Journal of Naval Science and Engineering
2012, Vol.8 , No.2, pp. 87-102

NETWORK MOTIFS AND INDEXING TECHNIQUES IN BIOLOGICAL NETWORKS

Yusuf KAVURUCU

Turkish Naval Academy, Tuzla, 34940, Istanbul, Turkey

ykavurucu@dho.edu.tr

Abstract

Subgraphs that occur in complex networks with significantly higher frequency than those in randomized networks are called Network Motifs. Such subgraphs are the basic building blocks of complex networks. They often play important roles on functioning of those networks. Finding network motifs is a computationally challenging problem. Finding network motifs often requires solving subgraph isomorphism problem which is NP-complete. Instead of this, several methods apply similarity queries on biological networks to find similar patterns that exist frequently according to given threshold values. As these networks are stored in databases, we need efficient methods for accessing and querying these databases. As these networks are generally represented as graphs in theory, several graph indexing methods are developed for answering queries on them. This paper summarizes network motifs and indexing techniques in biological networks.

BİYOLOJİK AĞLARDA AĞ MOTİFLERİ VE İNDEKSLEME TEKNİKLERİ

Özetçe

Karmaşık ağlarda rastgele oluşturulmuş ağlara oranda önemli derece daha fazla sıklıkta bulunan alt ağlar ağ motifleri olarak adlandırılır. Söz konusu alt ağlar ilgili karmaşık ağın temel yapı taşlarıdır. Bunlar genellikle ait oldukları karmaşık ağlarda önemli roller oynarlar. Ağ motiflerinin bilgisayar vasıtasıyla tespit edilmesi zor bir problemdir. Ağ motiflerinin tespiti genellikle NP-complete zorluk derecesine sahip alt ağ izomorfizm probleminin çözümünü gerektirir. Bunun yerine, çeşitli yöntemler, biyolojik ağlarda tanımlı oranlardan daha fazla sıklıkta bulunan benzer yapıları

tespit etmek için benzerlik sorguları uygularlar. Bu ağlar veritabanlarında saklanıldığından, bu veritabanlarına hızlıca erişebilecek ve veritabanını sorgulayabilecek etkili yöntemlere ihtiyaç duymaktayız. Söz konusu ağlar teorik olarak genelde çizge yapısında tanımlandığı için bu sorguları cevaplamaya yardımcı olacak çeşitli çizge indeksleme teknikleri geliştirilmiştir. Bu çalışmada, biyolojik ağlardaki ağ motifleri ve indeksleme teknikleri hakkında özet bilgi sunulmuştur.

Keywords: Biological Networks, Network Motif, Graph Indexing
Anahtar Kelimeler: Biyolojik Ağlar, Ağ Motifi, Çizge İndeksleme

1. INTRODUCTION

One of the fundamental goals of molecular biology is to understand the biological processes that are the driving forces behind organisms' functions. Recent advances in high throughput technology have resulted in an explosion of bioinformatics data to achieve this goal. Complete genome sequences of more than 100 organisms are now determined. Datasets containing DNA [1], proteins [2], comparative genomics data [3] and networks [4] are widely available.

Out of different bioinformatics data, biological networks constitute one of the most important classes. While most bioinformatics data, such as DNA and protein sequences and protein structures, show the contents and structure of individual bio-chemical entities, biological networks show how different bio-chemical entities interact with each other to perform vital functions. Understanding these interactions is critical as they can reveal significant information that is impossible or very difficult to achieve by analyzing individual entities that make up these interactions.

Numerous applications follow an interaction pattern that resembles biological networks. Wireless networks, sensor networks, homeland security, defense analysis, operations management are just a few examples to these applications. A critical common property of these applications is that although the individual entities may have specific function, they serve a role in the entire network by communicating with other entities. Thus,

eliminating or altering a single entity does not only affect the function of that entity, it may affect the function of other entities.

In order to find sub-networks with similar communication patterns in two networks, it is not sufficient to look at the identities of the nodes and the topologies of the matching sub-networks. This is because even when the two sub-networks are identical, their connection to the rest of their networks contains valuable information such as how fast the information disseminates and how robust the sub-network is to alterations. Therefore, role of this sub-network in the entire network has to be considered in order to obtain an accurate understanding of its function.

