Efficiently Enumerating All Connected Induced Subgraphs of a Large Molecular Network

Sean Maxwell[1], Mark R. Chance[1], and Mehmet Koyutürk[1,2]

[1] Center for Proteomics and Bioinformatics
[2] Department of Electrical Engineering and Computer Science
Case Western Reserve University, Cleveland, Ohio, USA
{sean.maxwell,mark.chance,mxk331}@case.edu

Abstract. In systems biology, the solution space for a broad range of problems is composed of sets of functionally associated biomolecules. Since connectivity in molecular interaction networks is an indicator of functional association, such sets can be identified from connected induced subgraphs of molecular interaction networks. Applications typically quantify the relevance (e.g., modularity, conservation, disease association) of connected subnetworks using an objective function and use a search algorithm to identify sets of subnetworks that maximize this objective function. Efficient enumeration of connected subgraphs of a large graph is therefore useful for these applications, and many existing search algorithms can be used for this purpose. However, there is a lack of non-heuristic algorithms that minimize the total number of subgraphs evaluated during the search for subgraphs that maximize the objective function. Here, we propose and evaluate an algorithm that reduces the computations necessary to enumerate subgraphs that maximize an objective function given a monotonically decreasing bounding function.

Keywords: connected subgraph enumeration, protein interaction networks, branch-and-bound algorithms.

1 Introduction

For many applications in systems biology, the connected induced subgraphs of molecular interaction networks are of particular interest since they represent sets of functionally associated biomolecules. For example, in the context of the systems biology of complex diseases, medical scientists are interested in identifying "dysregulated protein subnetworks", i.e., sets of proteins connected to each other via protein-protein interactions that exhibit collective differential expression between different phenotypes [3,4,5,6]. Similarly, gene set enrichment analysis aims to evaluate the statistical significance of the aggregate disease association of sets of genes that are defined *a priori*, and the connected subgraphs of molecular networks provide excellent candidate gene sets since they are functionally related through physical and functional interactions [15]. At the evolutionary scale, sets of orthologous proteins that induce connected subgraphs on networks of different species are shown to be useful in gaining insights into the conservation and modularity of biological processes across diverse taxa [7,11,16].

In all of these applications, an objective function is defined to score any given subnetwork in terms of its relevance to what is being sought by the application. For example, in the identification of disease-associated subnetworks, connected subnetworks that contain a large number of disease-associated gene products are of interest. This scoring function may be computed based on the network topology alone, or may also incorporate other data, such as gene expression [5], genome-wide association [10], or sequence homology [11]. Then the problem is abstracted as one of finding high-scoring (e.g., globally optimal, locally optimal, or above a certain threshold) subnetworks according to this scoring function.

Due to computational considerations, most methods designed to tackle these problems implement heuristic algorithms to search the space of connected induced subgraphs of a network. However, it was shown that exhaustive search may lead to the identification of more biologically relevant patterns as compared to those identified by simple heuristics [4,16,18]. Furthermore, it is often desirable to identify many high-scoring subnetworks as candidates to be further evaluated for statistical significance, as opposed to identifying a single subnetwork with maximum score.

The objective of this work is to develop efficient algorithms for enumerating all sets of vertices that induce a connected subgraph in a large network. Our main motivation is to facilitate effective exploration of the subnetwork space of molecular interaction networks by enabling pruning of the search space in large chunks. For this purpose, we focus on the case where the scoring function satisfies a *hereditary property.* A hereditary property in a graph $G = (V, E)$ is a property such that if a set $S \subseteq V$ of nodes satisfies the property, then all subsets $S' \subseteq S$ also satisfy the property [2]. For example, being a clique is a hereditary property because any induced subgraph of a clique is also a clique. The bounding functions used by branch-and-bound algorithms also exploit hereditary properties. For the purpose of finding all maximal cliques or finding all maximal vertex sets $S \subseteq V$ that "score" greater than a specified threshold, the hereditary property is useful for pruning out the search space. This is because, if the property does not hold for a vertex set S, then no superset of S needs to be evaluated.

