



University of Nebraska at Omaha
DigitalCommons@UNO

Faculty Books and Monographs

2015

Bioinformatics and Biomedical Engineering

Francisco Ortuño

Ignacio Rojas

Kathryn Dempsey Cooper

University of Nebraska at Omaha, kdempsey@unomaha.edu


Sachin Pawaskar

University of Nebraska at Omaha, spawaskar@unomaha.edu

Hesham Ali

University of Nebraska at Omaha, hali@unomaha.edu

Follow this and additional works at: <http://digitalcommons.unomaha.edu/facultybooks>

 Part of the [Bioinformatics Commons](#), and the [Computer Sciences Commons](#)

Recommended Citation

Ortuño, Francisco; Rojas, Ignacio; Cooper, Kathryn Dempsey; Pawaskar, Sachin; and Ali, Hesham, "Bioinformatics and Biomedical Engineering" (2015). *Faculty Books and Monographs*. Book 323.

<http://digitalcommons.unomaha.edu/facultybooks/323>

This Book is brought to you for free and open access by DigitalCommons@UNO. It has been accepted for inclusion in Faculty Books and Monographs by an authorized administrator of DigitalCommons@UNO. For more information, please contact unodigitalcommons@unomaha.edu.



Identification of Biologically Significant Elements using Correlation Networks in High Performance Computing Environments

Kathryn Dempsey Cooper, Sachin Pawaskar, and Hesham Ali

College of Information Science & Technology, University of Nebraska at
Omaha, NE 68182 USA
{kdempsey, spawaskar, hali}@unomaha.edu

Abstract. Network modeling of high throughput biological data has emerged as a popular tool for analysis in the past decade. Among the many types of networks available, the correlation network model is typically used to represent gene expression data generated via microarray or RNAseq, and many of the structures found within the correlation network have been found to correspond to biological function. The recently described gateway node is a gene that is found structurally to be co-regulated with distinct groups of genes at different conditions or treatments; the resulting structure is typically two clusters connected by one or a few nodes within a multi-state network. As network size and dimensionality grows, however, the methods proposed to identify these gateway nodes require parallelization to remain efficient and computationally feasible. In this research we present our method for identifying gateway nodes in three datasets using a high performance computing environment: quiescence in *Saccharomyces cerevisiae*, brain aging in *Mus Musculus*, and the effects of creatine on aging in *Mus musculus*. We find that our parallel method improves runtime and performs equally as well as sequential approach.

Keywords: high performance computing, parallel algorithms, correlation networks, gateway nodes

1 Introduction

As the popularity of network modeling for big biological data grows, the need for algorithms and methods that can analyze these data grows with it. Network modeling in biological data came of age in 2001 with the finding of small world property in complex system [12]; protein-protein interaction networks were one of the models analyzed. Then came the structure-function correspondence: in PPI's, hub nodes are speculated to be linked with essential genes or proteins [3, 11, 12]; nodes in a clique tend to correspond to proteins in complex [3,7,10,16], and the disassortativity of hubs could suggest that hub proteins are ancestral in nature [17]. The correlation network, where genes are represented as nodes, finds some measure of correlation between gene expression patterns to determine a relationship [13]. For example, linear relationships can be captured by the Pearson Correlation coefficient; networks built using this

measure have been found to tend toward assortativity [17], to have a lower hub lethality rate [5], and to contain clusters whose manipulation suggests that the expression system is robust to minor changes [6,7].

The goal of the identification of gateway nodes is to identify the key genes in mechanistic changes between states. Gene expression experiments, particularly where sample size is large, provide an ideal experimental setup where comparison of states (treated, untreated or different time points) can occur while other key parameters are held consistent (tissue type, organism type and strain, etc). As such, in this research, we identify three datasets and the gateway nodes between the states found within them. Then, we take this gateway node analysis approach and parallelize it.

The recent integration of high performance computing approaches and bioinformatics or biomedical informatics methods approaches have allowed for massive strides in systems biology, or the identification of the mechanistic dynamics of a system as a whole. Previous work in this area, for example, has improved sequence assembly via Energy Aware parallelization, which minimizes energy and computational resources while improving runtime [21]. This marriage of computing and biological expertise is critical in the advancement of technologies designed to diagnose and prevent diseases, and as such, continued research in this area is critical.

