Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. VI (2011), No. 1 (March), pp. 33-44

The Flag-based Algorithm - A Novel Greedy Method that Optimizes Protein Communities Detection

R. Bocu, S. Tabirca

Razvan Bocu
National University of Ireland, Cork
Department of Computer Science
E-mail: razvan.bocu@cs.ucc.ie

Sabin Tabirca
National University of Ireland, Cork
Department of Computer Science
E-mail: tabirca@cs.ucc.ie

Abstract: Proteins and the networks they determine, called interactome networks, have received attention at an important degree during the last years, because they have been discovered to have an influence on some complex biological phenomena, such as problematic disorders like cancer. This paper presents a contribution that aims to optimize the detection of protein communities through a greedy algorithm that is implemented in the C programming language. The optimization involves a double improvement in relation to protein communities detection, which is accomplished both at the algorithmic and programming level. The resulting implementation's performance was carefully tested on real biological data and the results acknowledge the relevant speedup that the optimization determines. Moreover, the results are in line with the previous findings that our current research produced, as it reveals and confirms the existence of some important properties of those proteins that participate in the carcinogenesis process. Apart from being particularly useful for research purposes, the novel community detection algorithm also dramatically speeds up the proteomic databases analysis process, as compared to some other sequential community detection approaches, and also to the sequential algorithm of Newman and Girvan.
Keywords: Interactome networks, protein-protein interactions, protein communities, cancer, greedy algorithm.

1 Introduction

1.1 Basic Considerations on Protein Networks and Their Importance

Interactome networks, or, more specifically, networks of proteins, determine a fundamental biological theoretical entity. Theoretical and practical endeavours often use interactome networks-related formalisms in order to analyze the protein interactions that determine a biological network, which is essential for the proper organization and function of a biological organism. These networks exhibit a complex structure, which implies that any research activity in the field is handled with inherent theoretical and technical difficulties. Nevertheless, the dynamics and the structure of these biological networks have to be accurately understood, as they play an important role on the function of a biological organism seen as a whole, regardless their degree of structural complexity. As a consequence, it is highly required to design and implement efficient proteomic data analysis techniques that can be integrated in any research framework that study the structure and properties of the interactome networks.

The aim of this paper is to present a novel and faster algorithm that performs the detection of communities in the interactome networks, based on a computationally-effective greedy technique.

The significant influence that proteins exercise on fundamental physiological processes has been demonstrated in a series of recent contributions. In this respect, this paper re-states our research's previous developments, apart from the algorithmic optimization itself, that cancer affects the most important proteins in the interactome network and, as a consequence, the normal function of the organism is greatly disturbed. An accurate understanding of the structure and importance of proteins requires the usage of efficient analysis techniques. In this context, the proper detection of protein communities is of utmost importance, because it is one of the fewer methods that allows an in-depth and informative analysis of protein networks, through the isolation of functionally-related protein communities.

The paper will briefly enumerate the most relevant existing works regarding the community detection. Furthermore, the novel algorithm will be described and analyzed. Also, its practical usability is assessed on real proteomic data.

1.2 Essential Previous Work

The Newman-Girvan algorithm is one of the methods used to detect communities in complex systems. A community is built up by a subset of nodes within which the node-node connections are dense, and the edges to nodes in other communities are less dense. It is important to note that there are a number of alternative algorithmic techniques for the detection of communities in networks. They include hierarchical clustering, partitioning graphs to maximize quality functions such as network modularity, k-clique percolation, and some other interesting algorithmic methods [1,2]. Nevertheless, the Newman and Girvan conceptual system is often chosen as a pretext for scientific contributions due to its structural articulation and its ability to be used in a wide range of practical situations [3]. The Newman-Girvan algorithm is particularly used to compute betweenness for edges (biological links) that connect the nodes (proteins) in a network.