A fundamental question on networks is identification of similarities between them. Finding similarities between two networks requires computing a mapping of the interacting entities of the input networks. The graph/subgraph isomorphism problems can be reduced to global/local network alignment problems in polynomial time. Given that the graph and subgraph isomorphism problems are GI-complete and NP-complete respectively [5], network alignment problem is GI and NP complete, too. A method designed for aligning two networks must handle the following difficulties:

* Each sub-network as well as the entire network shows a process. Therefore, network comparison cannot be considered as a mere graph alignment. The alignment should reflect parts of networks that have similar impacts on their organisms. In addition to focusing on the topological similarities and the similarities between the bio-chemical contents of the entities, the method should also consider the impact of the sub-networks on the alignment.

* The method should be able to search the similarities between a query network and a database of potentially large number of networks. A trivial solution is to sequentially align the query network to all the database networks one by one. This, however, is not practical as the comparison of even a pair of networks is a costly operation.

* The method should not only align two networks in which the interacting entities are compatible with each other (such as protein interaction networks - all the entities are proteins), but also can align networks having different types of entities (such as metabolic networks where enzymes interact through reactions and compounds).

2. NETWORK MOTIFS

Biology of a living cell involves many intricate networks of interdependent events and interactions among biomolecules. Examples of such networks include transcriptional or gene regulation networks, protein–protein interaction (PPI) networks, metabolic pathways, neural networks, etc. In 2002, Milo et al. [7] showed that networks from diverse fields (biological and non-biological) contain several small topological patterns that are so frequent that it is unlikely to occur by chance. Different networks tend to have different sets of such frequent local structures. These patterns, referred to as 'network motifs', are recognized as 'the simple building blocks of complex networks' [7]. The discovery spawned a multitude of research efforts in the past decade and the area is fertile to this day. Network motifs are also studied in such other networks as the electronic circuits and power distribution networks, ecological networks (food web), software engineering diagrams, molecular structures, World Wide Web (the Internet), and social networks, etc.

Biologists are interested in knowing whether the functional behavior of a motif can be predicted from its structural topology as well as whether the abundance in appearance of such a motif necessarily implies biological significance. Some studies also investigated how network motifs might be shaped by evolution. For example, when a network is placed under fixed environmental conditions, evolution optimizes the network topology for some specific functions, and no motifs form in this process [8]. But, when the same network is placed under varying environmental conditions where each condition demands different functional behavior from the network, several network motifs emerge. This happens since motifs, although having the same topology, are able to perform different tasks in different input

conditions. Yet other studies have argued that overabundance of a network substructure might be a secondary result of some other phenomena [9], or that network motifs might not have evolutionary traces [10].

All in all, the ability to computationally determine motifs in a given network is an essential step in furthering these research efforts. Given a network G and a set of random graphs, we need to identify all k-node (equivalently, size-k, throughout the article) sub-graphs that are statistically overrepresented in G. However, one of the difficulties is that determining if two graphs are topologically equivalent requires 'graph isomorphism' checking, a highly computation-intensive problem with no known polynomial-time solution. This problem is compounded by the fact that the number of sub-graphs of a given size in a network is exponential in both the network size and the sub-graph size. Moreover, real-life networks tend to be large and dense in many cases. The computations also need to be carried out on a large number of random graphs, typically ranging from hundreds to one thousand. By far, even the best-known algorithms cannot find motifs with more than 10 nodes in a large, dense network within a practical time frame without doing heuristics [11].

Motif finding algorithms use various strategies in order to overcome these difficulties. One of the notable strategies is the use of 'sub-graph sampling' through the target network instead of 'exact enumeration' to acquire an acceptable turn-around time. Another strategy is to generate all possible sub-graphs of a fixed size, and for each sub-graph count its frequency in the target network. The latter strategy, called 'motif-centric approach', can lead to reduction in isomorphism-related computations when coupled with other strategies, namely 'symmetry breaking' and 'mapping' [12]. However, this strategy suffers when looking for larger motifs as the number of sub-graphs of a given size grows exponentially [13].