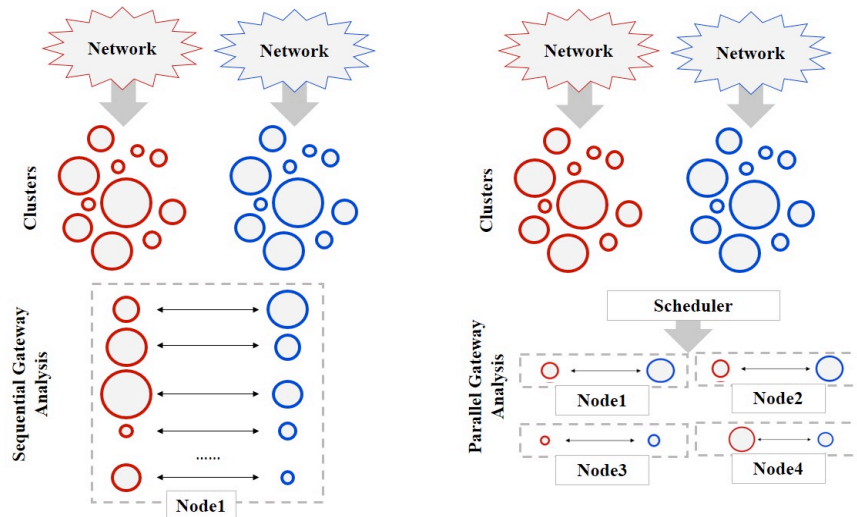


Figure 1. The sequential versus naively parallel gateway nodes analysis. On the left, we have two networks, which after clustering, need to have a sequential pairwise comparison of clusters. In the parallel approach, a scheduler (the master node) takes the number of jobs and distributes them evenly among nodes (worker nodes).

2 Experimental Suite

In this research, three datasets are presented to highlight the computational and biological power of the parallel gateway analysis. Known datasets were drawn from NCBI's Gene Expression Omnibus [9]. The first uses a model organism, *Saccharomyces cerevisiae*; this dataset is chosen for the vast array of knowledge available about the yeast organism, which allows for a more confident biological assessment of the gateway node functionality without actually performing any experiments *in vivo*. The second, GSE5078, is one of the datasets used in the original gateway node analysis; this dataset is used largely to determine if the same gateway nodes are identified sequentially versus in parallel. The final dataset is chosen for its large network size (more specifically, larger number of clusters) to highlight the scalability of the parallel method.

- *GSE5078*: Generated by Verbitsky *et al.* 2004 [14]; this dataset includes expression data from B1/6 mice hippocampus separated into two groups: Young (YNG), at 2 months, and Middle-Aged (MID) at 15 months. Both sets have 9 samples.
- *GSE8542*: Generated by Aragon *et al.* 2008 [18]; this dataset includes expression data from BY4742 yeast separated into two groups: quiescent (QUI) or non-quiescent (NON).

RandomClique: Six sets of "clusters" made by random generation of 100 cliques between the sizes of 5 and 100 nodes. The clusters were grouped into six sets, R1, R2, R3, R4, R5, and R6, all consisting of 100 cliques each. Comparisons of the faux networks were performed in the following matchups: R1 vs. R2 (R1-R2), R3 vs. R4 (R3-R4), and R5 vs. R6 (R5-R6).

2.1 Network Creation and Manipulation

Networks were created by pairwise calculation of the Pearson Correlation coefficient (as described in [8]) with a correlation (ρ) threshold of 0.85 to 1.00; hypothesis testing was performed using the Student's t-test and values with p-value > 0.0005 were thrown out. The resulting network uses gene probes as nodes and correlated expression patterns as edges. As a creation quality check, the networks were checked for duplicate and self-edges; none were found.

Clustering of the networks was performed with AllegroMCODE v.1.0 [16]. Nodes with degree less than 15 were not used in cluster finding, and a scoring cutoff of 0.2 (the default) was used. Clusters with a minimum K-core of 10 were found using a maximum search depth of 10.