The algorithm of Blondel et al. [26] is able to find high modularity partitions of large networks in a timely manner and, thus, to unfold a complete hierarchical community structure for the analyzed network. Contrary to some other community detection algorithms, the network size limits that this algorithm faces are mainly generated by the limited storage capacity on the processing machines, rather than by the computational complexity. The algorithm of Blondel et al. represents a different greedy approach to community detection that can be successfully applied to protein networks. The algorithm assigns proteins to their respective communities following a particular-to-general approach. Thus, proteins are progressively added to their correct clusters considering only the existence of inter-protein links. This way, at some point the modularity cannot increase any more and thus, the algorithm stops. This approach has the significant advantage of avoiding the usage of computationally expensive data structures, as it is the case with the algorithm of Clauset et al. Therefore, it is rather limited by storage (memory) requirements than by computational time.

The algorithm of Clauset et al. [25] has the merit to point out that the updates that are performed by the algorithm of Newman and Girvan involve a significant number of pointless operations, as a consequence of the sparse nature of the adjacency matrix. As a consequence, the algorithm is optimized in order to properly handle the sparsity of the input data sets. Thus, the three data structures that the algorithm uses contribute to optimizing the analysis process of protein networks. In this respect, it can be stated that the modularity can be updated in a sensibly quicker way using these three data structures, as compared to the algorithm of Newman and Girvan. Consequently, it can be inferred it is one of the few algorithms that can be used to determine community structure in the case of protein networks.

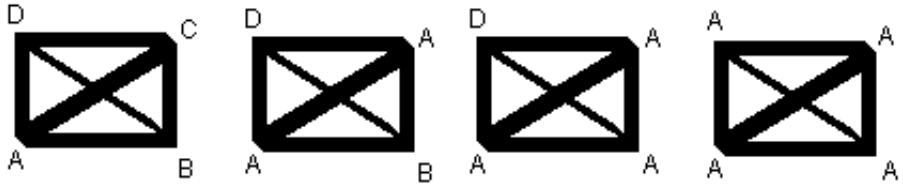

Figure 1: The flag values are updated one by one, starting with the leftmost representation. As a consequence of the complete nature of this network, all the nodes are flagged identically, thus constituting the only and network-wide community.

The following sections will describe the improved algorithm in more detail, along with some theoretical considerations that are mandatory for a good understanding of the new approach's structure and strengths.

2 General Presentation of the New Approach

2.1 Description of the Algorithm

The algorithm is designed according to a local-to-general approach. The iterative process starts with all the nodes considered as members of their own one-node community. Then, each node assigns itself to one of the neighbouring communities considering local information about its proximity. In order to accomplish this, we attached to each node in the network an additional data structure called *flag*, which holds information about the community to which a node currently belongs.

The mechanism that coordinates the flag-based community algorithm is the following. Let us suppose that a certain node *node* has the set of neighbours $\{node_1, node_2, node_3, ..., node_k\}$, with $k \leq n$, where n represents the number of nodes in the network. Each of these neighbours has a flag attached to it, which offers information about the community to which it belongs at a given time during the iterative process. In these conditions, *node* assigns itself to the optimal community, considering the community information offered by the flags of its neighbours. The algorithm is designed in such a way that it chooses the neighbouring community to which most of *node's* neighbours belong. At the beginning of the first iteration, each node is initialized with unique community information contained in its own flag. During the course of the algorithm's iterations, accurate information describing the community structure spreads network-wide. Thus, functionally or otherwise related nodes feature, in their flag field, similar community information, after a fairly reduced number of iterations. As many such consensus-based groups of nodes build, they continue to attract nodes from smaller neighbouring communities, while it is possible to do so. When the algorithm completes its last iteration, nodes that are characterized by the same flag information are grouped in the same community. The flag information updating process is illustrated in *Figure 1* on a 4-node *complete* network configuration. We say that a network configuration is complete if, for any pair of vertices (u, v), there is an edge that connects them.