Many well established algorithms exist to enumerate connected induced subgraphs, such as ReverseSearch [1] and Algorithm447 [8] which are both variations of depth first search, and new algorithms such as ConSubG[12] are an area of active research. For the purpose of exploiting a hereditary property to prune out the search space, conventional depth first enumeration algorithms exhibit an inherent drawback: These algorithms do not enumerate vertex sets in an order that will allow evaluation of a vertex set after all of its subsets are evaluated. In other words, if a connected induced subgraph $S \subseteq V$ does not satisfy the hereditary property, depth first enumeration methods are likely to needlessly enumerate many $S' \supset S$ either before or after S is evaluated and rejected. We refer to such redundant computations as "*unnecessary rejections*".

Here, we propose an enumeration algorithm that introduces two novel techniques to reduce the number of unnecessary rejections while searching for connected induced subgraphs satisfying a hereditary property: 1) We use *anchor* *vertices* to seed the search, with a view to enabling easy tracking of the connectedness of the set of vertices being enumerated. 2) We use a *breadth-first discovery, depth-first extension* approach to enumerate sets of vertices, with a view to enabling evaluation of most vertex sets before their supersets are enumerated.

We systematically evaluate the ability of the proposed algorithm in reducing the number of unnecessary rejections and the resulting earnings in terms of runtime. Our results show that the proposed method significantly reduces the number of unnecessary rejections without introducing additional overhead into the enumeration itself.

2 Methods

2.1 Problem Definition and Observations

Let $G = (V, E)$ be an undirected graph. A set $V' \subseteq V$ is said to be a *connected vertex set* if the subgraph induced by V' is connected, i.e., if for every pair of vertices $\{u, v\} \in V'$, there is a path in G from u to v that goes only through nodes in V'. Throughout this work we refer to connected node sets as S where it is implied that $S \subseteq V$ and S induces a connected subgraph of G.

Let $f : 2^V \to \mathbb{R}$ be a function used to score vertex sets. For example, if we are interested in identifying maximal cliques, then we can define $f(S) = |S|$ if S induces a clique, and 0 otherwise. If we are interested in identifying disease-associated subnetworks such that the disease association of vertex $v \in V$ is quantified as $\sigma(v)$, then we can define $f(S) = \sum_{v \in S} \sigma(v)/\sqrt{|S|}$ [9].

We consider a problem setup where we are given a threshold t, and we are interested in enumerating all connected node sets $S \subseteq V$ such that $f(S) \geq t$. We assume that we are given a bounding function $f_b : 2^V \to \mathbb{R}$ such that, for any $S' \supseteq S$, $f(S') \leq f_b(S)$. We say that node set S is rejected if $f_b(S) < t$. The bounding function is useful for pruning out the search space using a bottom-up enumeration algorithm, since $f_b(S) < t$ implies $f(S') < t$ for all $S' \supseteq S$, i.e., once S is evaluated and rejected, there is no need to generate and evaluate any superset of S. In general terms, this problem can be viewed as one of generating all maximal connected vertex sets that satisfy a given hereditary property.

In order to efficiently generate all maximal vertex sets that satisfy a hereditary property, we need an algorithm to enumerate the solution space correctly and efficiently. Any algorithm that solves this problem has to satisfy the following criteria in order to be correct and optimal:

- *Completeness:* All connected vertex sets S in G for which $f(S) \geq t$ should be generated and all generated vertex sets should be connected.
- *No redundant subgraph generation:* Each connected node set in G should be generated exactly once.
- *Optimal order of enumeration:* If S' and S are connected node sets and $S' \subset S$, then S' should be generated before S so that if $f_b(S') < t$ we try to avoid generating S.

The "completeness" criterion relates to the correctness of the algorithm while the "no redundant subgraph generation" and "optimal order of enumeration" criteria relate to efficiency. The "no redundant subgraph generation" criterion asserts that each candidate solution in the solution space should be considered exactly once since additional considerations will lead to redundant computation. The "optimal order of enumeration" criterion, on the other hand, facilitates optimal pruning of the search space by ensuring that all subsets of a connected node set are considered before the node set itself is considered.

While depth-first enumeration approaches satisfy the first two criteria, they lead to many unnecessary rejections because the depth-first order of enumeration does not satisfy criterion three. Avoiding all unnecessary rejections likely requires a breadth-first enumeration, but memory can be a limiting factor for breadth-first approach. Here, we propose a more balanced approach that keeps the size of the problem manageable while reducing the number of redundant rejections.

