



SOFTWARE TOOL ARTICLE

REVISED NetMatchStar: an enhanced Cytoscape network querying app [version 2; referees: 2 approved]

Fabio Rinnone¹, Giovanni Micale¹, Vincenzo Bonnici², Gary D. Bader³, Dennis Shasha⁴, Alfredo Ferro⁵, Alfredo Pulvirenti⁵, Rosalba Giugno⁵

¹Department of Math and Computer Science, University of Catania, Catania, 95125, Italy

²Department of Computer Science, University of Verona, Verona, 37134, Italy

³The Donnelly Centre, University of Toronto, Toronto, ON, M5S 3E1, Canada

⁴Department of Computer Science, Courant Institute of Mathematical Sciences, New York University, New York, NY, 10012, USA

⁵Department of Clinical and Experimental Medicine, University of Catania, Catania, 95125, Italy

v2 First published: 05 Aug 2015, 4:479 (doi: [10.12688/f1000research.6656.1](https://doi.org/10.12688/f1000research.6656.1))
 Latest published: 03 Nov 2015, 4:479 (doi: [10.12688/f1000research.6656.2](https://doi.org/10.12688/f1000research.6656.2))

Abstract

We present NetMatchStar, a Cytoscape app to find all the occurrences of a query graph in a network and check for its significance as a motif with respect to seven different random models. The query can be uploaded or built from scratch using Cytoscape facilities. The app significantly enhances the previous NetMatch in style, performance and functionality. Notably NetMatchStar allows queries with wildcards.



This article is included in the [Cytoscape apps](#) channel.

Open Peer Review

Referee Status:

Invited Referees

1 2

REVISED

version 2

published
03 Nov 2015

version 1

published
05 Aug 2015



1 Xifeng Yan, University of California, Santa Barbara USA

2 Shailay Dogra, Vishuo BioMedical Pte Ltd Singapore

Discuss this article

Comments (0)

Corresponding author: Rosalba Giugno (giugno@dmi.unict.it)

How to cite this article: Rinnone F, Micale G, Bonnici V *et al.* **NetMatchStar: an enhanced Cytoscape network querying app [version 2; referees: 2 approved]** *F1000Research* 2015, 4:479 (doi: [10.12688/f1000research.6656.2](https://doi.org/10.12688/f1000research.6656.2))

Copyright: © 2015 Rinnone F *et al.* This is an open access article distributed under the terms of the [Creative Commons Attribution Licence](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Grant information: RG, AP and AF have been founded by Programma Operativo Fondo Europeo per lo Sviluppo Regionale (PO-FESR 2007-2013), Linea di intervento 4.1.1.2. Grant number: CUP G23F11000840004.

Competing interests: No competing interests were disclosed.

First published: 05 Aug 2015, 4:479 (doi: [10.12688/f1000research.6656.1](https://doi.org/10.12688/f1000research.6656.1))

REVISED Amendments from Version 1

We addressed the referee comments. We modified last paragraph in the introduction and the first paragraph in the evaluation statistic section.

See referee reports

Introduction

Biological networks such as protein-protein interaction, transcription regulatory, gene regulatory, and metabolic networks are often referred to as complex systems¹. The term complex relates to the existence of non-trivial substructures contained within them. The study of complex systems involves the analysis of the way in which their elements interact rather than only their individual roles. Computationally, such a study entails the ability to query networks to find specific patterns of interactions.

Possible queries might include the identification of positive and negative autoregulation, coherent and incoherent feed forward loops, single-input modules and dense overlapping regulons² in a given target network N . Sub-networks that occur surprisingly often in a network may be preferred by evolution. For that reason, NetMatchStar offers the ability to compute a p-value against null models from seven distinct randomizing methods and suggests the one that shares the network properties of N in terms of degree distribution, cluster coefficient and assortativity.

The availability of computational tools for the analysis of biological networks has been helpful in providing novel biological insights on the function of many previously uncharacterized proteins. Several different methods have been developed for this purpose: (i) Network Motif finding³⁻¹², network querying¹³⁻¹⁵ and network alignment¹⁶⁻²¹ algorithms.

Most of the approaches dealing with this kind of graph analysis entail subgraph matching. Such a problem has been widely studied and several methods and systems have been proposed. The approaches can be categorized according to the methodology they use. The first category is the tree search based algorithm. Those methods look for a solution of the problem in a state space by making use of a depth-first approach. Algorithms using such approach include Ullmann²², VF2²³ and the recently introduced RI²⁴. The second category consists of algorithms using Constrained Programming techniques. Such methods aim at filter pairs of nodes which will not be in a matching solution. Many algorithms exploit such approaches²⁵⁻²⁷. The last category uses a database approach by exploiting the virtues of indexing²⁸⁻³¹. Such algorithms extract a set of features which define an index of the query that will be used for searching through the target one. The goal is to identify candidate subgraphs in the target one which are possibly isomorphic to the query. NetMatchStar works on Cytoscape 3.2+ and is based on the NetMatch software in¹³. It deals with both exact queries and approximate ones, in which wildcards are used to match unspecified number of elements.

NetMatchStar integrates the RI algorithm proposed for biological real networks which outperforms other existing algorithms²⁴.

RI uses a search strategy based on the topology of the query to effectively filter the space of solutions. We refer to NetMatchStar web page for use cases. For illustration purposes, NetMatchStar has been tested on a biological dataset^{24,32} and an overview of its performance concludes the paper.

Methods

A graph G is a pair (V, E) , where V is the set of nodes and $E \subseteq (V \times V)$ is the set of edges.

Using a graph Q to query a target network graph N means to perform a subgraph isomorphism, which entails finding an injective function that maps each node of Q to a unique node of N such that nodes and edges labels are preserved. Assessing the statistical significance of Q implies a simulation process, where first a set R of r random graphs are generated according to a specific model. Then the number of occurrences of Q in each random graph is counted and a p-value is computed which is defined as the fraction of the r graphs where Q occurs at least as often as in N . The lower the p-value is, the more significant Q is as a motif. The significance of Q can also be evaluated through the z-score, which is defined as the difference between the number of occurrences of Q in N and the average number of occurrences of Q in the r random graphs, divided by the standard deviation of the frequencies of Q in R . A strongly positive value of the z-score means that Q is significant as a motif.