The algorithm conducts the flags updating process in an iterative manner. At every iteration, each node updates its flag considering the community information that is stored in the flags of its neighbours. Formally, this idea can be expressed by the following function: $Flag_{current\_node}(t) = update\_flag(Flag_{node_1}(t-1), Flag_{node_2}(t-1), Flag_{node_3}(t-1), ..., Flag_{node_k}(t-1))$. Here, $Flag_{current\_node}(t)$ represents the flag of the currently analyzed node at time t, which is computed through the function $update\_flag$, which takes as arguments the flag values of its neighbours, $(node_1, node_2, node_3, ..., node_k)$. Following the node processing suggestion that is introduced by

the algorithm of Blondel et al., which is presented in a previous section, the order according to which nodes are selected to have their flag updated, is established in a random manner. It is immediate to note that, while there are n flags at the beginning of the first iteration, the number reduces over the course of the algorithm, resulting in the end in as many unique flag values as there are communities.

The algorithm should continue to run until no node changes the information in its flag anymore. However, there could be nodes in the network that are featured by an equal maximum number of neighbors in two or even more neighbouring communities. We addressed this potential shortcoming by instructing the algorithm to stop when each node in the network is described by a flag value that the maximum of its neighbours have. Let us consider $\{Flag_1, Flag_2, ..., Flag_s\}$ to be the set of the flags that exist in the network at a given moment. Moreover, let us note by $number\_neighbours(current\_node, Flag_{certain\_flag})$, the number of vicinities $current\_node$ has with nodes described by a $certain\_flag$. In these conditions, it can be stated that the algorithm is stopped when, considering each $current\_node$ in the network, the following condition is true:

$number\_neighbours(current\_node, Flag_{certain\_flag})$
is less or equal than
$number\_neighbours(current\_node, Flag_{some\_flag})$,

for all possible selections of $certain\_flag$, and considering that $current\_node$ is characterized by the information stored in $Flag_{some\_flag}$.

The algorithm configures the community structure in the network by grouping together nodes with similar flags. Let us note that, obeying this stop criterion, the algorithm establishes a community structure that ensures each node is connected to, at least, as many other nodes from its own community than it is with each other extra-community node.

We are able now to state the algorithm's execution pertains to the following general steps:

1. The flag that is attached to each node in the network is initialized with the appropriate value, which respects the following rule: $Flag_{current\_node}(0) \leftarrow current\_node\_identifier$. In other words, each node's flag initially stores the identifier of the vertex, for example the number that denotes the node's rank.

2. Following, the first effective processing iteration of the algorithm is initiated. It implies the arrangement of the nodes in a random manner, let us note the resulting vertex set with $\mathcal{R}$.

3. Then, for each $current\_node \in \mathcal{R}$, and respecting the order of the elements, we have that $Flag_{current\_node}(t) = update\_flag(Flag_{node_1}(t-1), Flag_{node_2}(t-1), Flag_{node_3}(t-1), ..., Flag_{node_k}(t-1))$. Let us recall that the function $update\_flag$ updates the information stored by the flag of $current\_node$, considering the flag that appears with the highest frequency among its neighbours.

4. The algorithm continues to iterate until each node stores in its flag the same information as the majority of its neighbours do. Otherwise, $t \leftarrow t+1$, and the next iteration is initiated from step 2.

The main steps of the algorithm are presented, in a very brief manner, in *Algorithm 1.*

The actual C implementation of the algorithm is sensibly more complex than it apparently seems to be, considering the pseudo code in *Algorithm 1*, as it includes some auxiliary decision

Algorithm 1 The flag based community detection algorithm - general structure

```
//Perform some initial checks
if checkDataSet() = FALSE then
  DisplayErrorMessage("It is a protein network, it should be un-weighted!")
  return errorCode
end if
DO Create adjacency list and store edges
DO Create storage space to store and count flags
DO Create the node ordering vector nodeOrder[]
DO Arrange nodes in random order relative to nodeOrder[]
DO Considering the random order, process all nodes and assign flags accordingly
for i = 0; i < numberOfNodes; i++ do
  unprocessed[i] ← TRUE
  noOfFlags[i] ← 0
end for
for i = 0; i < numberOfNodes; i++ do
  currentFlag ← getFlagOfNode(i)
  numberNeighboursCurrentNode ← getNumberNeighboursOfNode(current_flag)
  unprocessed[i] ← FALSE
  for j = 0; j < numberNeighboursCurrentNode; j++ do
    currentFlagNeighbour ← getFlagOfNode(j)
    if unprocessed[j] = TRUE then
      DO Continue
    else
      noOfFlags[currentFlagNeighbour] ← noOfFlags[currentFlagNeighbour] + 1
      if numberNodesWithFlag(currentFlag) <
      numberNodesWithFlag(currentFlagNeighbour) then
        DO setFlagOfNode(i, getFlagOfNode(j))
      end if
    end if
  end for
end for
//At this point, nodes are properly flagged
DO Revert nodes ranks in nodeOrder[] back to their un-randomized state, preserving flag
information
```