3. STRATEGIES FOR MOTIF-FINDING ALGORITHMS

Tasks involved in finding network motifs typically include the definition of frequency concepts, random graph generation, determining

statistical significance of the frequency of a sub-graph and deciding sub-graph isomorphism, etc.

The frequency of a subgraph H in the input graph G is the number of different occurrences of H in G. Those different occurrences may have common vertices and edges according to different definitions of frequency. As an example, three different types of frequency definitions are given in [16]. In these definitions, overlap of vertices and edges are handled differently. In the first one, overlap of both vertices and edges in different subgraph matches are allowed and every match is added into frequency value. The second frequency definition only allows vertex overlaps and counts edge-disjoint matches. Finally, the third one disallows overlapping in different subgraph matches.

According to different types of graphs (directed or undirected, labeled or unlabelled), various isomorphism concepts can be described, but generally, two graphs G and H are called as isomorphic if there is a mapping between their vertices such that each edge in G can be mapped to an edge in H and vice versa. Checking whether two graphs are isomorphic is a NP problem. On the other hand, a generalized version of this problem, namely subgraph isomorphism problem, is related with two graphs and checks whether one graph contains a subgraph which is isomorphic to other graph. This problem is NP-complete. Nevertheless, there exist some techniques such as canonical labeling [17] to solve this problem for small subgraphs in practice. In addition to this, canonical labeling was used for efficient pruning schemes, with respect to graph isomorphism, in some of the techniques presented in [18, 19].

Network motifs are not only recurrent structures in the given network, but also they have statistical significance with respect to some threshold values. These values can be described according to some concepts such as frequency, uniqueness, P-value and z-score. A subgraph is frequent if it occurs more than a threshold (such as F) in the given network. Additionally, a subgraph is unique if its frequency in the given network is higher than (at least a certain amount) its mean frequency in the randomized

graphs. The P-value is the probability of whether the frequency of the subgraph in a randomized network is greater than or equal to its frequency in the given network. That subgraph is called as statistically significant if P-value is less than a threshold (depends on the network). The final concept z-score has almost an opposite meaning according to P-value and represents the difference between the frequency in the given network and the mean frequency in the randomized network.

There are several random graph generation methods in the literature, but the important point is that the randomized graphs must have similar properties with the input network such as degree distribution of vertices and average path length. In this point of view, the most common random graph generation method used in network motif discovery is edge-switching method. It randomly selects two edges and exchanges the target vertices of each other so that the number of vertices and edges and the degree distribution of the vertices are preserved. There exist also other techniques such as 'Erdos–Re´nyi' (ER), 'Baraba´si–Albert' (BA) and 'Matching Method' for random graph generation.

According to above-mentioned concept and definitions, network motif discovery problem can be described as follows: Given a network G (represented as a graph), motif size k, frequency threshold F, number of random networks N and P-value P, discover all subgraphs in G that are consistent with the given thresholds.

The initial algorithms perform exhaustive search for discovering network motifs in the input networks. Milo not only defined the term "network motif" but also proposed an exhaustive search algorithm in his study [7]. However, exhaustive search has an exponential computational time with respect to motif or network size and it is infeasible to discover large size motifs by using only this approach.

Several strategies were developed to solve different parts of the network motif discovery problem. One of the problems is the exponential search space size with respect to motif or network size. A simple way of generating a size-k subgraph is to start with a size-2 (single edge) subgraph

and extend it with one vertex each time. A search tree, called as pattern growth tree, can represent this process. In this tree structure, each node represents a subgraph and its children nodes represent subgraphs that are extended by the parent subgraph by one vertex. In other words, the parent subgraph is actually a subgraph of its children subgraphs. There are several benefits of this tree structure if it is generated properly. First of all the pattern growth tree generated for size k-1 subgraphs in the previous executions can be used for searching size-k subgraphs which decreases the computational cost. Secondly, each subgraph can be generated only once to avoid redundant computation. Finally, downward closure property of frequency can be used to prune some parts of the tree for efficiency.

