2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress

Scalable Mining and Analysis of Protein-Protein Interaction Networks

Shaikh Arifuzzaman and Bikesh Pandey
Department of Computer Science
University of New Orleans, New Orleans, LA 70148 USA
Email: {smarifuz, bpandey}@uno.edu

***Abstract*—Protein-protein interaction (PPI) networks are the networks of protein complexes formed by biochemical events and electrostatic forces. PPI networks can be used to study diseases and discover drugs. The causes of diseases are evident on a protein interaction level. For instance, an elevation of interaction edge weights of oncogenes is manifested in cancers. Further, the majority of approved drugs target a particular PPI, and thus studying PPI networks is vital to drug discovery.**

The availability of large datasets and need for efficient analysis necessitate the design of scalable methods leveraging modern high-performance computing (HPC) platforms. In this paper, we design a lightweight framework on a distributed-memory parallel system, which includes scalable algorithmic and analytic techniques to study PPI networks and visualize them. Our study of PPIs is based on network-centric mining and analysis approaches. Since PPI networks are signed (labeled) and weighted, many existing network mining methods working on simple unweighted networks are unsuitable to study PPIs. Further, the large volume and variety of such data limit the use of sequential tools or methods. Many existing tools also do not support a convenient workflow starting from automated data preprocessing to visualizing results and reports for efficient extraction of intelligence from large-scale PPI networks. Our framework supports automated analytics based on a large range of extensible methods for extracting signed motifs, computing centrality, and finding functional units. We design MPI (Message Passing Interface) based parallel methods and workflow, which scale to large networks. The framework is also extensible and sufficiently generic.

***Keywords*-network mining, biological networks, protein-protein interaction, network visualization, massive networks, HPC systems.**

I. Introduction

Network (graph) is a powerful abstraction of interactions among entities in a system [1], [2]. The entities and their interactions are represented as nodes (vertices) and links (edges) of a network, respectively. Examples include biological networks [2], [3], the web graph [4], various social networks [5], and many other information networks. Mining biological data is of growing interest since they represent fundamental biochemical mechanisms in a cell or in a living organism [3]. Due to the advancement of data and computing technology, biology and related disciplines generate a large volume and variety of data [2], many of them are about proteins and protein-protein interactions [6]. PPI networks offer an excellent opportunity to study disease dynamics in molecular level and can be insightful for drug discovery [7]–[9]. However, large volume and variety of PPI datasets pose computational challenges, which motivates for scalability, both in algorithmic methods and analysis workflow. In this paper, we develop an HPC-based framework to apply network-centric approaches to study PPI networks.

Significance of PPI Networks. Proteins are linear chain biomolecules that are the basis of functional networks in all organisms. Aspects of their interactions are of growing interest [10], [11]. Protein-protein interaction (PPI) networks can be used to study disease and for drug discovery [12], [13]. It also reveals the causes of diseases– for instance, most cancers are caused by increasing interaction edge weights of oncogenes and decreasing interaction edge weights of tumor suppressor genes [12], [14]. Most human diseases are thought to have fewer than five causal protein-protein interactions; many have two or fewer causal interactions [15]. Further, PPI networks helps in drug discovery. Many approved drugs target a particular protein-protein interaction [12], [15].

Related Work. There have been a line of work focusing on purely the bio-chemical aspect of PPIs [12], [15]. Unlike those work, this paper stresses on computing (mining and analysis) aspect of knowledge discovery and demonstrates how we can relate our results to biochemical contexts. There have been earlier work suggestive of the effectiveness of network-based approaches for analyzing PPIs [12], [15]. Local and global PPI network structural motifs suggest therapeutic strategies. The centrality hub nodes of PPI networks can be good candidates for drug target. Works such as [12] use both of the global PPI information and pathway knowledge to reveal more biochemical insights. Most work related to general network analysis do not consider signed and weighted networks [16], [17]. However, PPI networks are both signed and weighted. Moreover, many existing methods are not scalable to large networks [18]–[20]. Scalable parallel and sampling based algorithms [21]–[24] are required to deal with large network data.

Challenges with Big Data. In the era of big data, we are deluged with network data from a wide range of areas. The volume of biological and bio-medical data is also growing rapidly. The string repository [25] has PPI networks with $9.6M$ proteins and $1380M$ interactions. There are many other public repositories [26] that share large biological datasets. The emergence of such large-scale network data motivates us to find scalable algorithms and tools for extracting useful

978-1-5386-1956-8/17 $31.00 © 2017 IEEE
DOI 10.1109/DASC-PICom-DataCom-CyberSciTec.2017.178

intelligence. In some cases, these networks do not fit into the main memory of a single computing node. Further, an algorithm having a high computational complexity might fail to work on networks with a few millions edges.

Contributions. In this paper, we describe our framework for highly scalable and rigorous methods for mining and analyzing PPI networks. To address the issues emerged from large-scale datasets, we develop a workflow consisting of scalable labeled graph analysis algorithms leveraging large distributed multi-core clusters. The key contributions are as follows.