statements and processing blocks of instructions. Nevertheless, the pseudo code summarizes the core of the process that performs the proper assignment of values to each node's flag and, as a consequence, the detection of the community structure.

2.2 Remarks Regarding the Complexity and Correctness

In spite of the *double for loop* structure that is also displayed in *Algorithm 1*, in practice, the flag based algorithm exhibits a near linear time complexity. There are several reasons that explain this efficient behaviour.

The initial allocation of flag values to each node, which is performed in the first part of the algorithm's execution, takes $O(n)$ time to complete. Considering each $current\_node$, its neighbours are initially grouped according to the information stored in their flags, which generates a worst-case time complexity of $O(number\_neighbours(current\_node))$. Then, in order to conduct all subsequent iterations, the algorithm selects the flag-related group that has the maximum size, and consequently assigns the relevant flag to $current\_node$. This selective flag filtering process is repeated for all the nodes in the network. Therefore, an overall run time of $O(m)$ is required to complete the main processing loop. The global time complexity results following a simple join operation, which takes into account the complexity required to perform the initial initializations, together with the time complexity that is generated by the main processing loop. As a consequence, it can be stated that the global time complexity of the algorithm is $O(n + m)$, where n denotes the number of nodes (proteins) in the processed network, while m represents the number of edges (biological links).

Another factor that explains the efficient behaviour of the algorithm resides in the fact the algorithm is able to correctly fill in the flags, and therefore assigns the nodes to the appropriate community, after only a few iterations. According to the experimental results of the tests we performed, the algorithm manages to fill approximately 80% of the flags with appropriate data after only three or four iterations. Furthermore, it can be stated that the community structure is fully determined after only six or seven iterations, in most cases. The next section, which offers a detailed account on all the algorithms' performance, will fully present the performance assessment process.

We have performed an assessment of the algorithm's correctness following the same empirical method with three stages, which has also been used to prove the correctness of the algorithm designed by Clauset et al. First, we determined the community structure of a 9000-node subset that was extracted from the aggregated protein data set. The output that the flag-based algorithm produced was compared to that generated by the parallel version of the Newman-Girvan algorithm. We found the two community partitions to be similar. Furthermore, we performed a comparative assessment considering the flag-based algorithm, together with Blondel's algorithm. The analysis took into account the *Amazon.com purchasing network* [22], and the aggregated protein data set. Let us recall that the Blondel's algorithm correctness has also been evaluated against the parallel version of the Newman-Girvan algorithm. We found that the flag-based algorithm produces a community structure that is identical to that generated by both reference algorithms, the Newman-Girvan parallel version and the approach of Blondel et al. Therefore, the algorithm's correctness is sufficiently demonstrated through this empirical method. Nevertheless, it is worth noting that we have also conducted a more formal verification regarding the algorithm's correctness, by going through the main steps suggested in *Algorithm 1*, and considering five randomized 20-node sample networks. The random networks were generated with Network Workbench Tool [23], and following the suggestions contained in [24]. The step by step execution of the algorithm on these networks revealed that it is able to correctly assign the flag values to the networks' nodes in all situations.

Table 1: Execution times - comparative analysis

Test number	Algorithm	Execution time
1	Flag-based	502
2	Blondel et al.	749

3 Comparative Analysis Regarding the Algorithm's Performance on Interactome Networks

The algorithm was implemented in C and run, as a sequential process, on a Beowulf class cluster of computers. Each node in the cluster provides 12 MB of cache memory for the sequential or parallel processes that run on it. Additionally, each node in the parallel world is powered up by an Intel Xeon 5420 processor, clocked at 2.5 GHz. The C language was selected as the backbone of all our implementations, because it is the closest-to-machine medium level programming language and, as a consequence, the gain in performance is noticeable.