2.2 Anchor Vertices

We first observe that the "completeness" and "no redundant subgraph generation" criteria can be satisfied by selecting a single $v \in V$ as an *anchor vertex* and enumerating all subgraphs containing v before removing v from G. In this way each $v \in V$ is chosen as a starting point and all subgraphs containing it are enumerated before v is removed from G. When $V \equiv \emptyset$ all subgraphs have been enumerated. An example of enumerating connected induced subgraphs from an anchor vertex is shown in Figure 1.

It is clear that, for some $S' \subset V$, this process generates many $S \supset S'$ before S' itself, and thus it does not satisfy the criterion of "optimal order of enumeration". The number of these unnecessary rejections can be reduced using heuristic choices for the anchor vertex based on measures of centrality (e.g., degree or betweenness centrality). In this work, we rather focus on reducing unnecessary rejections within each search anchored at a given vertex. In the following, we first describe our approach for reducing unnecessary rejections in the local search comprised of the anchor vertex and its neighbors, and then generalize our method to all connected induced subgraphs that contain the anchor vertex.

2.3 Efficient Enumeration of Spokes

Observe that, for a given anchor vertex $v \in V$, the neighbors of v can be treated as a set because v and any combination of its neighbors induce a connected subgraph of G. This collection of subgraphs ("spokes") can be represented as a *binomial tree*. A binomial tree is a data structure that can be used to enumerate all subsets of a set [14]. Any node n of a binomial tree has children that are copies of all branches rooted at siblings that precede n in the tree. The binomial tree that enumerates all spokes around the anchor vertex has a root node r labeled by the anchor vertex v and all descendants are labeled by neighbors of v. Since all vertices labeling nodes in T are connected to the vertex labeling the root, the

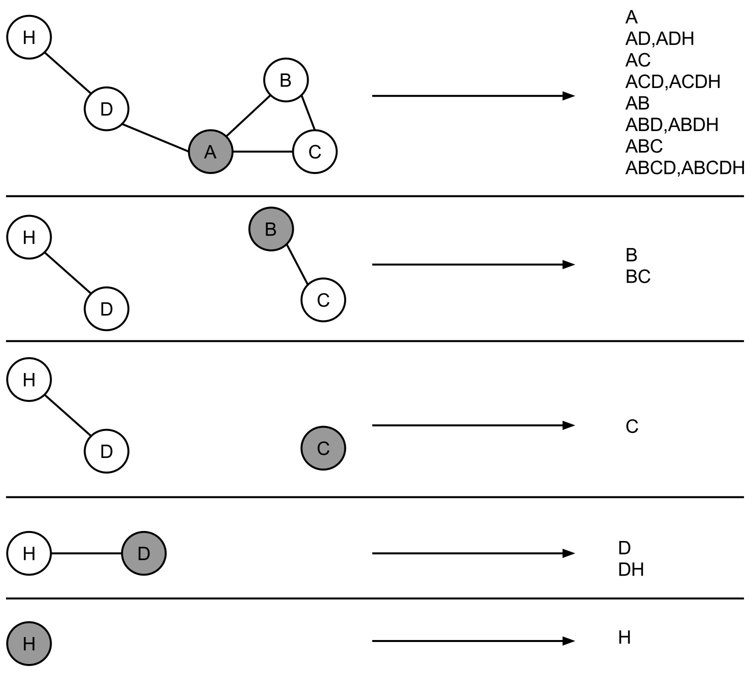

Fig. 1. Example illustrating the enumeration of all connected induced subgraphs of a graph using anchor vertices. On the left is the graph as each vertex becomes the anchor used for enumeration of all connected subgraphs that contain the anchor before it is subsequently removed from G. On the right are the connected subgraphs generated from each anchor vertex.

set of vertices that label each path from the root r of T to a node n represents a connected subgraph of G.