2.1 Gateway Node Identification

Per each dataset, gateway nodes are calculated as described in Dempsey 2014 and briefly here: Networks are first clustered to identify the dense groups within the network, and then the clusters are compared to determine if any nodes are shared

between them. If nodes are shared, the number of edges between them and the clusters they connect are determined to calculate a gatewayness score. This gatewayness score is calculated as:

$$gatewayness_{nodeA} = \frac{degree_{nodeA}}{degree_{all\ gateway\ nodes}} \text{ (Equation 1)}$$

In this equation the gateway node A being studied is defined as any node shared between two clusters of a different state and the total degree of all gateway nodes is the sum of the degree of any node shared between two clusters of a different state. If node A is the only gateway node between two clusters 1 and 2 and has a degree of 50, the gatewayness score will be $50/50 = 1.00$, or 100%. If there are two gateway nodes A and B, where the degree of A is 45 and the degree of B is 55, the gatewayness of A will be $45/(45+55) = 0.45$ or 45%, and the gatewayness of B will be $55/(45+55) = 0.55$ or 55%. Thus, gatewayness is a measure of the responsibility of a node's connectivity between two clusters of a different state.

One way to reduce the runtime of the gateway nodes analysis in large networks is by only allowing clusters of a certain density to be analyzed; for example, if a network has 100 clusters, a density filter can be imposed (say, where the edge density of the cluster is used to remove clusters); in previous studies, using a cluster density filter of 65% can remove up to 60% of the clusters analyzed. However, it is most beneficial to compare all possible clusters instead of imposing further restriction (and thus possibly removing more biological information), which our parallel algorithm approach allows for.

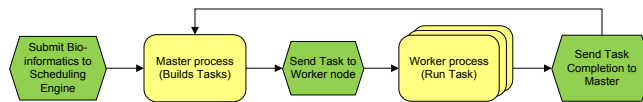


Figure 2: Parallel implementation process flow diagram

2.3 High Performance Computing Environment

The gateway node analysis is an easily parallelizable problem – the algorithm takes a pair of clusters, compares the nodes between them, and when nodes overlap between clusters, calculates the edge intersection between the two clusters. The runtime for this analysis increases in linear time increases when the size, density, or number of clusters increases. However, the problem can be scheduled to different processors by simply determining how many comparisons need to be made and then delegating them to respective worker nodes from one master.

As shown in Figure 1, the sequential approach and parallel approach differ only in the determination of gateway nodes. First, networks are created or downloaded (the networks are assumed to originate from the same set of probes – genes, gene products, proteins, etc., or such that nodes can be paired together according to some mapping function). Next, networks are clustered – using any type of clustering function desired – and the resulting clusters are forwarded to the gateway analysis. In this study, we use a specific clustering approach known for its identification of small,

dense clusters (MCODE), but any type of clustering can be made. Since this approach borrows from previous studies, the same clustering method used in Dempsey *et al.* 2013 was used for comparison. Finally, the gateway analysis approach uses anywhere from 1-64 nodes to identify gateway nodes using code written in Perl.

```

Int main(int argc, char **argv)
{
    int rank;
    MPI_Init(&argc, &argv);
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
    if (rank == 0) {
        Init();
        exec_master(); // Builds tasks & sends to worker
    } else {
        exec_worker();
    }
    MPI_Finalize();
}

static void exec_worker(void) {
    char rundate[16], runtime[16], cmd[256];
    Work work;
    MPI_Status status;
    while (true) { // Receive a message from the master
        MPI_Recv(&work, 1, Worktype, MASTER, MPI_ANY_TAG, MPI_COMM_WORLD, &status);
        if (status.MPI_TAG == EXITTAG) { // Check tag of the received message.
            return;
        }
        pBIProg->BuildCommandString(&work, cmd);
        ExecuteTask(cmd);

        // Send the result back to the master task
        strcpy(work.sNodeName, sProcessorName);
        work.iNode = RANK;
        strcpy(work.sRunDate, rundate);
        strcpy(work.sRunTime, runtime);
        work.iET = sw.ElapsedTime();
        MPI_Send(&work, 1, Worktype, MASTER, WORKTAG, MPI_COMM_WORLD);
    }
}