The comparative performance testing procedure made use of an aggregated biological data set that features $22,573$ proteins and $1,886,753$ biological links. The size of this input data set determines a problem space that can be hardly handled properly in terms of execution times through a sub-optimal sequential approach.

The testing procedure conducted a comparative performance assessment that made use of the biological data provided by the aggregated protein data set. The flag-based community detection algorithm is assessed against the optimal community detection algorithm that has already been introduced, the construct of Blondel et al. Let us recall that the algorithms were implemented in C and run on the same Beowulf class cluster of computers, which has been standard during the course of our research, as sequential processes.

In *Table 1*, durations are expressed in seconds. Additionally, the speedup generated by the flag-based algorithm is suggestively displayed in *Figure 2*. It can be noticed that, compared to the fastest community algorithm, which has been introduced in the previous chapter, the flag-based algorithm performs noticeably faster. Furthermore, seven smaller protein datasets (1000, 2000, 2500, 3000, 7000, 8000, 9000 proteins) have been considered. In this respect, *Figure 3* contains a visualization of a comparative analysis concerning the algorithms' performance.

4 Remarks Concerning the Accuracy of the Algorithms' Output

Community detection algorithms determine the community structure with various degrees of accuracy. In this respect, the main problem that may impede the output of such an algorithm is represented by the possibly inaccurate allocation of nodes (proteins) to their respective communities. We have consistently faced this problem during the current phase of our research. When speaking about protein data sets that require a proper community structure detection, the accuracy of the algorithm's output is mandatory, as even the slightest community structure misconfiguration may lead to incorrect deductions and conclusions. We made use of a measure called *modularity*, which assesses the quality of the community structure determined by the algorithm.

Suppose there are k clusters in the current iteration of the algorithm. A symmetric matrix E of size $k \times k$ is constructed according to the following procedure. An element e_{ij} in E represents the fraction of all edges that link the vertices in cluster i to the vertices in cluster j and e_{ii} represents the fraction of edges that connect vertices within cluster i. Thus, summation of row (or column) elements $c_i = \sum_{i=1}^{k} e_{ij}$ represents the fraction of all edges that connect vertices to and

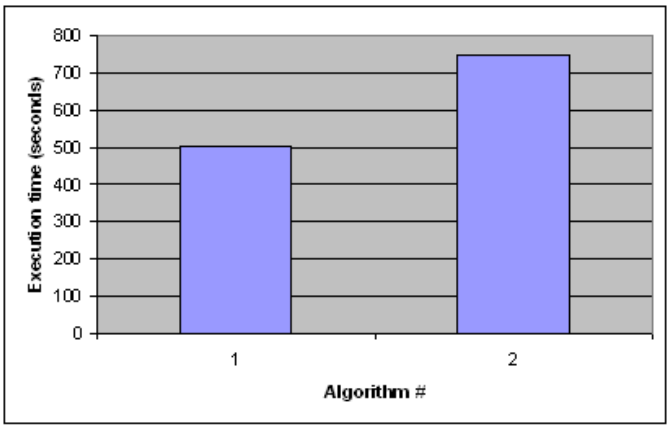

Figure 2: Community detection optimization in protein networks - the descending trend of the execution time induced by the flag-based community detection algorithm, as compared to the algorithm of Blondel et al. The algorithm numbers denote the following: 1 - Flag-based algorithm, 2 - Blondel et al.