(1) **An HPC-based scalable tool.** The tool includes scalable parallel methods (algorithms) for discovering functional units and extracting motifs in PPI networks. Our methods and workflow scale to large networks for a wide variety of network metrics.
(2) **An extensible (and generic) framework.** The tool currently includes diverse mining and analysis methods including counitng triangular motifs, community detection, computing diameter, and several centrality measures. Any new methods can easily be integrated with the tool.
(3) **Identification of relevance to biological or bio-medical contexts of PPI.** Our methods for signed motif extraction, centrality computation, and discovery of functional units can be used to identify target proteins and important hubs. Such network motifs and properties of a PPI network have useful implications for drug target discovery.
(4) **Promotion of interdisciplinary collaboration.** We anticipate this tool can facilitate multidisciplinary investigations consisting of experts from both computational and biological domains. Further, the tool can essentially be generalized to other related applications in neuroscience, medical informatics, and likes.

The rest of the paper is organized as follows. The datasets and computing resources are briefly described in Section II. We present the overview and architecture, capabilities, and evaluation of our framework in Section III, IV, and V, respectively. We conclude in Section VI.

II. Datasets and Resources

We present our datasets, computational model, and resources (experimental setup) below.

Datasets. We study PPI networks from StringDB database [25] for several organisms. The networks are represented as edge-lists with several interaction values based on various evidences such as interaction and coexpression scores. The datasets we used are summarized in Table I. These datasets contain edge weights valued on a scale of 0-1000 between two proteins. Any such weight corresponds the overall interaction score, which is the sum of all the categorical scores such as coexpression, neighborhood, and experimental scores. The datasets identify proteins using unique protein identifiers called Ensembl Protein IDs determined by Ensembl.org. Further details on these proteins and other genes can be found at Ensembl Genome Browser [27].

We also experimented on other datasets found from National Center for Biotechnology Information [28] and BioGrid [26].

TABLE I: A subset of datasets used in our experiments.

Network	Nodes	Edges	Source
Homo Sapiens	19247	4274001	StringDB [25]
Acetobacterium Woodii	3439	369956	StringDB [25]
Albugo Laibachii	5849	1443060	StringDB [25]
Dinoroseobacter Shibae	3567	412618	StringDB [25]
Bacillus Cytotoxicus	3765	298873	StringDB [25]

Many of these datasets have no quantifiable interaction scores that can be further analyzed. Even though we experimented on datasets from several sources, many of them are not presented in this paper for brevity.

Computation Model and Resources. The parallel algorithms our tool uses were developed for MPI based distributed-memory parallel systems where each processor has its own local memory. The processors do not have any shared memory, and they communicate via exchanging messages. Compute resources are the physical resources on which individual jobs are executed. Our current resource include two HPC Linux clusters at LONI (Louisiana Optical Network Infrastructure) [29] and the University of New Orleans (UNO). LONI QueenBee system is a 50.7 TFlops Peak Performance 680 compute node cluster running the Red Hat Enterprise Linux 4 operating system. Each node contains two Quad Core Xeon 64-bit processors operating at a core frequency of 2.33 GHz. The compute cluster at UNO is a small cluster with 2 large-memory computing nodes, each with 16 cores and 512GB of RAM, connected by QDR infiniband interconnect and running Linux operating system.

III. New Generation Graph Analytical Tool for PPI Networks

The use of network (graph) analysis for understanding protein interactions and their implication on broader aspects of biological process in organisms is still nascent [12], [15], and more studies are needed to demonstrate a clearer picture of results. In this paper, we hope to contribute to this literature by developing an HPC-based tool that helps assess both node- and clustering-based characterization of PPI networks.

The proposed framework builds upon and extends significantly the existing work on scalable algorithms for graph data preprocessing [21], counting triangular motifs [23], and efficient parallel load balancing schemes [24]. It complements the protein interaction literature with scalable algorithmic methods for efficient analysis. It is well established that causation of disease and drug discovery have significant correlation with network properties of nodes in PPI networks [11], [12], [15]. Based on the prior work of the authors on network-centric algorithms, for both sequential and parallel settings, and by leveraging open-source network analysis libraries such as SNAP [30] and NetworkX [20], we build an extensible computational framework for mining and analyzing PPI networks.

A. Architectural Overview of the Tool

Our framework for analyzing PPI networks is built on a distributed system consisting of a set of well-defined units (and services). The framework incorporates a Linux-based

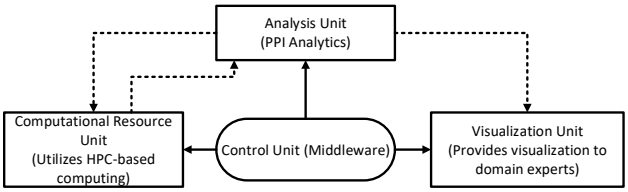


Fig. 1: Architectural overview of our framework for scalable mining, analysis, and visualization of PPI networks.

architecture with middleware developed with shell-script and C++ based codes and scripts. Our network analysis kernels are mostly developed in C++ with MPI libraries. We also have python-based application codes and scripts. For job submission, we use moab qsub scripts. All functional units are coupled loosely so as to support extensibility and modifications. Fig. 1 depicts the high-level architecture of the framework. We discuss the key components below.

Control Unit. The control unit employs the central communication and coordination mechanism for our tool. It provides asynchronous, loose coupling of the system components. The control unit initiates a workflow– put requests for executing jobs. Every analysis task is transformed into a job consisting of an analysis kernel. Additionally, the control unit facilitates task parallelism by distributing different serial tasks among separate MPI processes. Requests are handled and scheduled by PBS qsub scripts using moab scheduling mechanism. The control unit specifies the details about how a set of analyses is to be fulfilled, in the form of an embedded workflow. An analysis request contains the parameters to run the analysis. The request also contains the specification for the workflow to run, including both pre- and post-processing and inspecting the output. Based on this inspection, a new workflow can be initiated with a new set of parameters and analysis kernels.