A simple method that *reduces* unnecessary rejections using a binomial tree based approach constructs the tree by adding each neighbor vertex to the root as a new node n, and then adding copies of the branches rooted at each sibling of n as children of n. Copying a branch terminates whenever the set S represented by the path does not satisfy $f_b(S) \geq t$. The resulting local search is similar in spirit to the set enumeration tree (SE-tree) search of Rymon [19]. However, our approach is more closely related to the binomial tree because we construct an explicit tree where the set is defined by the path from the root to a node in the tree. It is important to note that depending on how rejections occur, T may no longer meet the definition of a binomial tree so moving forward we will refer to T as a *local search tree.* An example of constructing a local search tree is shown in Figure 2.

The local search method can be extended to vertices beyond the direct neighbors of the anchor vertex by following a path of T and treating the vertices that label the path as an *anchor set* around which another local search tree is constructed as shown in Figure 3. This leverages the local search for each anchor set

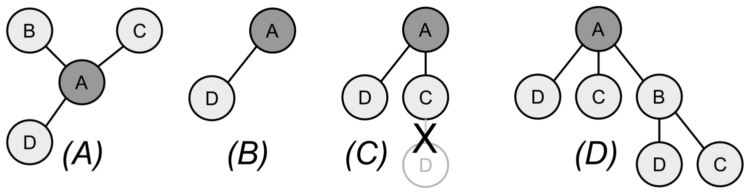

Fig. 2. Creating the local search tree T. **(A)** The input graph G with the anchor vertex A. **(B)** Exploring D with no previous branches yields the D branch. **(C)** Exploring C with the previous D branch evaluates ACD which is rejected resulting in branches C and D. **(D)** Exploring B with previous branches evaluates ABD and ABC (avoiding ABCD which contains the previously rejected ACD).

but the information is not used globally. In order to use the information from previous rejections globally we must modify our procedure as outlined in the following section.

2.4 Efficient Enumeration of All Connected Subgraphs

With slight modification we can combine the local search tree strategy with a conventional depth first approach to enumerate all subgraphs that contain the anchor vertex. Rather than using neighbors to construct the tree, we use the branches generated by depth first search through each neighbor to generate the tree. The top level procedure performs the depth portion of the search and it marks all neighbors as visited so they cannot be reached by continued depth search. The search space of neighbors is explored by the procedure that builds the local search tree from depth branches which we consider the breadth procedure. The only modification required to ensure correctness is that the breadth procedure stops cloning branches at nodes labeled by unvisited vertices adjacent to the vertex labeling the node that is appending the branch. This method represents our solution to the general case and is formalized in the **B**readth-first **D**iscovey, **D**epth-first **E**xploration (BDDE) algorithm. An example of the tree constructed by BDDE is shown in Figure 4 (C). An example of why the stop condition is necessary is show in Figure 4 (B).

2.5 Correctness

The following theorems are based on supporting lemmas in the supplementary materials[1]. Theorem 1 guarantees that our method satisfies the "completeness" and "no redundant subgraph generation" criteria during exhaustive enumeration, i.e, when f_b is satisfied by any S. Theorem 2 guarantees that our method satisfies the "optimal order of enumeration" criterion during exhaustive enumeration. Theorem 3 guarantees that our method satisfies the "completeness" criterion when f_b is selective.