Another problem is the exponential computational time with respect to enumeration of all occurrences of a size-k subgraph on the input network for large k values. One method to handle this problem is processing only random sample subgraphs in the input network with using a probabilistic approach. This method is called as sampling strategy and used by a couple of algorithms in the literature. Two versions of sampling method, namely edge sampling and node sampling, are used in the popular network discovery systems.

4. INDEXING ON BIOLOGICAL NETWORKS

Biological networks hold the information on how molecules work collectively to perform key functions. Because of this, extracting knowledge from biological networks has been an important goal in computational biology. One way to do this is their comparative analysis that aims to identify the similarities between them by aligning them. However, alignment of biological networks is a computationally challenging problem. Existing methods often map the global and local network alignment problems to graph and subgraph isomorphism problems, respectively. These two problems however have no known polynomial time solutions. In the literature, two approaches exist to tackle this problem. First one either ensures optimality or at least a user supplied confidence in the optimality [20, 21]. The second one encompasses the heuristic approaches that do not

provide any optimality guarantees [22, 23]. Both of these approaches are computation intensive, and thus, they require a significant amount of running time and memory space.

Given a database that contains a large set of biological networks, a similarity query returns all database networks that have a higher alignment score with the query network compared to a user-specified similarity threshold. Rapid growth in the size of biological network databases coupled with the costly network alignment necessitates efficient methods for accessing and querying these databases. Exhaustively aligning each query network with all the networks in a large database one by one is neither practical nor feasible. Therefore, alternative techniques that reduce the number of network alignments are direly in need. In this context, database indexing has been successfully used for similarity queries on traditional databases, such as relational, multi-dimensional or time series databases. Biological network databases have inherent properties that distinguish them from such traditional databases.

4.1. Feature Based Indexing

Several indexing methods exist for similarity searches in graph databases. Majority of these methods can be classified as feature based indexing methods. These methods start by picking specific features of the networks for filtering purposes. Then, they pick corresponding features from the query network and match them with the features that exist in the networks of the database. They prune database networks that have low similarity value according to those matching. Finally, they do exact matching for the remaining database networks with the query network.

Due to noisy and incomplete characteristics of biological networks, approximate matching has become much more useful than exact matching for querying them. Substructure Index-based Approximate Graph Alignment (SAGA) [24] is a recent study that applies approximate matching on these networks. SAGA uses fragments of database networks as features and tries to combine them together to find larger matches.

The aim of SAGA method is to find subgraphs in the graph database that are similar to the query graph. Here, similarity allows vertex mismatches, vertex gaps (insertion and deletion) and graph structural differences. Vertex mismatches corresponds to two vertices representing different entities but having similar functionalities. Vertex gaps represent the vertices in one graph that cannot be matched to any vertex in the other graph. Graph structural differences allow for differences in vertex connectivity relationships. Each vertex in a graph has both a label and a unique id in SAGA. This unique id is used to establish a total order among the vertices. SAGA applies approximate matching in describing similarity between two graphs.

The main idea behind SAGA is to generate an index, namely FragmentIndex on small structures of the database graphs and then used it for matching fragments of the query graph with the fragments in the database graphs. After that, those matching fragments are used for larger matching. Given a query graph, SAGA enumerates the fragments in the query in a similar way to database graphs. It probes FragmentIndex for each query fragment. Then, it filters out unmatched index entries. At the end of filtering step, SAGA produces a set of small fragment hits. In the following step, SAGA assembles those smaller hits into bigger matches. Finally, it examines each candidate match and produces a set of real matches.

There exists also some other feature based indexing techniques in the literature. GraphGrep chooses paths as index feature [25]. gIndex uses frequent subgraphs for the same purpose [26]. Both methods apply exact subgraph matching which have limited usage on biological networks. Grafil [27] extends gIndex to support approximate matching for modeling similarity on biological networks.

4.2. Tree Based Indexing

Tree based indexing arranges the database networks hierarchically at different nodes in a tree. Thus, each node of a tree is a summary of a subset

of the database networks. For a given query network, the methods in this category align the query network to each node starting from the root node. Then, they progressively move down through the tree and filter out branches (i.e., subsets of networks) in the process.