Computational Resource Unit. Once execution requests are identified, they are run on a specific physical machine. It is done by constructing system-specific job submission scripts and monitoring the progress of the execution. To achieve larger scalability, we need to speed up the analysis significantly and make use of the computing clusters efficiently. We design MPI (Message Passing Interface) based parallel computing techniques to scale our methods to large networks and to a large number of processors. Our motif counting methods are based on efficient MPI-based algorithms [23]. To execute a bunch of sequential analysis kernel, we design task parallelism: we distribute multiple kernels among a set of MPI processes. Since our tool is extensible, new methods (either serial or parallel) can easily be integrated. Our scripts automatically assign them to appropriate number of processors guided by the metadata of the executable method.

Analysis Unit. Analysis unit is the computational engine behind mining PPI networks. This unit consists of scalable network analysis kernels, both the ones developed from scratch for this tool and from open-source graph analysis algorithms. Since the description of this unit is rather involved, we present it in the next section separately. In conjunction to analysis unit, we have a *Data Management Sub-unit*: this unit is responsible for managing the data resources that reside on a system. The unit also deals with cleaning datasets, applying scores/thresholds, converting formats, storing or formating results, etc. There are several high performance services developed for data management. For instances, we implement *parallel read*, where processors can read disjoint portions of a file in parallel.

Data Report or Visualization Unit. Our report and visualization unit is based on gnuplot tool (http://www.gnuplot.info). We generate numerous statistics plots and distribution using gnuplot. Such capability is integrated with analysis unit, so generation of these tools are automated. Adding a new plot and visualization capability is straightforward and requires little C++ coding. A new visualization is modularized (and thus flexible and easy to maintain) by the virtue of being a C++ object.

We also use a java-based visualization library *Gephi* [31] for generating additional visualizations. Gephi is open source, modular, and easily extensible through plugins. It is also rich in visualization features. To create a visualization of a network, the network is converted into gexf format, an XML representation. The format allows for dynamically adding multiple attributes to nodes and edges. Any layout algorithms can be used to determine object locations. Statistics such as betweenness, pagerank, and degree can be applied to decide the size and color of the nodes and edges. Visualization by Gephi can give useful insights into a network by highlighting important nodes, edges and communities in a graph or a subgraph. The primary features and benefits of such visualization are as follows.

- Convenient layouts: Gephi provides several layout algorithms from the literature such as Force Atlas, Yifan Hu and Fruchterman Reingold [31].
- Feature-based organization: The node sizes can be proportional to their degrees, betweenness centrality, or other network metric.
- Subgraph visualization: It offers visualization of subgraphs, which is very useful, especially for massive networks. We have developed several heuristics for choosing subgraphs. First, find a seed (by random seed, central nodes, etc.); second, expand the seed by a BFS traversal.

Using Gephi orthogonal to gnuplot gives the user additional capabilities for visual analysis. The inputs and parameters needed for Gephi are automatically computed by our tool. The user can interact with the tool to configure different visualizations. Note that our framework allows for adding any open source visualization tool with little coding effort.

IV. Network Analysis Kernels

A suite of graph metrics (or analysis kernels) is used as the computational engine behind our framework. These kernels are of varying levels of complexity and computational intensity. We classify them into three categories based on the topological granularity they focus on– global, community, and local, as shown in Fig. 2. Note that our framework is readily extensible

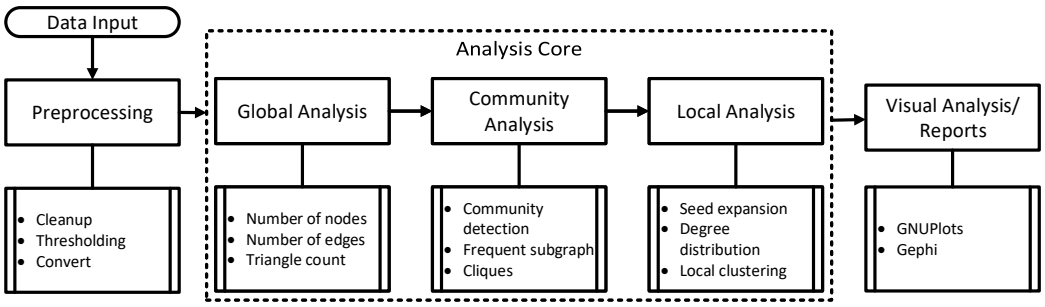


Fig. 2: Schematic diagram of our analysis workflow starting from data preprocessing to the generation of reports and visualization. The workflow supports a multi-level approach with a veriety of analysis kernels working on different topological granularity, starting from global to local analysis.

to include any graph kernels. Further, how many of these kernels will be used for a particular investigation depends on the requirements of the analysts.

We use the global metrics to measure the high-level properties of the PPI networks. These metrics are mostly less expensive and are intended to work on the entire graph. For more expensive and complex measures, we use parallel implementation of them. As instance, our tool adapts the parallel algorithms presented in [22]–[24] to find signed triangular motifs at scale. These algorithms are based on efficient partitioning and load balancing schemes and scale to large networks.