Exact querying

A simple enumeration algorithm to find Q in N generates all possible maps between the nodes of the two graphs and checks whether any generated map is a subgraph isomorphism. The common aim of existing algorithms is to discover unsuccessful mappings as early as possible and to filter them away²². NetMatchStar uses the algorithm RI proposed in²⁴, whose efficiency is mainly due to the choice of a search strategy, i.e. the ordering with which query nodes are mapped. For example, a variable ordering may begin with a query node having the highest degree or having the most uncommon label in the target graph. The variable ordering of RI is based only on the query graph topology. Roughly, the chosen order creates constraints as early as possible in the matching phase. The nodes having high valence and that are highly connected with nodes previously present in the ordering tend to come early in the variable-ordering. The aim of RI is to avoid costly pruning techniques by finding a static search strategy such that the number of constraints that are verifiable from a partial solution are maximized.

Approximate querying

Approximate queries are graphs containing wildcard structures. They may contain nodes and edges which can match any value of node or edge labels in the network and approximate paths constrained in length to be less than or greater than m , where m is a positive integer. NetMatchStar first matches all the specified subparts of the queries exactly and then joins the matches by network traversal. The network traversal phase checks that all traversed paths satisfy the query path constraints.

Random model generation

In NetMatchStar, the user can choose among seven different generative models to compute the statistical significance of a motif. In all cases, except for the shuffling model, the simulation starts with

the generation of a network with $|V|$ nodes having the same labels as the target network N and no edges. Then, new edges between existing nodes are added until we obtain a network with $|V|$ nodes and $|E|$ edges, just like N . In the following, we briefly describe each random model.

Shuffling model. In the shuffling model³³ an existing network is “rewired” by repeatedly swapping the destinations of two randomly chosen edges, where possible. The result is a graph with the same degree distribution of the original network.

Erdos-Renyi model. The Erdos-Renyi (ER) model³⁴ corresponds to a graph where two nodes connect each other randomly and independently. There are two variants of the ER model. In the $G(|V|, |E|)$ model the algorithm randomly creates a network uniformly over all networks that have $|V|$ nodes and $|E|$ edges. In the $G(|V|, p)$ model, edges between nodes are independently created with a user-defined probability p . NetMatchStar implements the $G(|V|, p)$ variant of the ER model.

Watts-Strogatz model. The Watts–Strogatz model³⁵ produces graphs characterized by the small-world property, where most nodes can be reached from every other by a small number of hops, when there is no direct link between them. The model works in two phases. In the first one a lattice of $|V|$ nodes is created where each edge is connected to d neighbors on its left and d neighbors on its right. Then, edges are randomly shuffled with rewiring probability β . Low values of β produce a quasi-regular graphs, where nodes have approximately the same degree, while high values of β produce networks which are very close to the ER model.

Barabasi-Albert model. Also known as the preferential attachment model, this model³⁶ creates graphs where the more connected a node is, the more likely it creates new links. Graphs generated with BA model are scale-free, meaning that the degree distribution follows a power law, with a few high-degree nodes and many low-degree nodes. The BA model starts with the creation of a complete initial seed network of k nodes. The remaining $|V| - k$ nodes are added one at a time. Each new node is attached to d existing nodes, such that the probability of selecting an existing node u is proportional to the degree of u .

Geometric model. The geometric model³⁷ describes graphs in which the information about the location of nodes in the space determines the topology and might be useful to represent spatially oriented networks (e.g. transportation and neuronal networks). In the geometric model each node is represented as a point in a d -space. An edge between two nodes exists if the distance between corresponding points is within a threshold r .

Forest-Fire model. In the Forest-Fire (FF) model³⁸, a new node v attaches to the network by iteratively exploring existing edges starting from one or more anchor nodes, called ambassadors, which are chosen randomly. At each step of the exploration, v creates out-links with newly discovered nodes with a forward probability p and in-links with a backward probability r , and continues exploration from those nodes. The FF model describes time-evolving networks where the number of edges grows super-linearly in the number of nodes and the distance between nodes shrinks as new nodes arrives.

Duplication model. In the duplication model³⁹ the duplication of the information is considered as a dominant evolutionary force for the growth of a network, such as in many biological networks. At each step of the duplication model a random node u is selected. Then, a new node v is created and connected to neighboring nodes of u with probability p . The lower is p , the more divergent is v as a copy of u .

Implementation

The NetMatchStar Cytoscape App has been developed in Java 7 on top of the Cytoscape 3.2 API. The software is composed by a core module, which implements basic algorithms and data structures, plus a user interface module that integrates the analyses into the Cytoscape interface. The core module provides data representations, graph analysis (i.e. graph matching and motif searching) and two different types of attribute comparator that differentiate in exact and approximate comparison. The CyNetworks are converted into graph structures to optimize the graph traversal procedures. The user interface is designed by following the Model-View-Controller architectural pattern. The Model component adds up result data representations to the functionality provided by the software’s core module. The View component implements the graphical panels of the interface. The main panel of the app adds up, as a further tab, to the Control Panel of the Cytoscape interface. This integrates the graphical panels where the user can select the networks to be processed, the parameters of the analysis, and the results. The Control component ensures the communication between the Model and the View by implementing the set of tasks performed by NetMatchStar. This component is developed by following the Cytoscape 3.1 app guidelines, such that every task is implemented as a Cytoscape Task Java class.

Operation

The main frame of NetMatchStar contains three tabbed panels:

- “Matching” panel (Figure 1), to specify the target and the query graphs and run the matching task;
- “Significance” panel (Figure 2), for the statistical significance of the query as a motif according to a specific random model;
- “Motif library” panel (Figure 3), which contains a set of pre-defined queries for the matching task.

In the following subsections, we will describe all the required steps for the matching and motif verification of a query graph in a target network.

Loading input data

Query and network graphs can be uploaded in NetMatch-Star, by clicking on the folder icon in the toolbar of “Matching” panel (Figure 1). Each uploaded network will be added to the Network list of Cytoscape. In the drop-down lists of “Network Properties” and “Query Properties” section, the user can select one of the uploaded networks as a query or target network for the matching and statistical significance tasks. Likewise, the user may upload node and edge labels as Cytoscape attributes and link them to the nodes and edges of the target network and query graph.

