OXFORD

Data and text mining

A Cytoscape app for motif enumeration with ISMAGS

Thomas Van Parys[1,2,3], Ine Melckenbeeck[4], Maarten Houbraken[4], Pieter Audenaert[3,4,*], Didier Colle[3,4], Mario Pickavet[3,4], Piet Demeester[3,4] and Yves Van de Peer[1,2,3,5,*]

[1]Department of Plant Systems Biology, VIB, Ghent 9052, Belgium, [2]Department of Plant Biotechnology and Bioinformatics, Ghent University, Ghent 9052, Belgium, [3]Bioinformatics Institute Ghent, Ghent University, Ghent, Belgium, [4]Department of Information Technology (INTEC), Ghent University – iMinds, Ghent 9052, Belgium and [5]Genomics Research Institute, University of Pretoria, Pretoria, South Africa

*To whom correspondence should be addressed.

Abstract

Summary: We present a Cytoscape app for the ISMAGS algorithm, which can enumerate all instances of a motif in a graph, making optimal use of the motif's symmetries to make the search more efficient. The Cytoscape app provides a handy interface for this algorithm, which allows more efficient network analysis.

Availability and Implementation: The Cytoscape app for ISMAGS can be freely downloaded from the Cytoscape App store http://apps.cytoscape.org/apps/ismags. Source code and documentation for ISMAGS are available at https://github.com/biointec/ismags. Source code and documentation for the Cytoscape app are available at https://gitlab.psb.ugent.be/thpar/ISMAGS_Cytoscape.

Contacts: Pieter.Audenaert@intec.ugent.be or Yves.VanDePeer@psb.vib-ugent.be

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Network motifs are small subgraphs which are overrepresented in an explored graph, compared to what would be expected statistically. The function of a graph is reflected in which motifs occur more often and which nodes make up these motifs (Milo *et al.*, 2002). In particular, motifs in protein–protein interaction networks and transcription regulatory networks correspond to cell functions (Grochow and Kellis, 2007). Therefore, these motifs are a powerful tool to analyze graphs. Moreover, the motifs have an ordered structure which can be divided in clusters, giving more detailed information about the nodes' function compared to classical clustering (Benson *et al.*, 2016). Many algorithms to enumerate them exist (Grochow and Kellis, 2007; Omidi *et al.*, 2009; Wernicke and Rasche, 2006, etc.) including a GPU-based method (Lin *et al.* (2015)). In this work, we focus on an application which can be run on a standard, off-the-shelf desktop or laptop without specialized hardware. ISMAGS (Houbraken *et al.*, 2014) is an efficient Java

implementation for the motif enumeration problem running on stock hardware.

When we talk about enumerating all instances of a certain motif, we mean finding every subgraph that is isomorphic to the given motif within an explored graph. These subgraphs will be called *motif instances*. Motifs are not *induced* subgraphs, which means nodes in motif instances may be connected by edges not in the original motif. As not all edges are known in biological networks, this is a useful approach. Between any pair of nodes in a found motif may, indeed, be unobserved edges. If any of these edges are found, the motif instance will still be valid.

Here, we first describe ISMAGS shortly, then we present the features of a new ISMAGS Cytoscape app. This app has all features of ISMAGS, is completely compatible with Cytoscape networks and accepts several formats for input of motifs. The visual interface makes the motif search easier, and it allows the user to efficiently use the results and gain insight in the connections between different motifs in a straightforward way.

2 ISMAGS: optimized motif enumeration

ISMAGS (the Index-based Subgraph Matching Algorithm with General Symmetries) is an efficient algorithm to enumerate all instances of a motif within a graph. It makes optimal use of a motif's symmetries to eliminate as much of the search space as quickly as possible.

Before the search, the motif's symmetry is analyzed. First, its automorphisms are found and the nodes are divided into automorphism orbits. These automorphisms are permutations of the nodes which leave the edge set unchanged, and orbits are the sets of nodes which can be mapped onto each other by these automorphisms. From the automorphisms, stabilizers are derived. These are sets of automorphisms which do not change the first n nodes of the motif, one set for each additional node. This means the first stabilizer is the set of automorphisms that keep one node unchanged. The permutations in the second set keep that node and another one unchanged and the last set only contains the identity permutation. Each stabilizer is a subset of the previous one, forming a stabilizer chain. In each stabilizer in a chain, the nodes which are not yet stabilized can still be mapped on a number of other nodes. The set of nodes a certain node can still be mapped to, just before it is stabilized, is called that node's coset representative. These coset representatives are used to impose symmetry-breaking constraints on the nodes of a motif instance. These constraints are imposed as an ordering of each node's unique index. This eliminates the possibility of finding different motif instances with the same graph nodes, which decreases the running time of the algorithm. More detailed information can be found in Houbraken *et al.* (2014).