One of the popular methods for graph indexing is Closure-Tree (CTree) [28]. Tree supports both subgraph and similarity queries. It organizes the networks in the database using a B-tree (namely C-tree) structure. Each leaf node represents a database network. Each internal node (called as graph closure) has structural information about its descendants in order to facilitate effective pruning. The internal nodes are actually hypothetical networks that are obtained by aligning the networks corresponding to their children nodes. An interesting property of the C-tree is the following: The score of the alignment of any query network with an internal node is at least as much as that with a leaf node rooted at that internal node. Following from this, given a query network, CTree algorithm starts aligning query to the root node. It then precedes to the children nodes. It prunes an entire subtree rooted at an internal node, if the alignment to that internal node has a score less than the given cutoff.

CTree supports both subgraph and similarity queries. CTree processes a subgraph query in two steps. In the first step, it traverses the C-tree and prunes nodes according to pseudo subgraph isomorphism and returns a candidate answer set. In the second step, it applies exact subgraph isomorphism on each candidate answer and returns the final answer set.

As exact similarity computing is expensive, CTree computes approximate graph similarity (or distance) using a heuristic graph mapping method. For this purpose, CTree contains a heuristic method, namely Neighbor Biased Mapping (NBM), in which the neighbors of a mapped vertex pair have higher chances to be mapped in the remaining iterations of the mapping process. CTree supports K-NN (K Nearest Neighbor) and range queries by using a priority queue that stores the nodes of C-tree.

There exist few algorithms that apply tree based indexing on graph databases. Berretti et. al. applied metric trees to attributed relational graph (ARG) databases for content-based image retrieval [29]. ARGs are clustered hierarchically according to their mutual distances and indexed by M-trees [30]. Queries are processed in a top-down manner by routing the query along the index tree where each node refers to a cluster. It uses triangular inequality to prune the unnecessary nodes. In the tree construction and at query time, the graph matching problem was solved by an extension of the A* algorithm that uses a lookahead strategy and a stopping threshold. Recently, this technique is used to model graphical representations of foreground and background scenes in videos [31]. The resulting graphs are clustered using the edit-distance metric, and similarity queries are answered using a multi-level index structure.

4.3. Reference Based Indexing

Reference based indexing summarizes the database networks using a small set of networks called references. The methods in this category align all database networks with all the references as a preprocessing step. Given a query network, instead of aligning it with the database networks, they align it with the references. Using these and precomputed alignments they filter a substantial subset of the database.

A recent indexing method for answering similarity queries on biological networks is Reference-based Indexing for Biological Network Queries (RINQ) [32]. RINQ uses a small set of networks as reference networks. Then, it aligns them with the database networks and stores all the alignment mappings and scores offline. After that, it aligns the given query network only with the reference networks. Finally, according to these alignment scores, it computes a lower and an upper bound for the similarity value between query network and each database network. By using these lower and upper bound values, RINQ prunes some of the database networks directly, selects some of the database networks as a part of result set without extra computation, and, applies exact matching for the remaining database networks.

RINQ has two major steps, namely index creation (which is done offline once) and query processing. The index creation step has two phases. In the first phase, RINQ creates a large set of candidate references from the database networks. Then, it selects a subset of these candidates as the actual reference networks according to their performances over a set of training queries.

The success of RINQ depends on the selection of reference networks. The candidate reference set in RINQ has the following properties:

* Each reference network has a small number of vertices so that the query network aligns with the reference network quickly. For this purpose, RINQ sets the size of each reference network as the size of largest query allowed.

* At least one reference network aligns well with any database network to find tight lower and upper bounds for any query network that aligns well with that reference network.

* Each reference network differs from the rest significantly to avoid redundant calculation in the further steps.

In the candidate reference set generation step, RINQ randomly selects a database network. Then, it randomly selects a vertex and extends it until it generates a subgraph having desired number of vertices. After that, it aligns that subgraph with the already generated candidate reference networks. If it is not similar to any of the candidate references, RINQ puts that subgraph into candidate reference network set. After this step, RINQ selects a subset of the candidate reference set as the actual reference set by using a set of training queries. RINQ selects actual reference set according to alignment scores with respect to training queries. It uses at most 100 networks in the actual reference set. It stores the alignment and score for each alignment between the query network and actual reference networks.