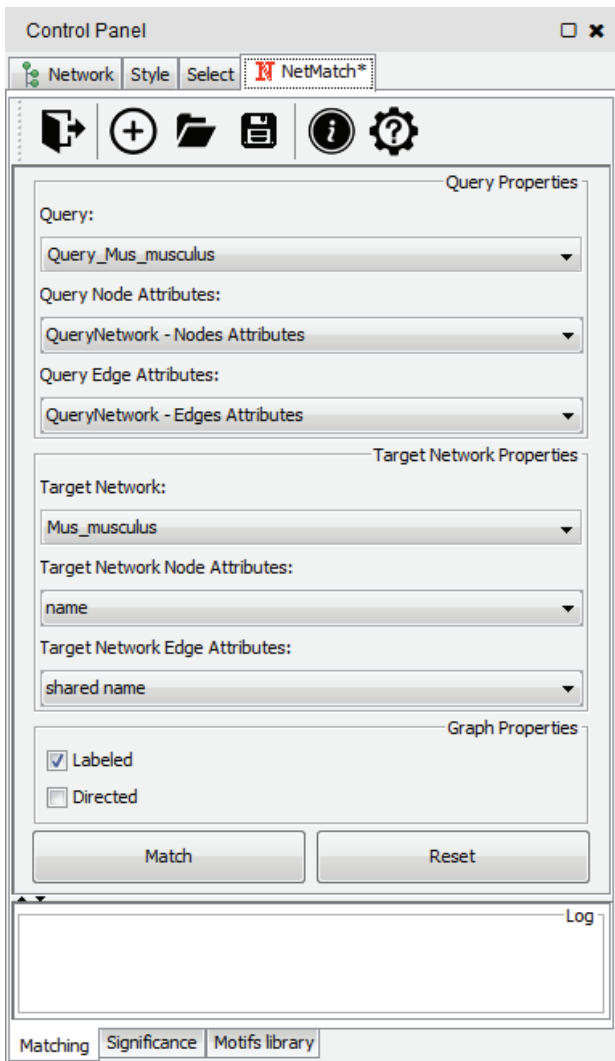


Figure 1. “Matching” panel in NetMatchStar. In this example, the network of Figure 4 has been provided as query, while the *Mus musculus* network provided in 24 has been chosen as target graph.

Drawing queries

Instead of loading an existing network, the user can create a query from scratch or by starting from a pre-defined set of queries.

To create a new query, the user must click on the “plus” icon of “Matching panel” (Figure 1). A new panel for the creation of a new network will be opened (Figure 4). A right click on the panel will open the standard Cytoscape menu to add, edit or remove elements of the graph. Such a menu also includes the “NetMatchStar” menu item, which lets the user change the label of a node or edge and set a path between two nodes. By default, newly added nodes and edges will be labeled with the wildcard “?”, corresponding to a node or a direct link between nodes with unspecified label. Any other character will be associated to a specific label. Paths between two nodes i and j are defined as special attributes for the edge (i, j) .

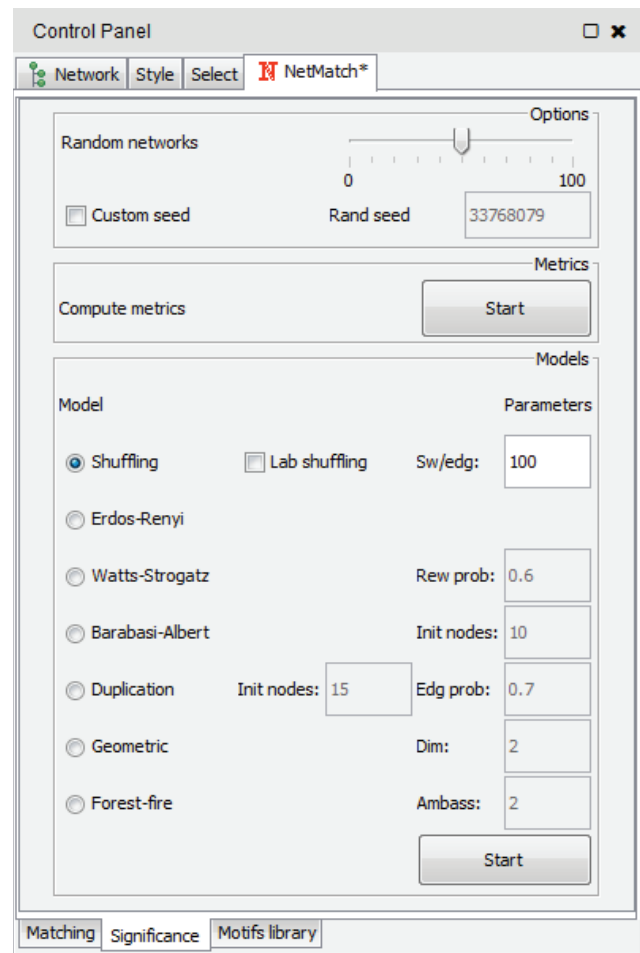


Figure 2. “Significance” panel in NetMatchStar.

The length of a path is specified by an expression of the form $aopb$, where a and b are two integers (or the wildcard “?”) and op is one of $<$, \leq , \geq , $>$, $=$. The “?” character is used to leave the minimum or maximum length of the path unspecified. For instance, the expression “? \leq 2” means that the corresponding path must have at most length 2, while “? $>$ 3” corresponds to a path of length greater than 3. A query with a “?” character in at least a node and/or edge is an approximate query for NetMatchStar.

By clicking on “Save” button on panel, the user can store the query graph created from scratch on disk as text files in a .SIF format with nodes and edges attribute files with extensions respectively .NA and .EA.

The pre-defined set of queries includes small topologies which have been identified as motifs in many real networks², such as feed-forward loops, diamonds, single-input modules and dense overlapping regulons. Figure 3 shows all the pre-defined queries that can be selected from the “Motifs library” tabbed panel. They are drawn as directed graphs but can be used to query both directed and undirected networks. By clicking on one of these topologies, the

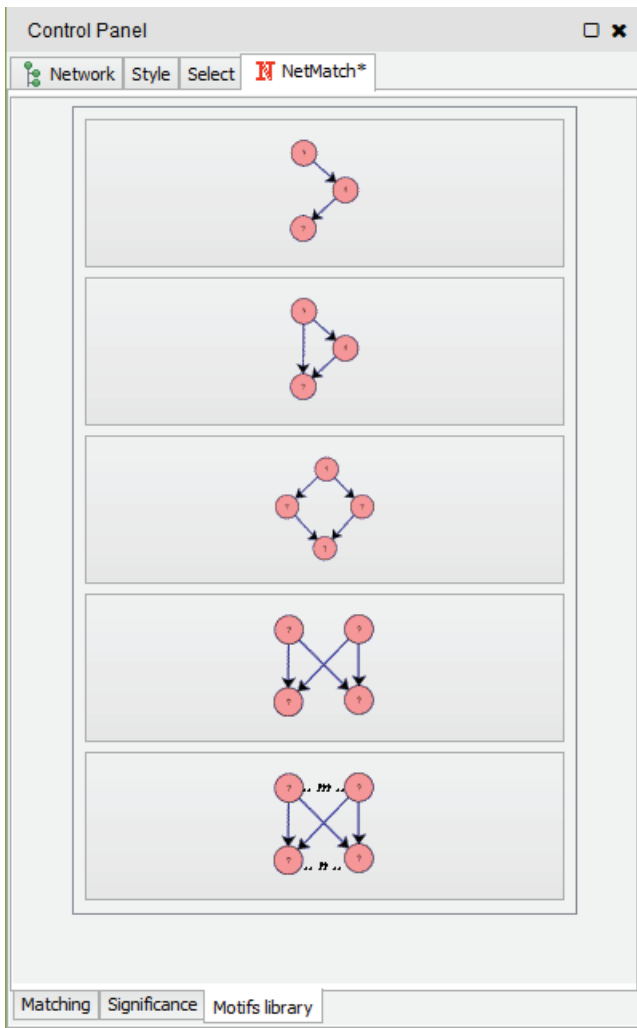


Figure 3. “Motif library” panel in NetMatchStar.