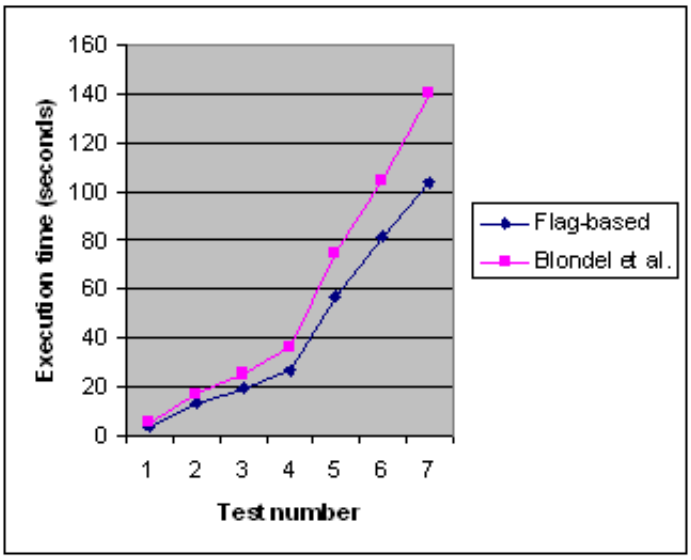

Figure 3: The flag-based community detection algorithm - comparative performance analysis on seven smaller protein datasets.

within cluster i. In these conditions, modularity is defined as $Q = \sigma_{i=1}^{k}(e_{ii} - c_i^2)$, which measures the fraction of the edges that connect vertices within the same cluster minus the expected value of the same quantity in the network [25]. For a random network with random decomposition, Q approaches 0. Values approaching $Q = 1$, which is the maximum, indicate strong clustering structure. The higher is the value, the stronger is the clustering structure in the network. The inclusion of modularity as a community assessment measure significantly improved the accuracy of the community detection process output.

In this context, it can be stated that, taking into consideration some further comparative tests we conducted, the accuracy of the community structure generated by the algorithm is comparable to that produced by the algorithm of Newman and Girvan, both in its sequential and parallel version. Therefore, the tradeoff created by the utilization of this greedy-based optimization

strategy is, in practice, nonexistent. Thus, while the sequential algorithm of Newman and Girvan generates, considering the aggregated protein data set, a community structure that is characterized by a modularity of 0.837, the flag-based algorithm builds a community structure whose modularity is 0.835. Additionally, the algorithms of Clauset et al. and Blondel et al. determine, each of them, a protein community structure whose accuracy is characterized by a modularity of 0.821 and 0.811, respectively. It is immediate to note that the flag-based algorithm is not only the fastest community detection algorithm that has been used, but also the most accurate greedy algorithm, according to the modularity of the community structure that has been determined.

5 The Algorithm and Its Suitability for Protein Networks Analysis

The flag-based community detection algorithm features a simple and intuitive structure that works around all the overheads that are inherent, in connection with sub-optimal algorithms, when a complex operation like community detection is performed. The degree of complexity and the associated overhead dramatically increase when a large networked structure, like the interactome, is processed. Therefore, suitable algorithms have to avoid the usage of less efficient data structures. Additionally, a proper balance between output's accuracy and computational time has to be sought and implemented accordingly.

The flag-based community detection algorithm has been designed with all the above principles in mind. Thus, the algorithm uses only simple array structures. Moreover, the community discovery method proposed by this algorithm determines a straightforward detection of protein communities. Thus, more than 50% of all the proteins in the aggregated data set are assigned to their respective communities after only a few iterations. This behaviour suggests that the algorithm is able to process any existing biological data set, regardless the degree of enrichment with new data in the foreseeable future.

As a final remark in this section, it is important to note that this gain in performance is not accomplished in the detriment of the output's accuracy. In this respect, it can be stated that the value of modularity demonstrates that the algorithm assigns proteins to correct communities with almost the same accuracy as the algorithm of Newman and Girvan, which is an exact construct that needs significantly more computational time. Considering all the greedy community detection algorithms that have been analyzed, the flag-based algorithm generates the most accurate proteomic community structure.

6 Analysis of Cancer-related Protein Communities

In order to extract the data that is relevant to cancer, we used the valuable data on protein families that is made available in the Pfam database [16].

Let us recall that the testing procedure made use of a compiled biological data set that features $22,573$ proteins and $1,886,753$ biological links. This size of the input data set determines a problem space that can be hardly handled properly in terms of execution times through a sequential approach.