RINQ calculates an exact lower bound and an approximate upper bound value for the similarity between the query network and each database network using reference networks. Computing upper and lower bounds take much less time than aligning the query network with the database networks. In the final step, if the upper bound value for a database network is lower than the cutoff value, it is filtered (cannot be in the result set). If the lower bound value is greater than the cutoff value, it is directly put into result set. If the lower bound value is lower and upper bound value is greater than cutoff value, RINQ applies the costly network alignment algorithm for that database network.

5. SUMMARY

Recent developments in technology have led to large amount of real network generation in various fields of life. Subgraphs that occur in these networks with significantly higher frequency than those in randomized networks are called Network Motifs. Such subgraphs are the basic building structures of these networks. It is essential to extract information from these complex and large networks by discovering network motifs in them. By this way, we can reveal the hidden knowledge behind huge and complex datasets in knowledge-based systems. The frequency of a subgraph in a large graph shows a possibility of being a network motif. But, it has to be analyzed further to determine whether it has a functional role for that large graph. In this context, additional significance concepts are defined to determine whether a subgraph is a motif or not.

One way to find similarities efficiently for fast discovery of network motifs is to create index structures for similarity queries. There are three different approaches to indexing biological network databases in the literature. In the first approach, feature based indexing extracts sets of predefined features from all database networks. Thus, the methods in this category summarize the database networks with these features. Given a query network, they extract same or similar features for that query network. They compare those features of the query with the feature set of the entire database. Using this comparison, they alter some of the networks in the

database quickly. The second approach, tree based indexing, arranges the database networks hierarchically at different nodes in a tree. Thus, each node of a tree is a summary of a subset of the database networks. For a given query network, the methods in this category start aligning the query network to each node starting from the root node. Then, they progressively move down through the tree and alter out branches (i.e., subsets of networks) in the process. The third approach, reference based indexing, summarizes the database networks using a small set of networks called references. The methods in this category align all database networks with all the references as a preprocessing step. Given a query network, instead of aligning it with the database networks, they align it with the references. Using these and precomputed alignments they alter a substantial portion of the database. Finally, we conclude our analysis by showing a comparison between the three approaches.

REFERENCES

[1] D.A. Benson, I. Karsch-Mizrachi, D.J. Lipman, J. Ostell, B.A. Rapp, and D.L. Wheeler. GenBank. Nucleic Acids Research, 28(1):15–18, January 2000.

[2] A. Bairoch, B. Boeckmann, S. Ferro, and E. Gasteiger. Swiss-Prot: juggling between evolution and stability. Briefings in Bioinformatics, 1:39–55, 2004.

[3] Michael Baudis and Michael L. Cleary. Progenetix.net: an online repository for molecular cytogenetic aberration data. Bioinformatics, 17(12):1228–1229, 2001.

[4] H Ogata, S Goto, K Sato,WFujibuchi, H Bono, and M Kanehisa. KEGG: Kyoto Encyclopedia of Genes and Genomes. Nucleic Acids Research, 27(1):29–34, 1999.

[5] Peter Damaschke. Graph-Theoretic Concepts in Computer Science, volume 484/1991 of Lecture Notes in Computer Science, pages 72–78. Springer Berlin / Heidelberg, 1991.

[6] Wojciech Szpankowski Mehmet Koyuturk, Ananth Grama. An efficient algorithm for detecting frequent subgraphs in biological networks. In ISMB/ECCB (Supplement of Bioinformatics), pages 200–207, 2004.

[7] Milo R, Shen-Orr S, Itzkovitz S, et al. Network motifs: simple building blocks of complex networks. Science 2002; 298:824–27.

[8] Kashtan N, Alon U. Spontaneous evolution of modularity and network motifs. Proc Natl Acad Sci USA 2005;102:13773–78.

[9] Sole´ RV, Valverde S. Are network motifs the spandrels of cellular complexity? Trends Ecol Evol 2006;21:419–22.

[10] Bottani S, Vergassola M, Mazurie A. An evolutionary and functional assessment of regulatory network motifs. Genome Biol 2005;6:R35.