```

Figure 3: Pseudo-code of parallel implementation

2.4 Parallel Implementation

The input dataset, consists of cluster files as mentioned above which are stored in their respective directories. Let us say that Organism1 cluster files are in Dir1 and contains *m* cluster files, and Organism2 cluster files are in Dir2 and contain *n* cluster files. The scheduling engines master process reads creates tasks for gateway analysis by comparing these files against each other. It takes two clusters as input and outputs any gateway nodes and their scores; a wrapper is used sequentially to run the script and deliver all possible combinations of clusters. The Big O of our parallel approach is $O(m*n)$. The master thread sends each task with the 2 files as input to worker processors running gateway analysis algorithm. The master thread manages the execution order of the gateway analysis step. Figure 2 below shows the process flow of our parallel implementation and the pseudo Code of this implementation is shown Figure 3. The code was implemented on the Tusker Cluster described below as well. Tusker is a 40 TF cluster consisting of 106 Dell R815 nodes using AMD 6272 2.1GHz processors, connected via Mellanox QDR Infiniband and backed by

approximately 350 TB of Terascale Lustre-based parallel filesystem. All experiments were run on this cluster.

TABLE I. TOP TEN GATEWAY NODES FOR MOUSE AND YEAST NETWORKS

Network	ID	Dataset	Nodes	Edges	Density	Clusters	Clustering Runtime
Young	YNG	GSE5078	12368	72967	0.095%	35	31.434 seconds
Middle-Aged	MID	GSE5078	12340	79176	0.104%	36	20.298 seconds
Non-quiescent	NON	GSE8542	1541	2515	0.212%	11	1.793 seconds
Quiescent	QUI	GSE8542	2543	5363	0.166%	62	2.671 seconds

3 Results

The results of our naively parallel gateway node analysis study are below. Table 1 describes the network sizes, edge density, number of clusters, clustering parameters, and clustering runtime. While the numbers of nodes and edges differ greatly due to difference in genome sizes, the density of the networks are relatively similar, and all networks are sparse. Using the same parameters to identify clusters in each network reveals a similar number of clusters in the mouse network (35 in the YNG and 36 in the MID) compared to the yeast network which has a more varied number (11 in the NON and 62 in the QUI). Clustering runtime appears to have no relationship with density, but rather seems to be linked to overall network size via edge count.

TABLE II. TOP TEN GATEWAY NODES FOR MOUSE AND YEAST NETWORKS

MOUSE - 0% Density		MOUSE - 65% Density		YEAST - 0% Density	
Gene ID	Gatewayness Score:	Gene ID	Gatewayness Score:	Gene ID	Gatewayness Score:
Map3k2	100.00%	Sla	100.00%	MCM21	33.33%
Pira1	100.00%	Matn3	100.00%	CPR5	33.33%
Ace	100.00%	Dio1	100.00%	TIM11	33.33%
Cts7	100.00%	Fbp1	100.00%	YGR164W	33.33%
Six3	100.00%	Ceacam12	100.00%	CBP4	33.33%
Immp11	100.00%	Ptpnb	100.00%	RPL1B	33.33%
Ythdf2	100.00%	Plin4	100.00%	GTR2	25.00%
Krt25	100.00%	Cldn1	100.00%	HGH1	25.00%
Tsks	100.00%	Akr1c21	100.00%	CRH1	25.00%
Vil1	100.00%	Ltc4s	100.00%	CLC1	25.00%

3.1 Model Organism – *S. cerevisiae* gateway nodes

There were 97 gateway nodes identified in the sequential and all parallel runs of the yeast network dataset; there were no gateway nodes with a score of 1.00. The gateway nodes identified in each respective run did not change with processor number. The density threshold used for yeast was 0%, meaning that any clusters that overlapped with one another were considered. While the yeast networks are relatively small, in larger networks, this all to all comparison with no density filter is desired. A density filter is typically used to reduce the amount of clusters to compare to improve runtime of gateway identification, but via naïve parallelization of the approach, all clusters can be compared. Further, this can be used to determine the distribution of gateway nodes and their relative functional impact according to cluster density, if such a relationship exists.