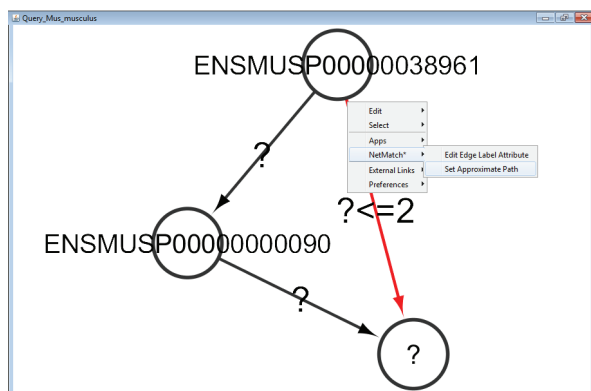


Figure 4. Panel for the creation of a query network in NetMatchStar. In this example, an approximate query with 3 nodes and 3 edges has been created, where 2 nodes have a specific label and one edge represents an approximate path of length at most 2 ($? \leq 2$). The remaining elements of the graph have an unspecified label (?). By selecting an edge and right-clicking, a menu will be shown for changing its label or set the approximate path.

user can visualize the query and modify it, as previously described, i.e. adding new nodes/edges, changing node/edge labels and setting paths between nodes. Modifying the pre-defined query does not change the original “library” entry, but only a copy of it.

Evaluating the Statistical Significance of motifs

The “Significance” panel (Figure 2) contains all the parameters for the evaluation of the statistical significance of a motif subnetwork. It consists of three subpanels. In the top subpanel the user can choose the number of random graphs to generate for the statistical test (between 0 and 100) and the seed for generating pseudorandom numbers. In the middle subpanel the user can compute a set of metrics for the target graph and sample random graphs, one for each model. Metrics include the average degree, the average clustering coefficient and the assortativity index. At the end of the computation, the resulting values are shown in a separate window. Usually, values of these metrics coherent with the one of the input network can suggest to the user which random model best describes the features of the input network.

The bottom subpanel let the user choose a random model and set its parameters (if any). In “Shuffling” model, “Lab shuffling” option can be selected for enabling shuffling also on node and edge labels (if present), while “sw/edg” denotes the number of successful swaps per edges. The “Erdos-Renyi” model has no parameters. In “Watts-Strogatz”, “Rew prob” is the probability of rewiring β . The “Barabasi-Albert” model defines “Init nodes”, the number of initial nodes in the complete seed network. The “Duplication” model has two parameters: “Init nodes”, the number of nodes in the initial seed network, and “Edg prob”, the edge duplication probability. In the “Geometric” model, parameter “Dim” denotes the dimension of the space where points are placed. Finally, “Forest-fire” contains parameter “Ambass”, that is the number of ambassadors nodes. For each model, all the remaining parameters are estimated based on the number of nodes and edges of the target network.

Managing results

Once a target network and a query has been provided in the “Matching” panel (Figure 1), the user can either look for all occurrences of the query within the input graph or check if the query is a motif or not.

In the first case, the user must click on the “Match” button in the “Matching” panel (Figure 1). Once the matching task has been completed, a table with all the occurrences of the query in the target will be shown as a tabbed panel in the “Result Panel” of Cytoscape (Figure 5) and the input graph will be visualized. For each occurrence, NetMatch-Star reports its nodes and an image depicting its topology. By selecting a row in the table, the user can visualize the corresponding occurrence in the target network. If the option “Create a new child network” is disabled, nodes of the occurrence will be highlighted in yellow within the input network, otherwise the occurrence will be visualized in a separate window. By clicking on “Save” button on result panel, the user can store the results as text file.

Recalling that the nodes of the network are not uniquely labeled and thus the query may have different matches, to check if a query is a motif, the user must click on one of the “Start” buttons of the “Significance” panel (Figure 2), depending on the random model that has

Match number	Occurrences	Nodes	Image
1	1	ENSMUSP00000096193, ENSMUSP00000038961, ENSMUSP00000000090, ENSMUSP00000080997	
2	1	ENSMUSP00000119603, ENSMUSP00000038961, ENSMUSP00000000090, ENSMUSP00000080994	
3	1	ENSMUSP00000115419, ENSMUSP00000038961, ENSMUSP00000000090, ENSMUSP00000080994	
4	1	ENSMUSP00000038961, ENSMUSP00000026561, ENSMUSP00000000090, ENSMUSP00000080997	
5	1	ENSMUSP00000038961, ENSMUSP0000015934, ENSMUSP00000000090, ENSMUSP00000080994	
6	1	ENSMUSP00000038961, ENSMUSP00000014457, ENSMUSP00000000090	

Figure 5. NetMatchStar result table for the matching between the query and the target networks of Figure 1.

been chosen to perform the significance test. When the simulation ends, a new window will appear with the following measures: the number of query occurrences in the real network, the mean and the standard deviation of the number of query occurrences in the random networks, the p-value and the z-score. The statistics of the test will be also reported on the “Log” panel located at the bottom of the “Matching” panel (Figure 1) and they can be consulted anytime.

Results

We evaluated the performance of NetMatchStar on the biological networks provided in 24 and compared it to the original NetMatch, developed for Cytoscape 2.8.

In Cytoscape others software are available for network motifs search. CytoKavosk⁴⁰ is based on counting all k-size sub-graphs of a given network graph, while GraMoFoNe⁴¹ emulates the interface of NetMatchStar by allowing users to define a query and finding all occurrences similar to the query, with respect to node and edge deletions and node similarities. NetMatchStar contains predefined motif structures, checks the significance of a motif with respect to seven different random models and allows user to draw queries containing wildcards and manage the approximation they need.