[1] `http://statler.case.edu/smaxwell/alcob2014/supplement.pdf`

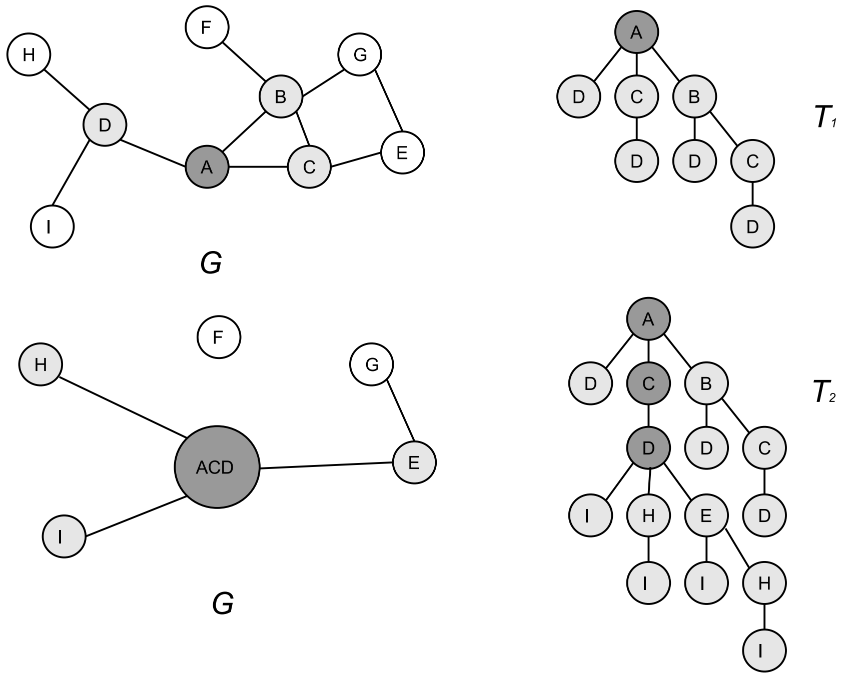

Fig. 3. Example illustrating the key idea of *anchor sets*. Initial tree T_1 generated from anchor vertex A in G (top) is extended by following path ACD in T_1 and treating the vertex set {ACD} as an anchor set. The anchor set can be isolated in G (bottom) the same way as an anchor vertex and a new local search tree is constructed anchored at ACD and appended to T_1 resulting in T_2. In this way all $S \subseteq V$ that contain the anchor vertex can be enumerated.

Theorem 1. *Given an input graph G, an anchor vertex $v \in V$ and a function f_b s.t. for any S, $f_b(S) \geq t$, BDDE uniquely enumerates all $S \subseteq V$ containing v.*

Proof: By Lemma 4 we know that the set represented by any path $P(n_k)$ in T induces a connected subgraph of G and by Lemma 5 we know that every path in T represents a unique set. By Lemma 7 we know that all $S \subseteq V$ containing v are represented by a path $P(n_k)$ in T. Therefore, we can conclude that because BDDE enumerates all paths of T, BDDE uniquely enumerates all $S \subseteq V$ containing v. $\square$

Theorem 2. *Given an input graph G, an anchor vertex $v \in V$ and a function f_b where $f_b(S) \geq t$ for any $S \subseteq V$, BDDE enumerates all connected induced subgraphs of G containing v in an order such that all $S' \subset S$ containing v are enumerated before S.*

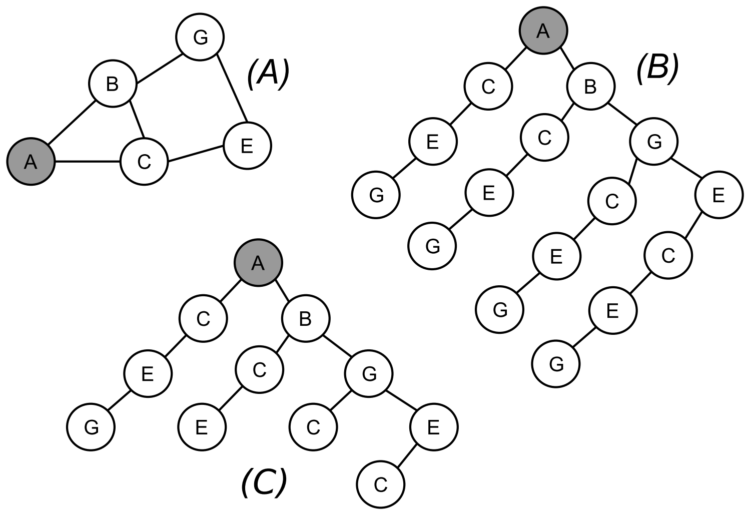

Fig. 4. (A) Graph G with anchor vertex A highlighted. **(B)** The enumeration tree generated by appending an un-pruned branch generated from $S =$AC to the branch generated through $S =$AB which exhibits several redundant instances of G and E. **(C)** The tree generated by pruning the branch generated through $S =$AC as it is added to the branch generated through $S =$AB and subsequent children.