We examined the protein communities our method determined and some interesting differences in the community sizes were noticed. Cancer proteins belong to more highly populated communities compared to non-cancer proteins. The explanation may reside in the fact that cancer proteins take part in more complex cellular (carcinogenic) processes than those proteins that are of lower importance in the interactome network and, consequently, have less influence on the

carcinogenesis. It can also be asserted that larger protein communities feature a larger or more complicated cellular mechanism, in which cancer proteins play an important role.

Proteins identified as members of more than one protein community are of particular interest. In general, each protein community represents and determines a distinct cellular process. Therefore, proteins that are part of multiple communities may generate multiple cellular processes, and can be considered to be at the intersection of distinct but adjacent cellular processes that are determined by particular protein communities, which are isolated by our community detection technique. The comparison between the cancer proteins population and the non-cancer proteins population reveals that cancer proteins reside at community junctions at a sensibly greater extent than their non-carcinogenic siblings. This particular feature of cancer proteins enforces their special importance in the interactome network seen as a whole and, as a consequence, their influence on all the physiological processes and related disorders.

Existing contributions distinguish between highly connected domains in peripheral cores (locally central) and highly connected domains in central cores (globally central). We noticed that globally central proteins represent an essential backbone of the proteome, exhibit at a high degree evolutionary conservation, and are essential for the organism. It is important to note that cancerous disease provokes mutations exactly to these globally central proteins. This observation supports and extends the findings of Wachi et al. (2005), who showed that differentially expressed proteins in squamous cell carcinoma of the lung tend to be global hubs [18]. Moreover, the findings reported in this paper support and extend the results generated by our research's previous stage. Practically, the above findings reveal the topological features of cancer proteins that are primarily displayed for cancer mutated proteins in exhibiting the highest betweenness centrality compared to the proteins that didn't loose their normal function. In other words, the carcinogenic process is generated by clusters of proteins that feature a central position in the protein network. As a consequence, the high adverse impact of any cancer form is, in our opinion, determined by the way the disease affects the fundamental proteins that coordinate the most essential processes in the metabolic and physiological chains.

The already gathered experimental information can be summed up into the following conclusions:

- The novel protein communities detection algorithm was designed and implemented and was found to accurately determine the functionally-related communities of proteins.
- We practically assessed the suitability and performance of the new approach on real proteomic data related to cancer and the interesting properties of the determined protein communities allowed us to infer an explanation regarding cancer evolution.
- The algorithm performs faster than the algorithms of Blondel et al. and Clauset et al., which represent two of the most efficient community detection solutions that have been designed by now.

6.1 Conclusions and Future Developments

The most important property of cancer proteins is their importance at the scale of the whole interactome. The flag-based algorithm was used to show that the globally central proteins are the ones that are the most affected in a carcinogenic process and are also located at the junction of the most important protein communities.

The resulting clustering algorithm allows us to explore protein-protein connectivity in a more informative way than is possible by just counting the interaction partners for each protein. It allows us to distinguish between central and peripheral hubs of highly connecting proteins, revealing proteins that form the backbone of the proteome. The fact that we observe an enrichment

of cancer proteins in this group and also their highest betweenness centrality values indicate the central role of these proteins. The domain composition of cancer proteins indicates the explanation for this topological feature: we have shown, based on our experiments' results, that cancer proteins contain a high ratio of highly malign domains. Therefore, all cancer drugs should be designed in such a way to prevent possible mutations to these highly-important proteins or, if the disease is already on the way, to contribute to reverting back to the original proteomic structure.

Moreover, it is important to note that the scientific presentation in this paper demonstrates that cautiously-designed greedy algorithms can produce an output whose accuracy is on par with that of similar conventional (exact) approaches.

The next stages of our research will involve further optimizations of the algorithms that are used for an efficient community structure detection in protein networks. Also, we intend to analyze even more biological data sets related to cancer and, possibly, other high-impact contemporary diseases.

Acknowledgment

This work is supported by the Irish Research Council for Science, Engineering and Technology, under the Embark Initiative program.