Figure 6 depicts the evaluation of NetMatchStar on three protein-protein interaction networks: *Mus musculus*, *Homo sapiens* and *Danio rerio*. They are large dense graphs. We randomly labeled networks with 32, 64, 128, 512 and 2048 synthetic labels and with 43 real labels corresponding to the Gene Ontology (GO) classes of the proteins (i.e. the nodes in the network). We used queries extracted from the networks by varying the number of nodes from

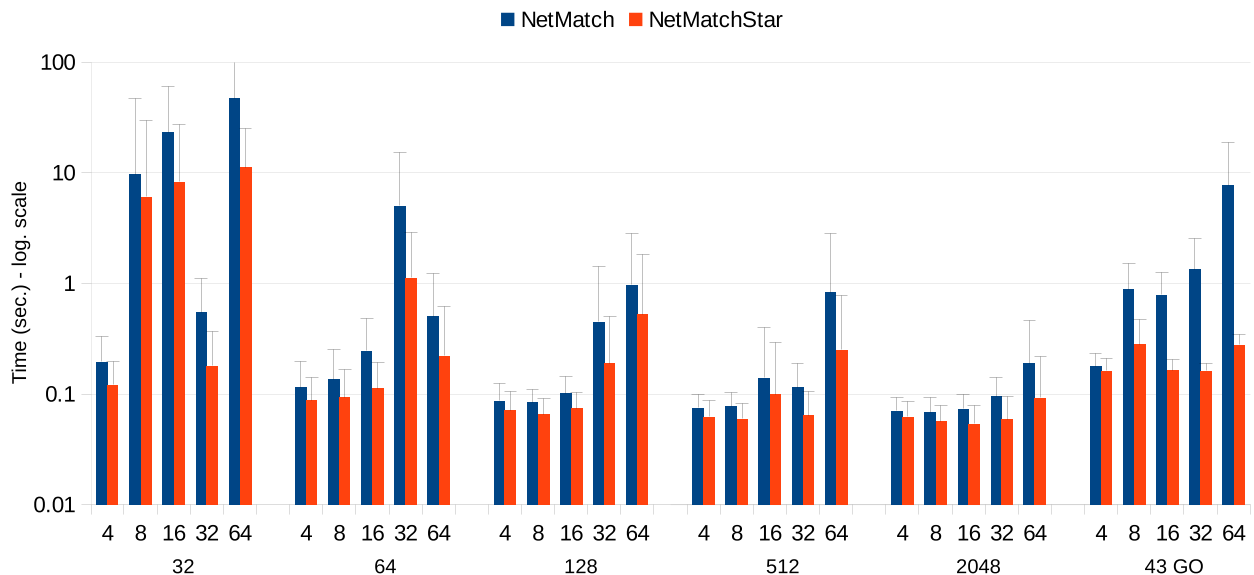


Figure 6. Query execution time on PPI networks.

4, 8, 16, 32, and 64 and density from low to high (up the 90% of edges among nodes are present).

Figure 7 evaluates NetMatchStar on protein back-bones graphs. They are large sparse graphs. The original labels are maintained since they are not unique (i.e., atoms names).

Figure 8 evaluates NetMatchStar on contact map graphs. They are dense medium graphs. The original labels are maintained since they are not unique (i.e., amino acids).

Figure 9 reports the querying performance of feed forward loop topology on *Mus musculus* with 512 labels. Queries are run exactly and approximated by unspecified one, two and all node labels and replacing one edge with an approximate path constrained to less than 3 and 7 edges.

Finally, for those queries we verified their statistical significance by using all random models (Figure 10) and we measured the average time required for generating random networks and searching the queries (Figure 11).

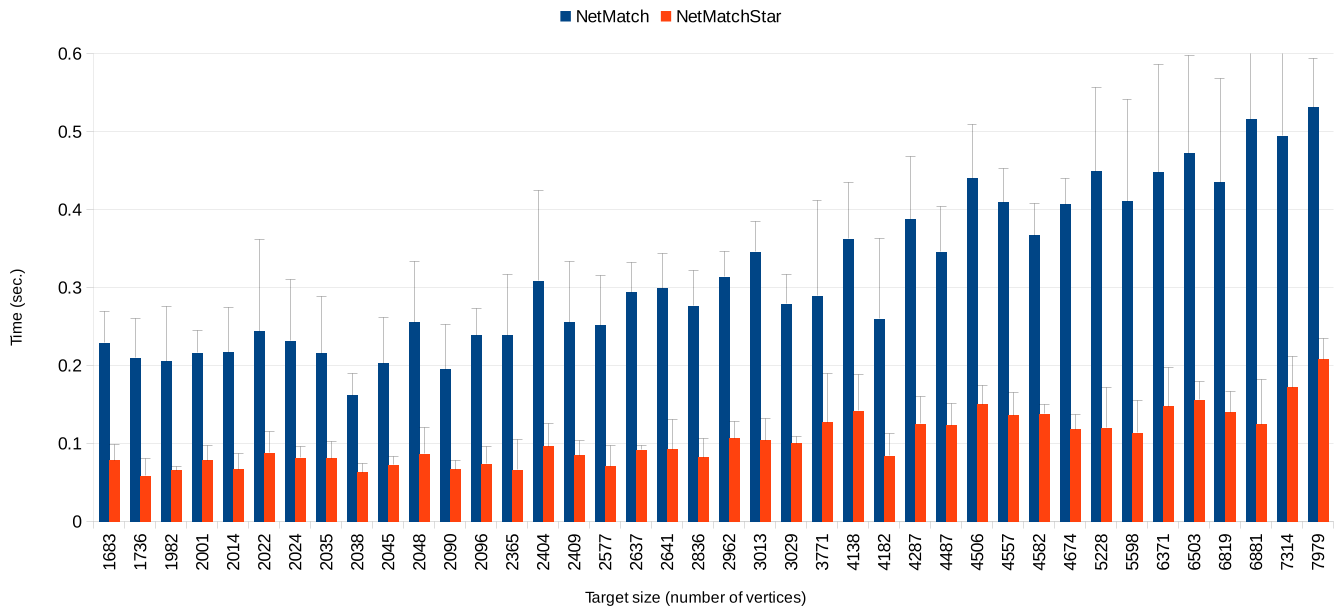


Figure 7. Query execution time on 3d-proteins.