Proof: By Theorem 1 we know that all $S \subseteq V$ containing v that induce a connected subgraph of G are enumerated, and by Lemma 8 we know that any $S' \subset S$ containing v must be generated before S. □

Theorem 3. *Given an input graph G, an anchor vertex v and a function f_b, if $f_b(S') < t$ all $S \not\supset S'$ are still enumerated by BDDE.*

Proof: We know by Theorem 1 that all $S \subseteq V$ containing v are enumerated by BDDE when no rejections occur, and by Lemma 9 we know that when a rejection of S' occurs it only eliminates $S \supset S'$. Therefore, we conclude that if an S' is rejected all $S \not\supset S'$ are still enumerated. □

3 Experimental Results

In order to systematically evaluate the performance of BDDE, we define a problem with a simple objective function that allows investigation of the effect of various parameters on performance. For this purpose, we use real-world networks to ensure that the network topology is practically relevant. We assign positive weights to all vertices of each network from a Gaussian distribution with mean m and standard deviation ρ. We define the objective function $f(S) = 1/\sum_{v \in S} w(v)$, where $w(v)$ denotes the weight of vertex v. Then, for a given threshold t and a maximum subgraph size k, we search for all connected subgraphs $S \subseteq V$ with $|S| \leq k$ and $f(S) \geq 1/t$ (note $1/t$ is used because we want to enumerate all subgraphs with a total weight less than t).

Datasets. In the experiments reported in this section, we utilize two real-world networks: 1) A citation network generated by Leskovec *et al.* [17] from the on-line arXiv journal, which consists of 5,241 vertices and 28,958 edges. 2) The protein-protein interaction (PPI) network obtained from the Human Protein Reference Database (HPRD) [13], which consists of 9,455 vertices and 37,080 edges.

Results. For each network, we set $m = 10$ and generate ten instances each for different values of ρ ranging from 1 to 9. On each instance, we perform enumerative search using the proposed algorithm and a standard DFS-based algorithm for t ranging from 5 to 50 for arXiv and 5 to 40 for HPRD. For each combination of ρ and t, we report the average of the performance measures across the ten randomized instances. The results for the HPRD network are shown in Figure 5. The results for the arXiv network follow similar trends and are available in the online supplemental materials.

Figure 5 shows the *rejection rate* for each algorithm. Here, rejection rate is defined as the fraction of subgraphs S with $f(S) < 1/t$ among all subgraphs that are enumerated. The rejection rate for BDDE is consistently lower than that of DFS though the relationship is more obvious at higher values of ρ. This relationship between ρ and rejection rate is indeed expected, since pruning is less effective when the weight distribution is more uniform across vertices. At the extreme case, when ρ=0 (all vertices have equal weight), only the leaf nodes of the enumeration tree are rejected and neither algorithm can take advantage of the knowledge on smaller subgraphs to prune out larger subgraphs. But as ρ grows, the enumeration tree becomes more imbalanced, and the benefit of the order of generation implemented by BDDE becomes more apparent.

Our computational tests also show that the two algorithms have similar runtimes for smaller values of the threshold t (when larger subgraphs are less likely to satisfy the objective criterion) and smaller values of ρ (when the vertex weights are more uniform). However, for larger values of t and ρ, BDDE consistently outperforms the DFS-based method. While runtime comparisons are largely implementation dependent we find this is a positive outcome and include additional figures and analysis in the supplementary material.

4 Conclusion

We have investigated the problem of reducing the number of subgraphs evaluated while enumerating all connected induced subgraphs $S \subseteq V$ that satisfy a hereditary property. Our proposed method displays a significant decrease in total number of subgraphs evaluated during enumeration compared to a classical depth first branch and bound approach. In addition, Theorems 1, 2 and 3 provide proof of correctness that all connected induced subgraphs S that satisfy $f_b(S) \geq t$ are enumerated. However, due to the potential for our method to use space exponential to the maximum size of S being enumerated, our method is best suited to enumerating all $|S| \leq k$ from G where k is chosen appropriate to the problem and the available memory.