We use another suite of metrics to investigate PPI networks at community level. Complex systems are organized in clusters or communities, each having a distinct role or function. In the corresponding network representation, each community appears as a dense set of nodes having higher connection inside the set than outside. Communities reveal the organization of complex systems and their function. For PPI networks, a community is often interpreted as a functional unit, and thus, community detection is also another important analysis kernel for PPI networks. We use several scalable algorithms for community detection such as Louvain [32] and label propagation [33]. We also use several related analysis kernels such as k-core decompositions. Such decompositions can leverage the higher-order structures to locate the dense subgraphs with hierarchical relations.

Computation on individual nodes are done by using local metrics. Local metrics are usually the slowest among the kernels. We implemented several distributed-memory algorithms such as computing local clustering coefficients and local jaccard indices. We are also in the process of adding more parallel kernels. Serial analysis kernels can also be used using task parallel execution as discussed in Section III. Further, it is also an attractive option to first identify important subgraphs by community analysis and then apply the local metrics on the subgraphs (which is smaller than the original graph). Centrality metrics such as local between centrality and closeness centrality are also important local metrics for identifying central nodes of bio-chemical significance.

A Multi-Level Approach. Our workflow suggests a multi-level approach for efficient analysis. It is generally advised to start analysis with the coarsest (global) and becoming finer at each iteration. Any structure identified as interesting at a coarse level are passed down to be analyzed at the next finer level. We generally identify three levels, based on the topological granularity levels, as mentioned above as global, community, and local levels. At the coarsest level, only the global metrics can be applied on the whole network. Communities and local metrics on individual nodes are not considered at this stage. We use efficient and scalable global metrics. Next, community-level metrics are computed. Individual communities can then be locally analyzed by applying local metrics. Note that such multi-level approach allows to work with even very scare resources (a commodity laptop) in a computationally efficient way. However, our parallel algorithms and scalable HPC-based framework allows to apply local metrics on the entire networks. Hence the analysts are not limited to follow the multi-level approach in a strict order; rather the approach serves as an organizational or workflow guide.

As for the analysis automation, a simple self-descriptory script serves as the starting point of the workflow. It is straightforward to specify the analysis kernels and input network to work on. After initializing the workflow, all the remaining steps such as data pre-processing, analysis, and generation of reports and plots are fully automated. The end-user can inspect the reports and plots and then re-run analyses with different parameters and kernels, if needed.

V. Experimental Results and Implications

We provide a flexible tool to support scalable data analytics for PPIs. The tool reveals useful patterns and properties from PPI networks by using appropriate mining and analysis techniques. We present a summary of computed network metrics, their biological relevance, scalability of the tool, and a comparison with previous tools below.

A. Computing Global Network Metrics

Our global analysis consists of metrics such as finding general statistics (e.g., number of edges, nodes), finding patterns and motifs, e.g., counting triangles, and finding diameter of the

TABLE II: Network properties of our datasets: degree, components, coreness, triangles, clustering coefficients (CC), and diameter statistics.

Networks	Degree			Components		Max. k-core	Triangles	Avg. CC	Diameter
	Min.	Max.	Avg.	# of Comp.	Max. Size				
Acetobacterium Woodii	1	2075	172.51	1	4192	146	6.26M	0.191	6
Albugo Laibachii	1	2676	493.44	21	5798	566	215.12M	0.476	6
Bacillus Cytotoxicus	1	1746	159.51	2	3803	146	6.41M	0.226	5
Dinoroseobacter Shibae	1	2371	229.04	1	3574	172	13.06M	0.297	5
Homo Sapiens	1	10853	444.12	1	19247	791	321.6M	0.231	6

networks. Table I shows the number of proteins and interactions for five PPI networks. Homo Sapiens dataset has a large number of proteins and their identified interactions. Albugo Laibachii dataset also has over a million protein interactions. We present several analyses on all five datasets of Table I below.

Finding Patterns or Motifs. Network motifs of size 3 and 4 are overrepresented in real-world networks generated through processes such as hyperlink creation, language formation, and personal social network propagation. Such structures in biological functional networks are suggestive of processes such as positive and negative feedback loops [6], [9], which have important implications for therapeutic strategies. We enumerate signed triangles for networks datasets of Table I. As shown in Table II, Homo Sapiens and Albugo Laibachii networks have $321.6M$ and $215.12M$ triangles, respectively, which indicates a high triangle density (triangles per node). In fact, Algugo Laibachii has the highest triangle density among the five datasets. Table II also shows average clustering coefficients (CC) of the five datasets. These values are large, indicating the proteins interact with the neighborhood quite closely.

Computing Diameters. We compute diameters to find insights about reachability and ease of communication and diffusion in PPI networks. The diameters are less than 6 for all networks (Fig. II), suggesting good reachability in the network. Any biochemical process originating in a particular protein can reach to the farthest protein in only six hops. Domain experts may find this information useful in designing drugs for target proteins. Our implementation of diameter kernel is adapted from SNAP [30] library.