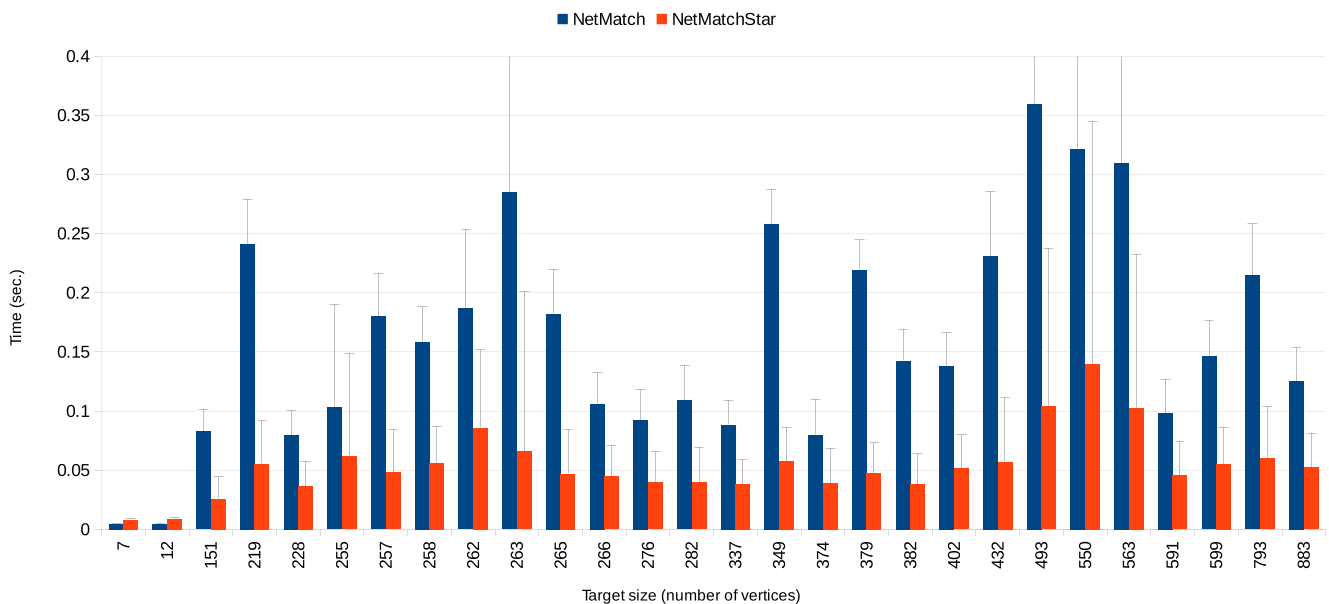


Figure 8. Query execution time on protein contact maps.

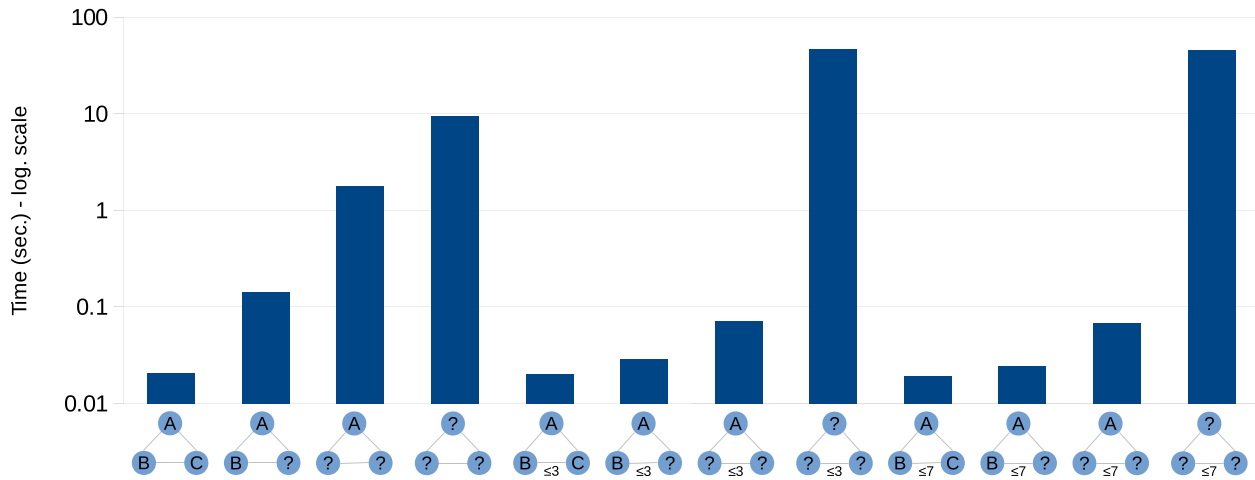


Figure 9. Feed forward loop with wildcards running time on *Mus musculus* with 512 labels.

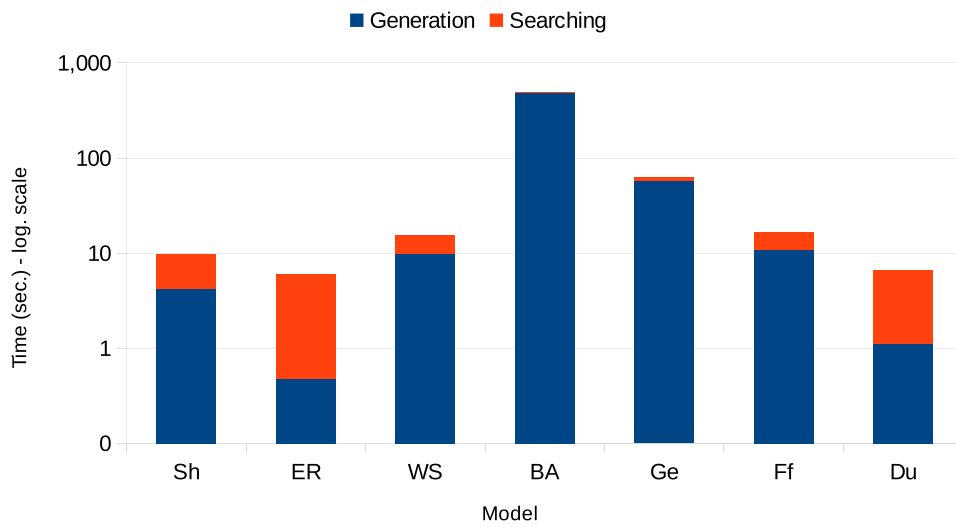


Figure 10. Running times for generating random networks and searching the feed forward loop on *Mus musculus* with no labels according to Shuffling (Sh), Erdos-Renyi (ER), Watts-Strogatz (WS), Barabasi-Albert (BA), Geometric (Ge), Forest Fire (Ff) and Duplication (Du) models.

