networks through sampling. Visualization, VIS 05, IEEE.

[2] Gilbert, AC and Levchenko, K. (2004) Compressing network graphs. In LinkKDD.

[3] Airoldi, EM and Carley, KM. (2005). Sampling algorithms for pure network topologies. SIGKDD Explorations, 7(2).

[4] Leskovec, J and Faloutsos, C. (2006). Sampling from large graphs. Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD'06.

[5] Arun, SM and Berger Wolf, TY. (2011). Benefits of Bias: Towards better Characterization of network sampling. Proceedings of KDD'11.

[6] Duraisamy, K, Dempsey, K, Ali, H, and Bhowmick, S. (2011). A noise reducing sampling approach for uncovering critical properties in large scale biological networks (2011). *High Performance Computing and Simulation 2011 International Conference (HPCS):* July 4-8. Istanbul, Turkey.

[7] Dempsey K, Duraisamy, K, Ali, H, and Bhowmick S. (2011). A parallel graph sampling algorithm for analyzing gene correlation networks (2011). *International Conference on Computational Science 2011*. June 1-3. Singapore.

[8] Dearing, PM, Shier, DR, and Warner, DD. (1988). Maximal Chordal Subgraphs. Discrete Applied Mathematics 20(3):181-190.

[9] Dempsey, K, Thapa, I, Bastola, D, and Ali, H. (2011) Identifying Modular Function via Edge Annotation in Gene Correlation Networks using Gene Ontology Search. *2011 BIBM Workshop on Data Mining for Biomarker Discovery:* November 2011. Atlanta, GA

[10] Reverter, A, & Chan, EK. (2008). Combining partial correlation and an information theory approach to the reversed engineering of gene co-expression networks. Bioinformatics (Oxford, England), 24(21), 2491-2497.

[11] Watson-Haigh, NS, Kadarmideen, HN, & Reverter, A. (2010). PCIT: An R package for weighted gene co-expression networks based on partial correlation and information theory approaches. Bioinformatics (Oxford, England), 26(3), 411-413.

[12] Ewens, WJ, & Grant, GR. (2005). Statistical methods in bioinformatics (Second Edition ed.). New York, NY: Springer.

[13] Zhang, B, & Horvath, S. (2005). A general framework for weighted gene co-expression network analysis. Statistical Applications in Genetics and Molecular Biology, 4, Article17.

[14] Dong, J, & Horvath, S. (2007). Understanding network concepts in modules. BMC Systems Biology, 1, 24.

[15] Carter, SL, Brechbuhler, CM, Griffin, M, & Bond, AT. (2004). Gene co-expression network topology provides a framework for molecular characterization of cellular state. Bioinformatics (Oxford, England), 20(14), 2242-2250.

[16] Opgen-Rhein, R, & Strimmer, K. (2007). From correlation to causation networks: A simple approximate learning algorithm and its application to high-dimensional plant gene expression data. BMC Systems Biology, 1, 37.

[17] Verbitsky, M, Yonan, AL, Malleret, G, Kandel, ER, Gilliam, T C, & Pavlidis, P. (2004). Altered hippocampal transcript profile accompanies an age-related spatial memory deficit in mice. Learning & Memory (Cold Spring Harbor, N.Y.), 11(3), 253-260.

[18] Benson, M, & Breitling, R. (2006). Network theory to understand microarray studies of complex diseases. Current Molecular Medicine, 6(6), 695-701.

[19] Barabasi, AL, & Oltvai, ZN (2004). Network biology: Understanding the cell's functional organization. Nature Reviews.Genetics, 5(2), 101-113.

[20] Pavlopoulous, G, Secrier, M, Charalampos, N, Moscholpoulous, Soldatos T, Kossia S, Aerts, J, Schneidder, R, Bagos, P. (2011) Using graph theory to analyze biological networks. BioData Min, (4)10.

[21] Bender A, Beckers J, Schneider I, Hölter SM et al. Creatine improves health and survival of mice. *Neurobiol Aging* 2008 Sep;29(9):1404-11. PMID: 17416441

[22] Yoon, J and Jung, WH. (2011). A GPU-accelerated bioinformatics application for large-scale protein interaction netowkrs. APBCPresentation.