B. Community and Subgraph based Analysis.

We execute community detection methods to reveal functional units in PPI networks. The community statistics are shown in Table III. For all PPI networks, a number of functional untis are detected: for example, for Homo Sapiens, five different functional units (group of proteins) are revealed by our community analysis. The modularity scores quantify the degree of cohesiveness (tightly coupledness) of protein in these communities. We can further inspect the community structures visually with Gephi, as shown in Fig. 3. Gephi supports interactive visualization– for example, the neighborhood of a particular node can be zoomed in and inspected for details. Further, we also decompose the graph into different connected components, when available, to find their properties. The component statistics reveal whether the network consists of a single or multiple connected component, as shown in Table II. For example, Homo Sapiens has a single connected component, whereas the Albugo Laibachii network consists of several components. Another important neighborhood and subgraph based metric is k-coreness. We also investigate k-cores of different networks. Table II reports the maximum coreness for each of the five PPI networks. Homo Sapiens has a maximum coreness of 791– it has a subgraph where each node has degree at least 791. This indicates a large cohesive group. Fig. 4 shows k-core distribution of several PPI networks. K-core decompositions can leverage the higher-order structures to locate the dense subgraphs with hierarchical relations.

TABLE III: Community statistics of 5 PPI networks.

Networks	Comm. Size		# of Comm.	Modularity
	Max.	Avg.		
Acetobacterium Woodii	1721	419	10	0.170
Albugo Laibachii	2281	739	9	0.159
Bacillus Cytotoxicus	1441	317	12	0.135
Dinoroseobacter Shibae	1173	595	6	0.129
Homo Sapiens	7296	4013	5	0.207

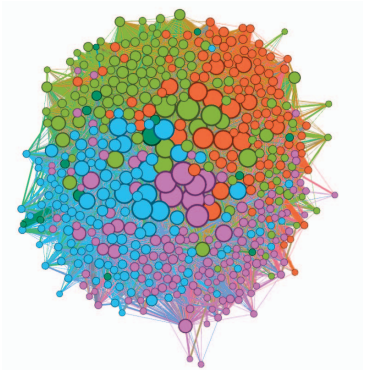
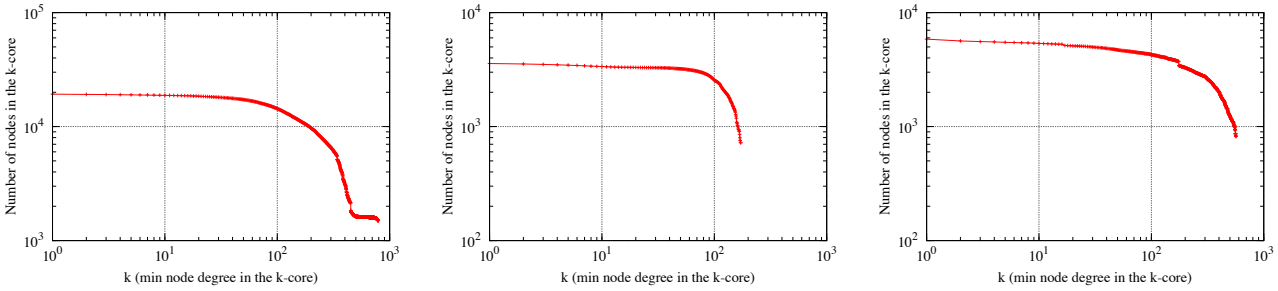


Fig. 3: Community structure in a subgraph of Homo Sapiens PPI network. Node colors are based on community membership and node sizes on degrees. The plot is generated by Gephi and can further be interactively investigated.

C. Analysis of Local Metrics

We computed several local metrics such as clustering coefficient (CC) on nodes, degree distribution, expanding the neighborhood of a node (seed expansion), to find properties on individual nodes. Fig. 5 shows that all networks have a few high degree nodes whereas most of the nodes have small degrees. Fig. 6 shows the CC distribution of three PPI networks. Most of the nodes (proteins) have clustering



(a) Homo Sapiens (b) Dinoroseobacter Shibae (c) Albugo Laibachii

Fig. 4: Kcore distribution of three PPI networks. Coreness is suggestive of the existence of cohesive group and neighborhood. All the above networks have large coreness consisting of a large portion of nodes.

coefficients centered around the global average, even though a small percentage of nodes have large clustering coefficients. Running local metrics can reveal further insights about an individual node and its neighborhood.

D. Detecting Central Nodes

The presence of central "hub" regulators is a prominent feature in biological networks [9]. Such nodes make especially attractive drug targets, because they are often central to multiple biochemical pathways involved in processes like cell proliferation [15]. The case is similar to social networks, where nodes with high centrality can be called *central individuals*, and are important to graph propagation processes, such as gossip [35]. Along the same spirit, we compute various centrality metrics for PPI networks to find influential regions. We present below our experiment on Homo Sapiens dataset for *betweenness*, *closeness*, and *degree* centrality.

Cross-checking central nodes for Homo Sapiens. We found that the following three proteins have the highest centrality scores for Homo Sapiens: ENSP00000344818 (UBC protein), ENSP00000351686 (PRDM10 protein), and ENSP00000328973 (TSPO protein) (shown in Table IV).

TABLE IV: Top three proteins based on centrality metrics.