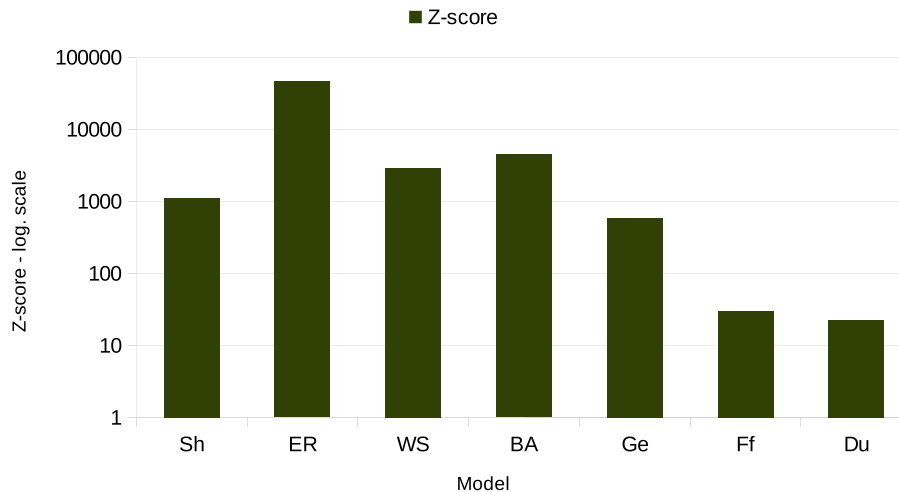


Figure 11. Z-scores for the feed forward loop on *Mus musculus* with no labels according to Shuffling (Sh), Erdos-Renyi (ER), Watts-Strogatz (WS), Barabasi-Albert (BA), Geometric (Ge), Forest Fire (Ff) and Duplication (Du) models.

Summary

This paper presented the biological network querying system NetMatchStar for Cytoscape 3.2.1. NetMatchStar improves upon its predecessor NetMatch in usability and performances. Moreover, it allows a comprehensive evaluation of statistical query significance. Future work includes semantic and ontological similarity search.

Software availability

This section will be generated by the Editorial Office before publication. Authors are asked to provide some initial information to assist the Editorial Office, as detailed below.

Software available from

<http://apps.cytoscape.org/apps/netmatchstar>,
<http://alpha.dmi.unict.it/netmatchstar/>

Latest source code

<https://github.com/fabiorinnone/NetMatchStar/tree/v3.1>

Link to source code as at time of publication

<http://dx.doi.org/10.5281/zenodo.1904542>

License

Creative Commons Attribution-NonCommercial-ShareAlike 3.0 License.

Author contributions

RG conceived the project. FR, GM and VB enabled the porting of the code to the new versions of Cytoscape (from version 3.0 onward) and extended the functionality of the application. In particular, GM has worked on the statistical significance and VB on the graph matching algorithm. RG, GM, and AP wrote the main parts of the paper. All authors have tested, validated, improved the software and the manuscript.

Competing interests

No competing interests were disclosed.

Grant information

RG, AP and AF have been funded by Programma Operativo Fondo Europeo per lo Sviluppo Regionale (PO-FESR 2007-2013), Linea di intervento 4.1.1.2. Grant number: CUP G23F11000840004.

Acknowledgments

We would like to thank A. Cannella and D. Garofalo who worked in a preliminary porting on the software for the current version of Cytoscape; D. Skripin who developed NetMatch, M. Mongiovì who implemented, in the old version of the system, the motif verification based on shuffling model, and G. Pigola who worked in the old version in query output visualization and optimization. We also are grateful to all users of NetMatch for their contributions and suggestions.