Gene list analysis of the gateway nodes [15] was performed using PantherDB's tool (version 8.1) [19, 20]. Gateway nodes were functionally classified according to Molecular Function, Biological Process, and Pathway. The results of these classifications are shown in Figures 4 and 5. The classifications of genes in terms of Molecular Function (MF) and Biological Process (BP) are largely standard with the majority of genes involved in metabolic processes and catalytic activity (the profile of BP and MF classification in the mouse dataset is very similar – see Figure 4). However, in the pathway classification set, telling evidence of gateway biological impact emerges. The EGF receptor signaling pathway has been implicated as an upstream regulator in astroglial cells in the transition from quiescence to reactivity [1]. The PDGF signaling pathway plays a similar role; stimulation of cell growth and proliferation; quiescence stems out of the metabolism by activation of certain elements [2]. Glycolysis, the third most pathway identified via the gateway node classification, plays a major role in the shift from non-quiescence to quiescence. Glucose levels available in media can be used to stimulate the shift from non-quiescence to quiescence; [2] suggests that this is due to the inherent changes caused in glucose metabolism when glucose is lacking or present in media.

3.2 Known dataset – GSE5078 gateway nodes

There were 172 gateway nodes identified in the sequential and all parallel runs of the mouse network dataset at 0% density threshold in mice; for the 65% density threshold, 25 gateway nodes were identified. In parallel and sequential runs for both parameterizations, all gateway nodes matched. Functional classification of gateway nodes at 0% density threshold and 65% density thresholds are shown in Figures 5 and 6. The gateway nodes identified at 65% match up with those identified in Dempsey *et al.* The gateway nodes identified within this dataset have been found to be related to aging. One example of this is Klotho and Ins2 (not listed in the top 10 gateway nodes, shown in Table 2), which are involved in the insulin signaling pathway, which has long been known to be involved in biological aging.

Of the top twenty gateway nodes identified in the mouse datasets for 0% and 65% densities, nine (45%) are protein binding molecules (*Map3k2*, *Ace*, *Six3*, *Kr25*, *Vill*, *Sla*, *Fbp1*, *Ptprb*, and *Cldn1*), meaning that their gene products they bind with

other proteins; nearly all of the gateway genes identified are pleiotropic, or having a number of roles in the cell. This follows with the concepts proposed in Dempsey *et al.*, that gateway nodes are tied to the mechanistic changes in expression that occur to restore homeostasis in changing environments within the cell.

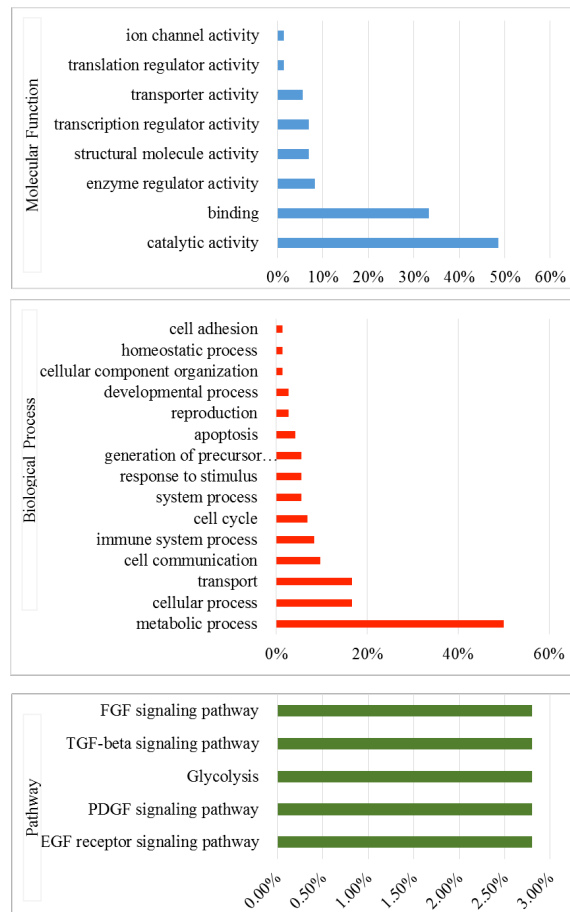


Figure 4 The functional classifications of the yeast gateway nodes at 0% density. Blue – Biological process, Red – Molecular Function, and Green – Pathway. The axis is the percentage of genes in the gateway node list with that annotation compared to the background (mouse genome).