Proteins	Betweenness	Closeness	Degree
ENSP00000344818	0.0798	0.6949	0.5639
ENSP00000351686	0.0094	0.6014	0.3425
ENSP00000328973	0.0082	0.5907	0.3129

The existing literature of PPI supports the importance of the above three proteins. Ubiquitin C (UBC) protein, as its name suggests, is a protein available ubiquitously around the eukaryotic tissues. This explains the higher value of betweenness centrality for this protein. UBC protein is encoded by the UBC gene which regulates cellular ubiquitin levels under stress [36]. UBC protein contributes to liver development and hence, lack of UBC genes in unborn fetuses leads to embryonic lethality [37]. PRDM10 is a protein that has been linked to the transcriptional regulation [38]. Some studies on mice have indicated that this may also help in the development of the Central Nervous System [39]. TSPO protein, encoded by the TSPO gene, is found in the outer mitochondrial membrane. Generally, TSPO has been linked with cholesterol transport with mixed evidence [40] and has also been associated with immune response [41] and heart regulation [42] depending on the kind of tissue it is working in.

E. Scalability Analysis

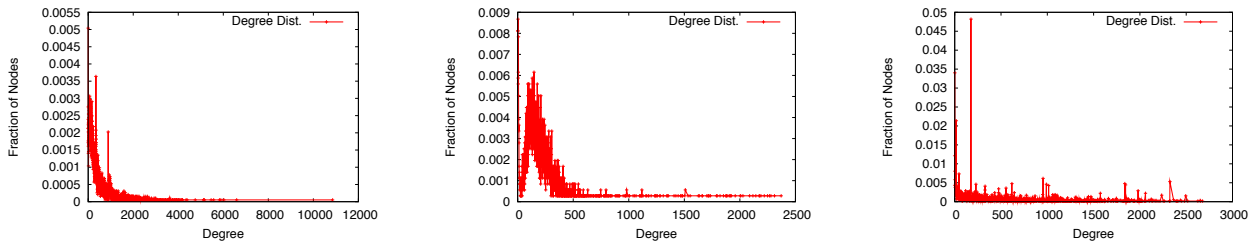
We use scalable algorithmic methods for compuitng various network metrics. For example, we adapt the methods in [23] to count triangular motifs. The speedup factors for this methods on three PPI networks is give in Fig. 7. The method shows good speedups and scales almost linearly to a large number of processors. In addition to parallel algorithms, we use efficient sequential methods in a task parallel fashion. We allocate multiple MPI processors and distribute computing kernels among those processors. In effect, this results in a parallel workflow with sequential kernels. Such task parallel design significantly speedup the analysis. As shown in Table V, our HPC-based workflow achieves almost ten-fold speedup over a serial workflow with ten sequential kernels. Note that this speedup is in excess to what we already achieve with parallel methods such as triangle counting.

TABLE V: Workflow scalability: runtime performance for ten analysis metrics with sequential workflow and our HPC-based parallel workflow.

Networks	**Runtime (sec.)**		**Speedup**
	Seq. workflow	Our workflow	
Acetobacterium Woodii	576	62	9.29
Albugo Laibachii	820	95	8.63
Bacillus Cytotoxicus	540	58	9.31
Dinoroseobacter Shibae	680	72	9.44
Homo Sapiens	1280	130	9.85

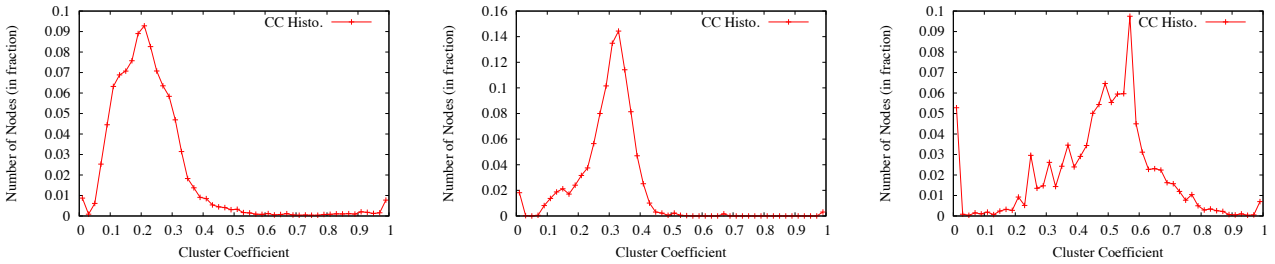
F. Comparison with Other Network Analysis Tools

There exist several network analysis tools such as NetworkX [20], Pajek [19], SNAP [30], PEGASUS [43], and CINET [44], [45]. NetworkX is an open source python-based software package for studying complex networks. NetworkX contains a large collection of network algorithms. Pajek is a tool for the analysis and visualization of networks having thousands to millions of vertices. Stanford Network Analysis Project (SNAP) is a general purpose network analysis library. Another



(a) Homo Sapiens (b) Dinoroseobacter Shibae (c) Albugo Laibachii

Fig. 5: Degree distribution of three PPI networks. There are a few nodes with large degress.



(a) Homo Sapiens (b) Dinoroseobacter Shibae (c) Albugo Laibachii

Fig. 6: Clustering coefficient (CC) histogram of three PPI networks. Most nodes have the clustering coefficients around the global average.

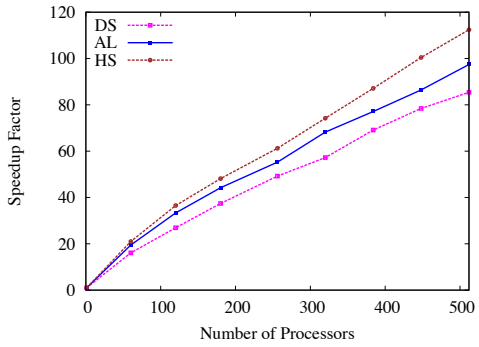


Fig. 7: Speedup factors of triangle counting algorithm with three PPI networks– Homo Sapiens (HS), Dinoroseobacter Shibae (DS), and Albugo Laibachii (AL).

toolkit Network Workbench provides an online portal for network researchers. PEGASUS is a peta-scale distributed graph mining system that provides large-scale algorithms for several graph mining tasks and runs on clouds. CINET is another versatile web-based tool for analyzing unlabeled (unsigned) networks.