References

1. Albert R, Barabási AL: **Statistical mechanics of complex networks.** *Rev Mod Phys.* 2002; **74**(1): 47.
[Publisher Full Text](#)
2. Milo R, Shen-Orr S, Itzkovitz S, et al.: **Network motifs: simple building blocks of complex networks.** *Science.* 2002; **298**(5594): 824–827.
[PubMed Abstract](#) | [Publisher Full Text](#)
3. Mete M, Tang F, Xu X, et al.: **A structural approach for finding functional modules from large biological networks.** *BMC Bioinformatics.* 2008; **9**(Suppl 9): S19.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
4. Rhrissorakrai K, Gunsalus KC: **MINE: Module identification in networks.** *BMC Bioinformatics.* 2011; **12**: 192.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
5. Adamcsek B, Palla G, Farkas IJ, et al.: **CFinder: locating cliques and overlapping modules in biological networks.** *Bioinformatics.* 2006; **22**(8): 1021–1023.
[PubMed Abstract](#) | [Publisher Full Text](#)
6. Wernicke S, Rasche F: **FANMOD: a tool for fast network motif detection.** *Bioinformatics.* 2006; **22**(9): 1152–1153.
[PubMed Abstract](#) | [Publisher Full Text](#)
7. Wernicke S: **Efficient detection of network motifs.** *IEEE/ACM Trans Comput Biol Bioinform.* 2006; **3**(4): 347–359.
[PubMed Abstract](#) | [Publisher Full Text](#)
8. Alon U: **Network motifs: theory and experimental approaches.** *Nat Rev Genet.* 2007; **8**(6): 450–461.
[PubMed Abstract](#) | [Publisher Full Text](#)
9. Bader GD, Hogue CWV: **An automated method for finding molecular complexes in large protein interaction networks.** *BMC Bioinformatics.* 2003; **4**: 2.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
10. Grochow JA, Kellis M: **Network motif discovery using subgraph enumeration and symmetry-breaking.** In *Research in Computational Molecular Biology*. Springer, 2007; **4453**: 92–106.
[Publisher Full Text](#)
11. Ribeiro P, Silva F: **Discovering colored network motifs.** In *Complex Networks V*. Springer, 2014; **549**: 107–118.
[Publisher Full Text](#)
12. Ribeiro P, Silva F: **G-Tries: a data structure for storing and finding subgraphs.** *Data Min Knowl Discov.* 2014; **28**(2): 337–377.
[Publisher Full Text](#)
13. Ferro A, Giugno R, Pigola G, et al.: **NetMatch: a Cytoscape plugin for searching biological networks.** *Bioinformatics.* 2007; **23**(7): 910–912.
[PubMed Abstract](#) | [Publisher Full Text](#)
14. Banks E, Nabieva E, Peterson R, et al.: **NetGrep: fast network schema searches in interactomes.** *Genome Biol.* 2008; **9**(9): R138.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
15. Bruckner S, Huffner F, Karp RM, et al.: **Topology-free querying of protein interaction networks.** *J Comput Biol.* 2010; **17**(3): 237–252.
[PubMed Abstract](#) | [Publisher Full Text](#)
16. Micale G, Pulvirenti A, Giugno R, et al.: **GASOLINE: a Greedy And Stochastic algorithm for optimal Local multiple alignment of Interaction NETworks.** *PLoS One.* 2014; **9**(6): e98750.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
17. Micale G, Continella A, Ferro A, et al.: **GASOLINE: a Cytoscape app for multiple local alignment of PPI networks.** [v2; ref status: indexed, <http://f1000r.es/4f7>] *F1000Res.* 2014; **3**: 140.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
18. Kalaev M, Bafna V, Sharan R: **Fast and accurate alignment of multiple protein networks.** *J Comput Biol.* 2009; **16**(8): 989–999.
[PubMed Abstract](#) | [Publisher Full Text](#)
19. Sahraeian SME, Yoon BJ: **SMETANA: accurate and scalable algorithm for probabilistic alignment of large-scale biological networks.** *PLoS One.* 2013; **8**(7): e67995.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
20. Flannick J, Novak A, Srinivasan BS, et al.: **Graemlin: general and robust alignment of multiple large interaction networks.** *Genome Res.* 2006; **16**(9): 1169–1181.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
21. Liao C, Lu K, Baym M, et al.: **IsoRankN: spectral methods for global alignment of multiple protein networks.** *Bioinformatics.* 2009; **25**(12): i253–258.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
22. Ullmann JR: **An algorithm for subgraph isomorphism.** *J ACM.* 1976; **23**(1): 31–42.
[Publisher Full Text](#)
23. Cordella LP, Foggia P, Sansone C, et al.: **A (sub)graph isomorphism algorithm for matching large graphs.** *IEEE Trans Pattern Anal Mach Intell.* 2004; **26**(10): 1367–1372.
[PubMed Abstract](#) | [Publisher Full Text](#)
24. Bonnici V, Giugno R, Pulvirenti A, et al.: **A subgraph isomorphism algorithm and its application to biochemical data.** *BMC Bioinformatics.* 2013; **14**(Suppl 7): S13.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
25. Solnon C: **AllDifferent-based filtering for subgraph isomorphism.** *Artif Intell.* 2010; **174**(12–13): 850–864.
[Publisher Full Text](#)
26. Zampelli S, Deville Y, Solnon C: **Solving subgraph isomorphism problems with constraint programming.** *Constraints.* 2010; **15**(3): 327–353.
[Publisher Full Text](#)
27. Ullmann JR: **Bit-vector algorithms for binary constraint satisfaction and subgraph isomorphism.** *J Experimental Algorithmics (JEA).* 2010; **15**: 1.6.
[Publisher Full Text](#)
28. Han WS, Lee J, Lee JH: **Turbo_{iso}: towards ultrafast and robust subgraph isomorphism search in large graph databases.** In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*. ACM, 2013; 337–348.
[Publisher Full Text](#)
29. Shang H, Zhang Y, Lin X, et al.: **Taming verification hardness: an efficient algorithm for testing subgraph isomorphism.** *Proceedings of the VLDB Endowment.* 2008; **1**(1): 364–375.
[Publisher Full Text](#)
30. Zhang S, Li S, Yang J: **GADDI: distance index based subgraph matching in biological networks.** In *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology*. ACM, 2009; 192–203.
[Publisher Full Text](#)
31. Zhao P, Han J: **On graph query optimization in large networks.** *Proceedings of the VLDB Endowment.* 2010; **3**(1–2): 340–351.
[Publisher Full Text](#)
32. Szklarczyk D, Franceschini A, Kuhn M, et al.: **The STRING database in 2011: functional interaction networks of proteins, globally integrated and scored.** *Nucleic Acids Res.* 2011; **39**(Database issue): D561–D568.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
33. Milo R, Kashtan N, Itzkovitz S, et al.: **On the uniform generation of random graphs with prescribed degree sequences.** *Condensed Matter.* 2004; **2**: 1–4.
[Reference Source](#)
34. Erdos P, Renyi A: **On random graphs i.** *Publicationes Mathematicae.* 1959; **6**: 290–297.
[Reference Source](#)
35. Watts DJ, Strogatz SH: **Collective dynamics of ‘small-world’ networks.** *Nature.* 1998; **393**(6684): 440–442.
[PubMed Abstract](#) | [Publisher Full Text](#)
36. Barabasi AL, Albert R: **Emergence of scaling in random networks.** *Science.* 1999; **286**(5439): 509–512.
[PubMed Abstract](#) | [Publisher Full Text](#)
37. Penrose M: **Random Geometric Graphs.** Oxford Studies in Probability 5. Oxford University Press, 2003.
[Reference Source](#)
38. Leskovec J, Kleinberg J, Faloutsos C: **Graphs over time: densification laws, shrinking diameters and possible explanations.** In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*. KDD '05, New York, NY, USA, ACM, 2005; 177–187.
[Publisher Full Text](#)
39. Chung F, Lu L, Dewey TG, et al.: **Duplication models for biological networks.** *J Comput Biol.* 2003; **10**(5): 677–687.
[PubMed Abstract](#) | [Publisher Full Text](#)
40. Masoudi-Nejad A, Ansariola M, Kashani ZR, et al.: **CytoKavosh: a Cytoscape plug-in for finding network motifs in large biological networks.** *PLoS One.* 2012; **7**(8): e43287.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
41. Blin G, Sikora F, Vialette S: **GraMoFoNe: a Cytoscape plugin for querying motifs without topology in protein-protein interactions networks.** In Hisham Al-Mubaid, editor, *Bioinformatics and Computational Biology (BIOB' 10)*. International Society for Computers and their Applications (ISCA), 2010; 38–43.
[Reference Source](#)
42. Rinnone F, Micale G, Bonnici V, et al.: **NetMatch-Star: v3.1.** *Zenodo.* 2015.
[Data Source](#)

Open Peer Review

Current Referee Status:



Version 1

Referee Report 21 September 2015

doi:10.5256/f1000research.7150.r10336



Shaillay Dogra

Vishuo BioMedical Pte Ltd, Singapore, Singapore

Algorithmically, is a useful utility that has been implemented by the authors. Bringing out its multiple use-case scenarios in biological setting with interpretations for example will help novice users adapt this more widely. Biological explanations of network interpretations will strengthen understanding and usage. Also, there are 7 different algorithms implemented. In which kind of data or user-problem which algorithm is recommended, according to authors or empirical observations?

Minor points:

- "Algorithms using such a technique include.." -- kindly mention here itself
- "the RI algorithm proposed for biological real networks which outperforms other existing algorithms.." -- please elaborate on this RI algorithm
- "Values of these metrics can suggest to the user which random model" - can the authors recommend some approximation of good/bad values?

I have read this submission. I believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Competing Interests: No competing interests were disclosed.

Referee Report 15 September 2015

doi:10.5256/f1000research.7150.r10314



Xifeng Yan

Computer Science Department, University of California, Santa Barbara, Santa Barbara, CA, USA

NetMatchStar improves the previous NetMatch tool, a Cytoscape app developed by the authors. It has more functions (e.g., supporting approximate querying with wild cards) and runs faster. It shall be interesting to users who are looking for tools that are able to find all the occurrences of a query graph in a network and check for its significance under different null models. The manuscript has detailed description and evaluation results.

I have read this submission. I believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Competing Interests: No competing interests were disclosed.