When the motif's symmetries are known, the search can begin. The motif is built by adding nodes to a motif instance, mapping graph nodes to motif nodes. Each time a new node needs to be mapped, the motif node to which the least graph nodes can be mapped is chosen to cut down the search tree as much as possible, as quickly as possible. Specialized data structures, such as checklists for mapped nodes and iterators for each motif node's candidate mappings, ensure the algorithm is iterative. For more information, see Demeyer *et al.* (2013).

ISMAGS can handle edges of different types, both directed and undirected, within one graph. ISMAGS is a very flexible tool and accepts many different types of networks.

3 Implementation

When using Cytoscape with the ISMAGS app, a new tab appears in the control panel, named ISMAGS. Here, the network in which the motif must be searched and the motif itself must be selected from two drop-down menus. Any opened Cytoscape network may be selected for either of them. ISMAGS is fully compatible with Cytoscape networks, both for motif and search network. Additionally, motifs can also be defined by their string form, as is done in ISMAGS following an encoding scheme introduced in Demeyer *et al.* (2013). When a network is selected in the motif selection menu, its string form appears in the text box below. Changes in the string form will show in the network and vice versa (Fig. 1).

Upon selecting the motif network, the different motif edge types appear in the edge table in the left panel. This table lets the user edit how the symbols from the motif string form map to the interactions in the motif network. An edge type can be set to be undirected, which will ignore its direction while searching for motif instances. Also, the color of the edge types can be changed.
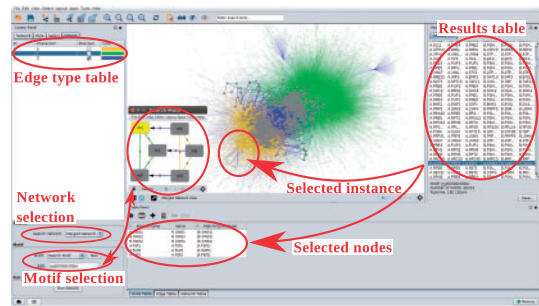

Fig. 1. The ISMAGS-Cytoscape interface. The interface features are marked in red

Table 1. Runtime and number of found instances of n-star motifs in a random graph with 100 nodes and 1000 edges

Number of nodes	Number of instances	runtime (ms)	Runtime per instance (μs)
2	1000	0.6456	0.6456
3	19 709	10.9004	0.5531
4	127 690	71.5114	0.5600
5	611 818	621.3434	1.0156
6	2 310 562	2924.0342	1.2655
7	7 153 491	11616.0336	1.6238
8	18 637 409	25689.6910	1.3784

Search results are shown in the Results Panel on the right. The panel shows a brief summary of the run: the motif, the number of found instances and the runtime of the algorithm. The results table lists every found instance of the motif. Selecting one or more instances will highlight the corresponding nodes in the network. Edges and nodes not part of any motif will be faded so they do not obscure the view of the motif instances. This allows for easy visual inspection of the motifs' position within the network, which may show patterns that are not obvious when using text results. With the built-in features of Cytoscape, the resulting subnetwork can then be exported and explored further, allowing for efficient subsequent analysis without needing to switch programs. The result set can also be saved in a format compatible with ISMAGS.

4 Usage

Motifs are used in a wide variety of fields where network analysis is useful. Our ISMAGS-application allows researchers to find significant motifs in reasonably large networks, and thus support and facilitate other highly-relevant research. In computational biology, motifs may correspond to cellular transcription machinery or cell-cycle regulation (Grochow and Kellis, 2007) and may be used to predict regulatory modules (Zhang *et al.*, 2008)). Also, motifs can be clustered intelligently, yielding new information about the nodes' function (Benson *et al.*, 2016).

4.1 Runtime

To assess the application's running time, a number of motifs were searched in random graphs. As extensive runtime benchmarks of the ISMAGS algorithm can be found in Houbraken *et al.* (2014) we focus here on benchmarking the Cytoscape app itself. All of these tests were performed on a computer with 4 Intel® Core™ i5-3230M CPU @ 2.60 GHz CPUs and 7.7 GiB internal memory.