All the above tool vary in generality, interface, types of networks they support, and the availability of HPC-based resources and frameworks. Many of the above tools, e.g., NetworkX, do not include scalable parallel algorithms or support scalable computing on HPC resources. Some of them, e.g., CINET, lack support for signed networks. Only a few (e.g., CINET) supports workflow coordination. To the best of our knowledge, the novelty of our framework comes collectively from its lightweight (i.e., no need for complex setup or installation of extraneous/expensive support tools), capability to work on signed and weighted networks, offering multi-approach with varying topological granularity, its simple yet efficient workflow coordination, and the availability and incorporation of data and task parallelism through the careful design of distributed-memory algorithms and other HPC techniques. The framework is also extensible and sufficiently generic for many related applications.

We also want to comment that our tool is not a competitor of other existing graph analysis tools. Our tool complements the capabilities of existing tools in several aspects, is extensible, and can integrate many open-source scalable algorithms.

VI. Conclusion

Interests for PPI networks are growing in biological and medical sciences applications for studying diseases and discovering drugs. The emergence of large volume of PPI datasets challenges efficient and scalable mining of such networks. In this paper, we presented an analytical framework for PPI networks, which addresses the challenges of big data through a flexible tool based on parallel algorithms and other HPC techniques. We demonstrated the scalability and application

of the tool on several PPI networks consisting of millions of edges from a variety of sources. Our tool is effective in identifying central nodes and other interesting patterns. We also introduced different level of analysis granularity to efficiently work with available resources. The tool is also lightweight, flexible, and extensible. We believe that this tool will be useful in tackling emerging large volume and variety of PPI networks (and other related biological networks) and gaining useful insights from them.

ACKNOWLEDGMENTS

This work has been partially supported by Louisiana Board of Regents RCS Grant LEQSF(2017-20)-RD-A-25 and College of Sciences Internal Grant (University of New Orleans, Spring 2017).

REFERENCES

[1] M. Newman, "The structure and function of complex networks," *SIAM Review*, vol. 45, pp. 167–256, 2003.

[2] M. Girvan and M. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.

[3] J. Chen and S. Lonardi, *Biological Data Mining*. Chapman & Hall/CRC, 2009.

[4] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener, "Graph structure in the Web," *Computer Networks*, vol. 33, no. 1–6, pp. 309–320, 2000.

[5] H. Kwak *et al.*, "What is twitter, a social network or a news media?" in *WWW*, 2010.

[6] R. M. Ewing, P. Chu, F. Elisma, H. Li, P. Taylor, S. Climie, L. McBroom-Cerajewski, M. D. Robinson, L. O'Connor, M. Li *et al.*, "Large-scale mapping of human protein–protein interactions by mass spectrometry," *Molecular systems biology*, vol. 3, no. 1, p. 89, 2007.

[7] J. S. Bader, A. Chaudhuri, J. M. Rothberg, and J. Chant, "Gaining confidence in high-throughput protein interaction networks," *Nature biotechnology*, vol. 22, no. 1, pp. 78–85, 2004.

[8] J.-D. J. Han, N. Bertin, T. Hao, D. S. Goldberg, G. F. Berriz, L. V. Zhang, D. Dupuy, A. J. Walhout, M. E. Cusick, F. P. Roth *et al.*, "Evidence for dynamically organized modularity in the yeast protein–protein interaction network," *Nature*, vol. 430, no. 6995, pp. 88–93, 2004.

[9] B. Schwikowski, P. Uetz, and S. Fields, "A network of protein–protein interactions in yeast," *Nature biotechnology*, vol. 18, no. 12, pp. 1257–1261, 2000.

[10] J.-F. Rual, K. Venkatesan, T. Hao, T. Hirozane-Kishikawa, A. Dricot, N. Li, G. F. Berriz, F. D. Gibbons, M. Dreze, N. Ayivi-Guedehoussou *et al.*, "Towards a proteome-scale map of the human protein–protein interaction network," *Nature*, vol. 437, no. 7062, pp. 1173–1178, 2005.

[11] U. Stelzl, U. Worm, M. Lalowski, C. Haenig, F. H. Brembeck, H. Goehler, M. Stroedicke, M. Zenkner, A. Schoenherr, S. Koeppen *et al.*, "A human protein-protein interaction network: a resource for annotating the proteome," *Cell*, vol. 122, no. 6, pp. 957–968, 2005.

[12] D. C. Altieri, "Survivin, cancer networks and pathway-directed drug discovery," *Nature Reviews Cancer*, vol. 8, no. 1, pp. 61–70, 2008.

[13] P. K. Brastianos, S. L. Carter, S. Santagata, D. P. Cahill, A. Taylor-Weiner, R. T. Jones, E. M. Van Allen, M. S. Lawrence, P. M. Horowitz, K. Cibulskis *et al.*, "Genomic characterization of brain metastases reveals branched evolution and potential therapeutic targets," *Cancer discovery*, 2015.