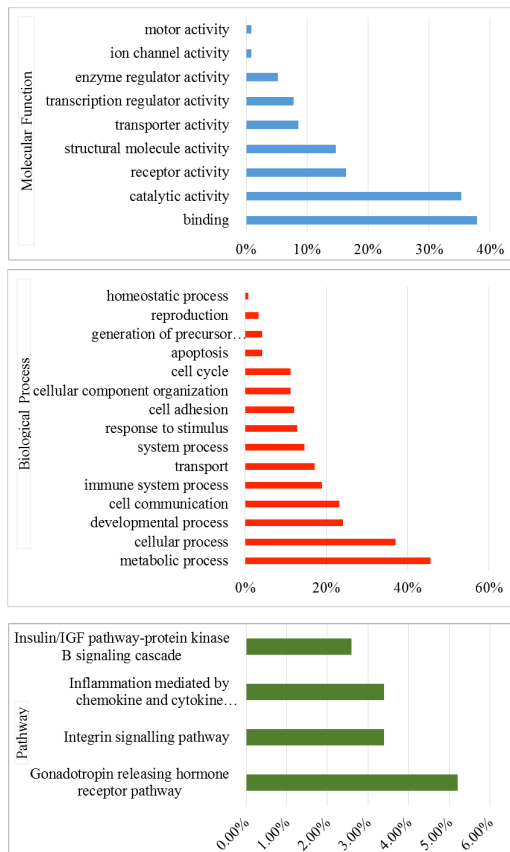


Figure 5. Left: The functional classifications of the mouse gateway nodes at 0% density. Blue – Biological process, Red – Molecular Function, and Green – Pathway. The axis is the percentage of genes in the gateway node list with that annotation compared to the background (mouse genome).

3.3 Scalability

The smaller model network analyses (mouse and yeast) both ran in minimal time sequentially – 144 seconds for yeast, 305 seconds for mouse at 0%, and 309 seconds at 65%. While this time requirement hardly calls for parallelization, extending the gateway node analysis into larger and more dimensional studies will require analysis of much larger networks and datasets at many more states. Systems biology

approaches nearly guarantee that the data available will continue. Regardless, parallelization of the gateway node analysis in these models shows good scalability, as shown in Figure 7.

The random networks are designed to represent the scalability of these larger networks, and on this larger view, the scalability of this naively parallel approach does not disappoint. For the R1-R2 analysis, the runtime takes 68 minutes using 1 processor, and 1 minute and 25 seconds using 64 processors, a speedup of 48.6. The runtime and speedup for the random runs are shown in Figures 8 and 9. The naively parallel approach described reduces runtime, particularly as networks get larger.

4 Discussion

In recent years, modeling of high throughput biological data via network or graph theoretic modeling has emerged as a popular tool for analysis. The correlation network model, used to represent gene expression data, is one of many different types of models that rely on correlation of expression patterns to form internal graph structures. One of these structures, the gateway node, has been found to represent co-

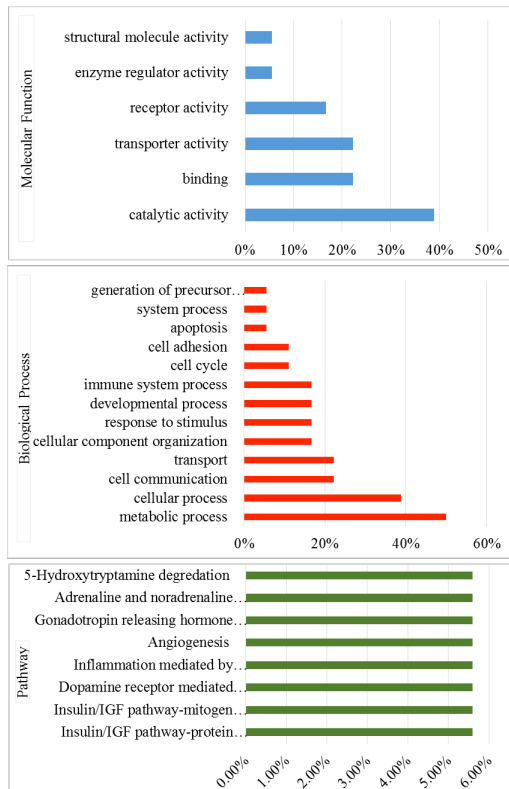


Figure 5. Left: The functional classifications of the mouse gateway nodes at 65% density. Blue – Biological process, Red – Molecular Function, and Green – Pathway. The axis is the percentage of genes in the gateway node list with that annotation compared to the background (mouse genome).