Algorithm 1. The BDDE algorithm. Enumerates all S that contain anchor vertex v and satisfy $f_b(S) \geq t$. Returns the root node of the enumeration tree T. Entry point is DEPTH($\emptyset$,v,[ ]).

```
 1: procedure BREADTH($S, n, U$)
 2:     if $v_n \in U$ then                         ▷ Prune branch by stop condition
 3:         return null
 4:     end if
 5:
 6:     $S' \leftarrow S \cup v_n$                  ▷ Prune branch by bounding function
 7:     if $f_b(S') < t$ then
 8:         return null
 9:     end if
10:
11:     $n' \leftarrow \Upsilon(v_n)$               ▷ Create a new tree node labeled by $v_n$
12:     for all $\{n^* : nn^* \in B\}$ do           ▷ Recursively copy child branches
13:         $n'' \leftarrow$ BREADTH($S', n^*, U$)
14:         if $n'' \neq null$ then
15:             $B \leftarrow B \cup n'n''$          ▷ Append child branch
16:         end if
17:     end for
18:     return $n'$
19: end procedure

20: procedure DEPTH($S, v, \beta$)
21:     $S' \leftarrow S \cup v$
22:     if $f_b(S') < t$ then
23:         return null
24:     end if
25:     $n \leftarrow \Upsilon(v)$
26:     $\beta' \leftarrow [\,]$
27:     for $i = 1$ to $|\beta|$ do
28:         $n' \leftarrow$ BREADTH($S', \beta[i], \chi_n$)
29:         if $n' \neq null$ then
30:             $B \leftarrow B \cup nn'$
31:             push($\beta', n'$)
32:         end if
33:     end for
34:     for all $v \in \chi_n$ do                   ▷ Note: Derive $\chi_n$ from $S$ and $v$
35:         $n' \leftarrow$ DEPTH($S', v, \beta'$)
36:         if $n' \neq null$ then
37:             $B \leftarrow B \cup nn'$
38:             push($\beta', n'$)
39:         end if
40:     end for
41:     return n
42: end procedure
```

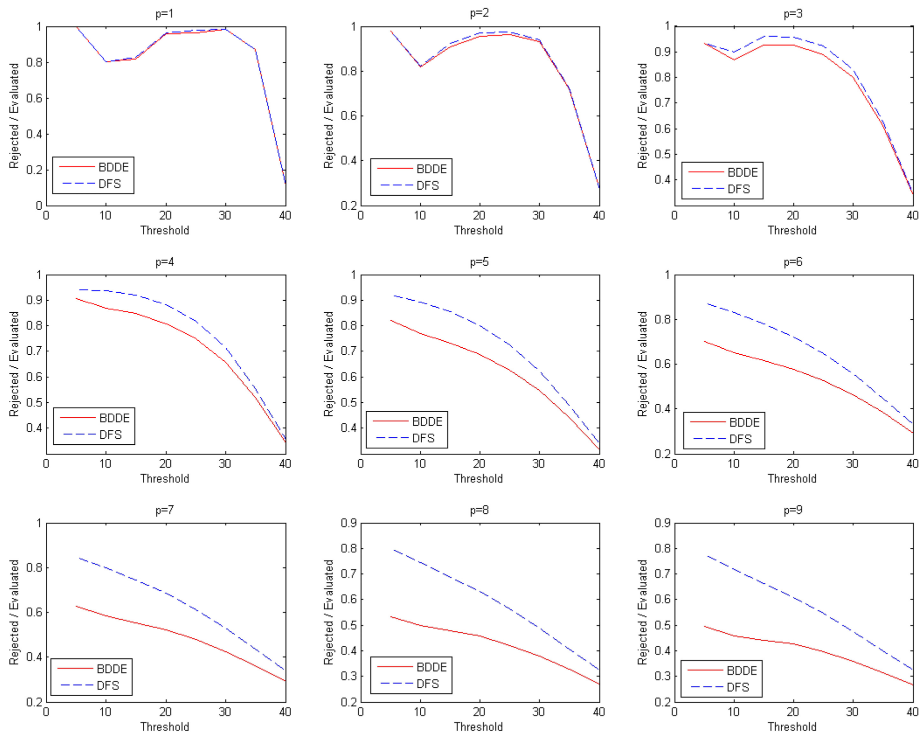

Fig. 5. Rejection rate analysis for enumerating all subgraphs up to size 4 satisfying $f(S) \geq 1/t$ in the HPRD network. Each pane plots the average rejection rate (fraction of subgraphs S with $f(S) < 1/t$ among all subgraphs that are enumerated) for each algorithm versus threshold $5 \leq t \leq 40$ where the node scores were sampled from a Gaussian distribution with mean m=10 and standard deviation $1 \leq \rho \leq 9$.