[11] Kashani ZRM, Ahrabian H, Elahi E, et al. Kavosh: a new algorithm for finding network motifs. BMC Bioinformatics 2009;10:318.[

[12] Grochow JA, Kellis M. Network motif discovery using sub-graph enumeration and symmetry-breaking. Res Comp Mol Biol 2007;4456:92–106.
[13] Kuramochi M, Karypis G. Frequent sub-graph discovery. In: Proceedings of IEEE International Conference on Data Mining 2001, San Jose, California, 313–20.
[16] Schreiber F, Schwobbermeyer H. Frequency concepts and pattern detection for the analysis of motifs in networks. LectNotes Comput Sci 2005;3737:89–104.
[17] Babai, L., Luks, E. M., 1983. Canonical labeling of graphs. In: STOC. pp 171-183.
[18] Kuramochi M, Karypis G. Finding frequent patterns in a large sparse graph. Data Mining and Knowledge Discovery 2005, 11(3), 243–271.
[19] Baskerville, K., Paczuski, M., 2006. Subgraph ensembles and motif discovery using an alternative heuristic for graph isomorphism. Physical Review E 74 (5), 51903.
[20] R. Y. Pinter, O. Rokhlenko, E. Yeger-Lotem, and M. Ziv-Ukelson, Alignment of metabolic pathways, Bioinformatics. 21(16), 3401-3408 (Aug, 2005). doi: 10.1093/bioinformatics/bti554.
[21] T. Shlomi, D. Segal, E. Ruppin, and R. Sharan, QPath: a method for querying pathways in a protein-protein interaction network, BMC Bioinformatics. 7, 199 (2006). doi: 10.1186/1471-2105-7-199.
[22] F. Ay and T. Kahveci. SubMAP: Aligning Metabolic Pathways with Subnetwork Mappings. In RECOMB, pp. 15-30 (2010)
[23] F. Ay, T. Kahveci, and V. de Crecy-Lagard, A fast and accurate algorithm for comparative analysis of metabolic pathways, J. Bioinformatics and Computational Biology. 7(3), 389-428 (2009).
[24] Y. Tian, R. C. McEachin, C. Santos, D. J. States, and J. M. Patel, SAGA: a subgraph matching tool for biological graphs, Bioinformatics. 23(2), 232-239 (2007). doi: 10.1093/bioinformatics/btl571.
[25] R. Giugno and D. Shasha. GraphGrep: A fast and universal method for querying graphs. In Proc. 16th Int Pattern Recognition Conf, vol. 2, pp. 112-115 (2002). doi: 10.1109/ICPR.2002.1048250.
[26] X. Yan, P. S. Yu, and J. Han. Graph indexing: a frequent structure-based approach. In SIGMOD, pp. 335-346, ACM, New York, NY, USA (2004). ISBN 1-58113-859-8. doi: http://doi.acm.org/10.1145/1007568.1007607.
[27] X. Yan, P. S. Yu, and J. Han. Substructure similarity search in graph databases. In SIGMOD Conference, pp. 766-777 (2005).
[28] H. He and A. K. Singh, Closure-Tree: An index structure for graph queries, International Conference on Data Engineering. 0, 38 (2006).
[29] S. Berretti, A. D. Bimbo, and E. Vicario, Efficient matching and indexing of graph models in content-based retrieval, IEEE Trans. Pattern Anal. Mach. Intell. 23(10), 1089-1105 (2001).
[30] P. Ciaccia, M. Patella, and P. Zezula. M-tree: An efficient access method for similarity search in metric spaces. In VLDB'97, Proceedings of 23rd International Conference on Very Large Data Bases, pp. 426-435, Morgan Kaufmann (1997). ISBN 1-55860-470-7.
[31] J. Lee, J.-H. Oh, and S. Hwang. Strg-index: Spatio-temporal region graph indexing for large video databases. In SIGMOD Conference, pp. 718-729 (2005).
[32] G. Gulsoy and T. Kahveci, RINQ: Reference-based indexing for network queries, Bioinformatics [ISMB/ECCB]. 27(13), 149{158 (2011).