Bibliography

[1] J. Yoon, A. Blumer and K. Lee, *An algorithm for modularity analysis of directed and weighted biological networks based on edge-betweenness centrality*: Bioinformatics, 2006.

[2] M. Girvan and M.E.J. Newman, *Community structure in social and biological networks*: State University of New Jersey, 2002.

[3] D. Ucar et al., *Improving functional modularity in protein-protein interactions graphs using hub-induced subgraphs*: Ohio State University, 2007.

[4] K. Lehmann and M. Kaufmann, *Decentralized algorithms for evaluating centrality in complex networks*: IEEE, 2002.

[5] J. Griebsch et al., *A fast algorithm for the iterative calculation of betweenness centrality*: Technical University of Munchen, 2004.

[6] G.H. Traver et al., *How complete are current yeast and human protein-interaction networks?*: Genome biology, 2006.

[7] R. Bunescu et al., *Consolidating the set of known human protein-protein interactions in preparation for large-scale mapping of the human interactome*: Genome biology, 2005.

[8] U. Brandes, *A faster algorithm for betweenness centrality*: University of Konstanz, 2001.

[9] B. Preiss, *Data structures and algorithms with object-oriented design patterns in C++*: John Wiley and sons, 1998.

[10] EMBL-EBI, *The IntAct protein interactions database.* URL: http://www.ebi.ac.uk/intact/site/index.jsf, 2009.

[11] A. Grama et al., *Introduction to parallel computing, second edition*: Addison-Wesley, 2003.

[12] University of California, *The DIP protein interactions database*. URL: http://dip.doe-mbi.ucla.edu/, 2009.

[13] R. Bocu and S. Tabirca, *Betweenness Centrality Computation - A New Way for Analyzing the Biological Systems*: Proceedings of the BSB 2009 conference, Leipzig, Germany, 2009.

[14] L.C. Freeman, *A set of measures of centrality based on betweenness*: Sociometry, Vol. 40, 35-41, 1977.

[15] P.F. Jonsson and P.A. Bates, *Global topological features of cancer proteins in the human interactome*: Bioinformatics Advance Access, 2006.

[16] Wellcome Trust Sanger Institute, *The Pfam protein families database*. URL: http://pfam.sanger.ac.uk/, 2009.

[17] R. Bocu and S. Tabirca, *Sparse networks-based speedup technique for proteins betweenness centrality computation*: International Journal of Biological and Life Sciences, 2009.

[18] S. Wachi et al., *Interactome-transcriptome analysis reveals the high centrality of genes differentially expressed in lung cancer tissues*: Bioinformatics, 21, 4205-4208, 2005.

[19] G. Palla et al., *Uncovering the overlapping community structure of complex networks in nature and society*: Nature, 435, 814-818, 2005.

[20] P.F. Jonsson et al., *Cluster analysis of networks generated through homology: automatic identification of important protein communities involved in cancer metastasis*: BMC Bioinformatics, 7, 2, 2006.

[21] R. Bocu and S. Tabirca, *Proteomic Data Analysis Optimization Using a Parallel MPI C Approach*: IEEE Computer Society, The First International Conference on Advances in Bioinformatics and Applications, 2010.

[22] A. Clauset, M.E.J. Newman and Ch. Moore, *Finding community structure in very large networks*: Phys. Rev. E 70, 066111, 2004.

[23] S. Schnell, S. Fortunato and R. Sourav, *Is the intrinsic disorder of proteins the cause of the scale-free architecture of protein-protein interaction networks?*: Proteomics 7 no. 6, 961-964, 2007.

[24] V. Batagelj and U. Brandes, *Efficient generation of large random networks*: Physical Review E. 71:036113-036118, 2005.

[25] R. Bocu, *Detecting community structure in networks*: Eur. Phys. J. B 38, 321-330, 2004.

[26] V.D. Blondel, J.-L. Guillaume, R. Lambiotte and E. Lefebvre, *Fast unfolding of communities in large networks*: Journal of Statistical Mechanics, arXiv:0803.0476v2 [physics.soc-ph], 2008.