[14] K. Chin, C. O. De Solorzano, D. Knowles, A. Jones, W. Chou, E. G. Rodriguez, W.-L. Kuo, B.-M. Ljung, K. Chew, K. Myambo *et al.*, "In situ analyses of genome instability in breast cancer," *Nature genetics*, vol. 36, no. 9, pp. 984–988, 2004.

[15] A. L. Hopkins, "Network pharmacology: the next paradigm in drug discovery," *Nature chemical biology*, vol. 4, no. 11, pp. 682–690, 2008.

[16] S. Suri and S. Vassilvitskii, "Counting triangles and the curse of the last reducer," in *20th international conference on World Wide Web*, 2011.

[17] N. Chiba and T. Nishizeki, "Arboricity and subgraph listing algorithms," *SIAM Journal on Computing*, vol. 14, no. 1, pp. 210–223, 1985.

[18] S. Fortunato and A. Lancichinetti, "Community detection algorithms: a comparative analysis," in *4th International ICST Conference on Performance Evaluation Methodologies and Tools*, 2009.

[19] Pajek network analysis tool. http://vlado.fmf.uni-lj.si/pub/networks/pajek/.

[20] Networkx tool. https://networkx.github.io/.

[21] S. Arifuzzaman and M. Khan, "Fast parallel conversion of edge list to adjacency list for large-scale graphs," in *23rd High Performance Computing Symposium*, 2015.

[22] S. Arifuzzaman, M. Khan, and M. Marathe, "A Space-efficient Parallel Algorithm for Counting Exact Triangles in Massive Networks," in *17th IEEE International Conference on High Performance Computing and Communications*, 2015.

[23] S. Arifuzzaman, M. Khan, and M. Marathe, "PATRIC: A parallel algorithm for counting triangles in massive networks," in *22nd ACM International Conference on Information and Knowledge Management*, 2013.

[24] S. Arifuzzaman, M. Khan, and M. Marathe, "A fast parallel algorithm for counting triangles in graphs using dynamic load balancing," in *2015 IEEE BigData Conference*, 2015.

[25] String: functional protein association networks. https://string-db.org/.

[26] Biogrid: Database of protein, chemical, and genetic interactions. https://thebiogrid.org/.

[27] Ensembl genome browser. http://www.ensembl.org.

[28] National center for biotechnology information. https://www.ncbi.nlm.nih.gov/genome/viruses/retroviruses/hiv-1/interactions/browse/.

[29] Louisiana optical network infrastructure. https://loni.org/.

[30] Snap. http://snap.stanford.edu/.

[31] Gephi - the open graph viz platform. https://gephi.org/.

[32] V. Blondel, J. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 10, p. 10008, 2008.

[33] U. Raghavan, R. Albert, and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," *CoRR*, vol. abs/0709.2938, 2007.

[34] K. Henderson, T. Eliassi-Rad, C. Faloutsos, L. Akoglu, L. Li, K. Maruhashi, B. A. Prakash, and H. Tong, "Metric forensics: A multi-level approach for mining volatile graphs," in *Proc. of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2010.

[35] A. Banerjee, A. Chandrasekhar, E. Duflo, and M. O. Jackson, "Gossip: Identifying central individuals in a social network," *CoRR*, vol. abs/1406.2293, 2014.

[36] O. Wiborg, M. Pedersen, A. Wind, L. Berglund, K. Marcker, and J. Vuust, "The human ubiquitin multigene family: some genes contain multiple directly repeated ubiquitin coding sequences." *The EMBO journal*, vol. 4, no. 3, p. 755, 1985.

[37] K. Ryu *et al.*, "The mouse polyubiquitin gene ubc is essential for fetal liver development, cell-cycle progression and stress tolerance," *The EMBO journal*, vol. 26, no. 11, pp. 2693–2706, 2007.

[38] "String prdm10," https://string-db.org/cgi/network.pl?taskId=eU6OEL2pwmaP.

[39] Ncbi prdm10. https://www.ncbi.nlm.nih.gov/gene/56980.

[40] J.-J. Lacapere and V. Papadopoulos, "Peripheral-type benzodiazepine receptor: structure and function of a cholesterol-binding protein in steroid and bile acid biosynthesis," *Steroids*, vol. 68, no. 7, pp. 569–585, 2003.

[41] M. Pawlikowski, "Immunomodulating effects of peripherally acting benzodiazepines," *Peripheral benzodiazepine receptors*, pp. 125–135, 1993.

[42] X. Qi, J. Xu, F. Wang, and J. Xiao, "Translocator protein (18 kda): a promising therapeutic target and diagnostic tool for cardiovascular diseases," *Oxidative medicine and cellular longevity*, vol. 2012, 2012.

[43] U. Kang, C. E. Tsourakakis, and C. Faloutsos, "Pegasus: A peta-scale graph mining system implementation and observations," in *Proc. of the 9th IEEE International Conference on Data Mining*, 2009.

[44] Cinet system. http://cinet.vbi.vt.edu/granite/granite.html.

[45] S. E. Abdelhamid, R. Aló, S. M. Arifuzzaman *et al.*, "CINET: A cyberinfrastructure for network science," in *Proceedings of the 8th IEEE International Conference on e-Science (e-Science 2012), Chicago, IL, USA*, October 2012, pp. 1–8.