Table 2. Runtime and number of found instances of *n*-clique motifs in a random graph with 100 nodes and 1000 edges

Number of nodes	Number of instances	runtime (ms)	Runtime per instance (μs)
2	1000	0.6456	0.6456
3	1325	3.7654	2.8418
4	236	4.948	20.9661
5	4	3.4232	855.8000
6	0	4.005	

Table 3. Runtime and number of found instances of 6-star motifs in a random graph with 100 nodes and *m* edges

Number of edges	Number of instances	runtime (ms)	Runtime per instance (μs)
1000	2 310 562	2924.0342	1.2655
1200	5 925 211	5832.298	0.9843
1400	13 015 764	14 522.382	1.1158
1600	26 786 482	40 077.673	1.4962
1800	46 664 113	78 907.906	1.6910

Table 4. Runtime and number of found instances of 6-star motifs in a random graph with *n* nodes and 10*n* edges

Number of nodes	Number of edges	Number of instances	runtime (ms)	Runtime per instance (μs)
100	1000	2 310 562	2924.0342	1.2655
200	2000	5 317 675	5159.928	0.9703
300	3000	7 539 912	7373.249	0.9779
400	4000	10 114 976	8973.533	0.8872
500	5000	12 482 687	11 683.605	0.9360
600	6000	15 667 131	15 503.34	0.9895
700	7000	19 308 241	18 257.883	0.9456
800	8000	20 678 893	20 471.952	0.9900
900	9000	24 202 997	24 574.815	1.0154

A first test searches motifs in a random graph with 100 nodes and 1000 edges. These motifs were all undirected *n*-node star motifs, i.e. motifs consisting of one node connected with all *n* – 1 other nodes, which do not need to have any other edges amongst themselves. The result can be seen in Table 1.

While the running time rises quickly, so does the number of found instances. Motifs that appear more rarely, like clique motifs, are found more quickly. Cliques are motifs in which all nodes are connected. ISMAGS's performance for these motifs can be seen in Table 2.

Finally, in Tables 3 and 4, the running time of ISMAGS in networks of different sizes can be seen. In Table 3, a star motif was searched in networks with a fixed number of nodes but variable number of edges; in Table 4, the same motif was searched in networks with a variable number of nodes but fixed proportion of edges to nodes. Cytoscape save files for all of these tests can be found in the supplementary materials.

5 Conclusion

We have presented the ISMAGS app for Cytoscape, which can quickly and efficiently list all instances of a given motif in a graph. The clear interface offers all available functions in a straightforward layout. The visual output helps understanding the search results' meaning. The Cytoscape integration allows the use of a variety of networks and efficient export and use of the results. The combination of these features makes the entire motif searching procedure intuitive to apply, efficient and easy to interpret, supporting new research.

Funding

References

Benson,A.R. *et al*. (2016) Higher-order organization of complex networks. *Science*, **353**, 163–166.

Demeyer,S. *et al*. (2013) The index-based subgraph matching algorithm (ISMA): fast subgraph enumeration in large networks using optimized search trees. *PloS One*, **8**, e61183.

Grochow,J.A. and Kellis,M. (2007) *Research in Computational Molecular Biology: 11th Annual International Conference, RECOMB 2007*, Oakland, CA, USA*, April 21–25, 2007. Proceedings*, chapter Network Mo, pp. 92–106. Springer, Berlin, Heidelberg.

Houbraken,M. *et al*. (2014) The Index-based Subgraph Matching Algorithm with General Symmetries (ISMAGS): exploiting symmetry for faster subgraph enumeration. *PloS One*, **9**, e97896.

Lin,W. *et al*. (2015). Network motif discovery: A GPU approach. In: *2015 IEEE 31st International Conference on Data Engineering*, pp. 831–842.

Milo,R. *et al*. (2002) Network motifs: simple building blocks of complex networks. *Science*, **298**, 824–827.

Omidi,S. *et al*. (2009) MODA: an efficient algorithm for network motif. *Genes Genet. Syst*., **84**, 385–395.

Wernicke,S. and Rasche,F. (2006) FANMOD: a tool for fast network motif detection. *Bioinformatics*, **22**, 1152–1153.

Zhang,Y. *et al*. (2008) Network motif-based identification of transcription factor-target gene relationships by integrating multi-source biological data. *BMC Bioinformatics*, **9**, 1–18.