regulation with distinct groups of genes at different conditions or treatments. The structure that results typically represents 1-10% of the original network, making them a desirable target for deciphering the mechanistic changes between states or environments. As network size and dimensionality grows, however, the methods proposed to identify these gateway nodes require parallelization to remain efficient and computationally feasible. In this research we have presented our method for identifying gateway

nodes in three datasets using a high performance computing environment: quiescence in *Saccharomyces cerevisiae*, brain aging in *Mus Musculus*, and the effects of creatine on aging in *Mus musculus*. The results show that our parallel method improves runtime and performs equally as well as sequential approach, meaning that as network dimensionality and size increases, we will have the tools required to analyze the entire system.

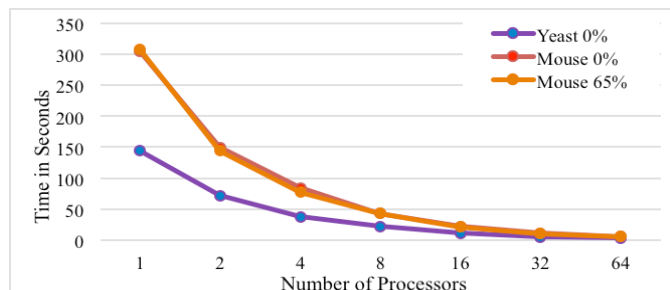


Figure 7. Scalability for the Yeast and Mouse networks. The x-axis represents the number of processors used and the y-axis represents the time in seconds.

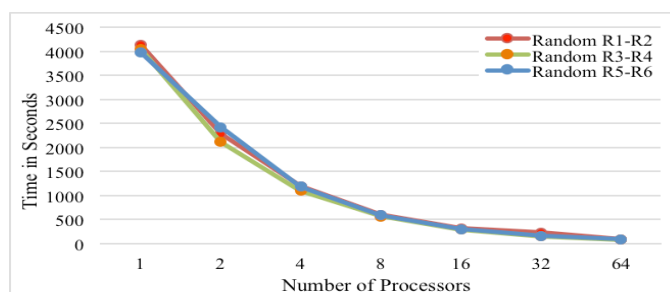


Figure 8. Scalability for the Random networks. The x-axis represents the number of processors used and the y-axis represents the time in seconds.

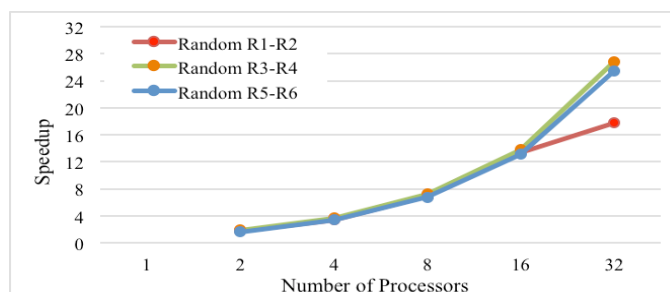


Figure 9. Speedup for the Random Networks. The x-axis represents the number of processors and the Y-axis represents speedup.

References

1. Smith, T.F., Waterman, M.S.: Identification of Common Molecular Subsequences. *J. Mol. Biol.* 147, 195--197 (1981)
2. May, P., Ehrlich, H.C., Steinke, T.: ZIB Structure Prediction Pipeline: Composing a Complex Biological Workflow through Web Services. In: Nagel, W.E., Walter, W.V., Lehner, W. (eds.) *Euro-Par 2006. LNCS*, vol. 4128, pp. 1148--1158. Springer, Heidelberg (2006)