Acknowledgements. We would like to thank Harold Connamacher for his invaluable feedback on our method and related work. This work is supported in part by the Case Comprehensive Cancer Center (Grant P30-CA-043703) and US National Science Foundation (Grant CCF-0953195). This publication was also made possible by the Clinical and Translational Science Collaborative of Cleveland, UL1TR000439 from the National Center for Advancing Translational Sciences (NCATS) component of the National Institutes of Health and NIH roadmap for Medical Research. Its contents are solely the responsibility of the authors and do not necessarily represent the official views of the NIH.

References

1. Avis, D., Fukuda, K.: Reverse search for enumeration. Discrete Applied Mathematics (1993)
2. Bollobas, B.: Hereditary properties of graphs asymptotic enumeration global structure and colouring. Documenta Mathematica, 333–342 (1998)

3. Chowdhury, S., Koyuturk, M.: Identification of coordinately dysregulated subnetworks in complex phenotypes. In: Berger, B. (ed.) Pacific Symposium on Biocomputing, pp. 133–144 (2010)
4. Chowdhury, S., Nibbe, R., Chance, M., Koyuturk, M.: Subnetwork state functions define dysregulated subnetworks in cancer. Journal of Computational Biology 18(3), 263–281 (2011)
5. Chuang, H.Y., Lee, E., Yu-Tsueng, L.D., Ideker, T.: Network-based classification of breast cancer metastasis. Molecular Systems Biology (2007)
6. Dao, P., Wang, K., Collins, C., Ester, M., Lapuk, A., Sahinalp1, S.C.: Optimally discriminative subnetwork markers predict response to chemotherapy. Bioinformatics (July 2011)
7. Flannick, J., Novak, A., Srinivasan, B., McAdams, H., Batzoglou, S.: Graemlin: General and robust alignment of multiple large interaction networks. Genome Research (2006)
8. Hopcroft, J., Tarjan, R.: Efficient algorithms for graph manipulation. Communications of the ACM 16(6) (1973)
9. Ideker, T., Ozier, O., Schwikowski, B., Siegel, A.F.: Discovering regulatory and signalling circuits in molecular interaction networks. Bioinformatics 18(suppl. 1), S233–S240 (2002), http://dx.doi.org/10.1093/bioinformatics/18.suppl_1.s233
10. Jia, P., Zheng, S., Long, J., Zheng, W., Zhao, Z.: dmGWAS: dense module searching for genome-wide association studies in protein-protein interaction networks. Bioinformatics 27(1), 95–102 (2011)
11. Kalaev, M., Smoot, M., Ideker, T., Sharan, R.: Networkblast: comparative analysis of protein networks. Bioinformatics (2008)
12. Karakashian, S., Choueiry, B.Y., Hartke, S.G.: An algorithm for generating all connected subgraphs with k vertices of a graph (May 2013), http://www.math.unl.edu/~shartke2/math/papers/k-subgraphs.pdf
13. Kesheva, P., et al.: Human protein reference database: 2009 update. Nucleic Acids Research 37, 767–772 (2009)
14. Knuth, D.: The Art of Computer Programming, Combinatorial Algorithms Part 1, vol. 4. Addison-Wesley (2012)
15. Konga, B., Yanga, T., Chenb, L., Qin Kuanga, Y., Wen Gua, J., Xiaa, X., Chenga, L., Hai Zhang, J.: Proteinprotein interaction network analysis and gene set enrichment analysis in epilepsy patients with brain cancer. Journal of Clinical Neuroscience (2013)
16. Koyutürk, M., Kim, Y., Subramaniam, S., Szpankowski, W., Grama, A.: Detecting conserved interaction patterns in biological networks. Journal of Computational Biology (2006)
17. Leskovec, J., Kleinberg, J., Faloutsos, C.: Graph evolution: Densification and shrinking diameters. ACM Transactions on Knowledge Discovery from Data (2007)
18. Patel, V., Gokulrangan, G., Chowdhury, S., Chen, Y., Sloan, A., Koyutrk, M., Barnholtz-Sloan, J., Chance, M.: Network signatures of survival in glioblastoma multiforme. PLOS Computational Biology 9 (2013)
19. Rymon, R.: Search through systematic set enumeration. Tech. rep., University of Pennsylvania (August 1992)