3. Foster, I., Kesselman, C.: *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann, San Francisco (1999)
4. Czajkowski, K., Fitzgerald, S., Foster, I., Kesselman, C.: Grid Information Services for Distributed Resource Sharing. In: 10th IEEE International Symposium on High Performance Distributed Computing, pp. 181--184. IEEE Press, New York (2001)
5. Foster, I., Kesselman, C., Nick, J., Tuecke, S.: *The Physiology of the Grid: an Open Grid Services Architecture for Distributed Systems Integration*. Technical report, Global Grid Forum (2002)
6. National Center for Biotechnology Information, <http://www.ncbi.nlm.nih.gov>
1. Liu, Chen, Johns, and Neufeld. Epidermal growth factor receptor activation: an upstream signal for transition of quiescent astrocytes into reactive astrocytes after neural injury. *J Neurosci* 2006;26(28):7532-40.
2. Laporte D, Lebaudy A, Sahin A, Pinson B, Ceschin J, Daignan-Fornier B, Sagot I. Metabolic status rather than cell cycle signals control quiescence entry and exit. *J Cell Biol*. 2011 Mar 21;192(6):949-57. doi: 10.1083/jcb.201009028. Epub 2011 Mar 14.
3. Barabasi, A. L., & Oltvai, Z. N. (2004). Network biology: Understanding the cell's functional organization. *Nature Reviews Genetics*, 5(2), 101-113.
4. Bult, Cj, Eppig JT, Kadin JA, Richardson JE, Blake JA; and the members of the Mouse Genome Database Group. 2008. The Mouse Genome Database (MGD): mouse biology and model systems. *Nucleic Acids Res* 36(Database Issue):D724-8.
5. Dempsey, K. and Ali, H. On the discovery of Cellular subsystems in correlation networks using centrality measures (2014). *Current Bioinformatics*, 7(4).
6. Duraisamy, K., Dempsey, K., Ali, H., and S. Bhowmick (2011). A noise reducing sampling approach for uncovering critical properties in large scale biological networks. *High Performance Computing and Simulation 2011 International Conference (HPCS)*: July 4-8. Istanbul, Turkey.
7. Dong, J., & Horvath, S. (2007). Understanding network concepts in modules. *BMC Systems Biology*, 1, 24.
8. Ewens, W. J., & Grant, G. R. (2005). *Statistical methods in bioinformatics* (Second Edition ed.). New York, NY: Springer.
9. Edgar, R., Domrachev, M., and AE. Lash (2002). Gene Expression Omnibus: NCBI gene expression and hybridization array data repository. *Nuc Acid Res* 30(1):207-10.
10. Enright A.J., Van Dongen S., Ouzounis C.A. *An efficient algorithm for large-scale detection of protein families*. *Nucleic Acids Research* 30(7):1575-1584 (2002).
11. Hao D, Li C (2011) The dichotomy in degree correlation of biological networks. *PloS one* 6: e28322. doi: 10.1371/journal.pone.0028322.
12. Jeong, H., Mason, S. P., Barabasi, A. L., & Oltvai, Z. N. (2001). Lethality and centrality in protein networks. *Nature*, 411(6833), 41-42.
13. 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.
14. Verbitsky, M., Yonan, A. L., Malleret, G., Kandel, E. R., 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.
15. Subramanian, A., Tamayo, P., Mootha, VK., Mukherjee, S., Ebert, BL., Gillette, MA., Paulovich, A., Pomeroy, SL., Golub, TR., Lander, ES., and JP Mesirov (2005). Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proc Natl Acad Sci* 102(43):15545-15550.
16. Yoon, JS and WH Jung (2011). A GPU-accelerated bioinformatics application for large-scale protein interaction networks. APBC poster presentation.
17. Newman, MEJ. (2002). Assortative Mixing in Networks. *Phys Rev Lett*, 89(20):208701.
18. Aragon AD, Werner-Washburne M (2008). Characterization of differentiated quiescent and non-quiescent cells in yeast stationary-phase cultures. *Mol Biol Cell*, 19(3):1271-80.
19. Miu H, Muruganujan A, and Thomas P (2012). PANTHER in 2013: modeling the evolution of gene function, and other gene attributes, in the context of phylogenetic trees. *Nucl Acids Res*, 41(Database issue):D377-86.
20. Thomas P, Kejariwal A, Guo N, Mi H, Campbell MJ, Muruganujan A, Lazareva-Ulitsky B. Applications for protein sequence-function evolution data: mRNA/protein expression analysis and coding SNP tools. *Nuc Acids Res*, 34(Suppl2):W645-W650.
21. Pawaskar, S., Warnke, J., Ali, H. An energy-aware bioinformatics application for assembling short-reads in high performance computing systems. *HPCS 2013 Proceedings IEEE* (2012):154-160.