11-18-2021

# Development of a Novel Algorithm to Remove Spurious Edges from Biological Networks Through Functional Enrichment

Jeffrey Page
jmp5426@rit.edu

RIT

# Development of a Novel Algorithm to Remove Spurious Edges from Biological Networks Through Functional Enrichment

By

Jeffrey Page

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Bioinformatics

Committee: Dr. Gary Skuse, Dr. Gordon Broderick, and Dr. Matt Morris

School/Department of Thomas H. Gosnell School of Life Sciences

College of Science

Rochester Institute of Technology

Rochester, NY 14623-5603

November 18, 2021

**Table of Contents:**

**Table of Figures and Tables**

**Abstract**

The field of systems biology has facilitated the modelling of large and complex biological networks. These networks, generated from prior knowledge contained in the corpus of medical and scientific literature, or from experimental data are being used to model differing macromolecule networks associated with distinct disease states. While these networks are vital in understanding disease pathology and possible treatment options, they are rife with spurious interactions. These interactions arise from the methods used to create such networks, where the ability to discriminate between direct and indirect relationships is a challenge. To combat these spurious interactions an algorithm that leverages functional enrichment in biological networks was developed. Here, functional enrichment refers to two or three node functional motifs that are ubiquitous in biological networks. The algorithm developed removes edges from an existing network based on that edge's involvement in functional motifs relative to every other edge's involvement. In this work, the application of this algorithm was explored using real-world clinical disease networks. Furthermore, a software package was developed to identify an edge's membership in functional motifs with respect to the network being explored. The tools developed in this work are the first to critically analyze an edge's relationship to functional motifs in terms of network inclusion. Therefore, the principles outlined in this work can be employed in future works aimed at removing spurious edges. These principles will also produce higher quality biological networks for the understanding of disease pathology and the development of more effective treatment options.

## Introduction

Graphs are the fabric of complex systems such as social networks, metabolic networks, or even gene regulatory networks[1]. Mathematical models of networks, or graphs, are composed of edges and nodes, where each node is a discrete object, and each edge is a connection between said nodes. The use of networks in studying proteins within the human body has been vital in predicting and characterizing interactions between proteins to further understand our own biology. The two basic types of networks used to model biological systems are undirected graphs and directed graphs. Direction here refers to a distinction in edge characteristics; in a directed graph an edge is unidirectional, in an undirected graph every edge is bidirectional[2]. This distinction allows for robust edge classification where nodes can have cause and effect on each other; for example, one node may promote the activity of another node in a directed network or inhibit the activity of another node. Directed graphs are generally more useful when modeling biological systems as they allow for an edge to have a mode of action. For example, in a PPI (protein-protein interaction) network, directed edges can be classified as being activating or inhibiting[3].

Directed networks have been leveraged in the discovery of new drug targets in networks associated with differing disease states[4]. The creation of these biological networks has been accelerated through the usage of text-mining software. A researcher can create a network by mining the pairwise interactions of the molecules of interest(protein, metabolite, mRNA) from large bodies of research along with each edge's mode of action (activation or inhibition). Biological networks can also be constructed though the usage of wet lab data. Scientists can use data from HPLC, GC-MS, LC-MS, RNA-seq, among a plethora of other types of data, to reverse

engineer networks[5,6]. However, creation of these networks necessitates validation and/or modification of the inferred edges.

**Approaches to creating plausible directed networks**

The process of reverse engineering biological networks from prior knowledge(text-mining), or from wet lab data is currently plagued by the specter of inaccuracy. Inaccuracy here refers to the inclusion of edges that do not belong, or the exclusion of edges that do belong. Each method to reverse engineer networks comes with its own unique challenges. Methods to recover networks from wet lab data results in the occurrence of spurious edges as the methods that process this data cannot discriminate between direct and indirect relationships[7]. The construction of networks from text-mining scientific literature results in the inclusion of edges that once again arise from indirect relationships. Generally, text-mining algorithms to reverse engineer biological networks are lacking in precision. Since most methods to reverse engineer biological networks from data cannot seem to distinguish true interactions from spurious interactions, there is a tremendous need for an algorithm to prune false interactions.

There are various methods to identify and remove spurious edges from biologically informed networks. Most of these methods invoke different structural principles common across previously established biological networks, such as modularity, connection density etc. For example, work done by Liming Pan and colleagues removes edges based on the principle that two nodes will have a high probability of making a link between them if they share some common neighbors or are connected by short paths[8]. Other researchers such as Guimerà and others, remove spurious edges using properties of stochastic block modelling[9]. These properties are that nodes in a real network are organized into modules, and that each network contains nodes that have certain roles and connect to other nodes in distinct ways according to their role.

Novel approach to spurious edge removal

There are very few methods that trim spurious edges from biological networks on a purely functionally informed basis. Functionality here refers to the way a network dynamically processes the flow of information. It can be inferred by how it incorporates functional subgraphs, or network motifs into its structure. The work done by Uri Alon highlights the ubiquitous nature of two and three node motifs in biological networks[10]. These motifs work to form regulatory circuits within a network that can regulate and temper a response to a given stimuli presented to them. There are eight differing types of 3 node motifs (figure 1); each of which can change when a given signal presented to them to produce the desired biological
effect.

Each motif works in a distinct way, allowing an intricate interplay of proteins that result in cellular processes such as the upregulation or downregulation of the expression of a particular gene. They do so by serving as filters, pulse generators and response accelerators among various other functions. The three node motifs presented in figure 1 are present in many differing biological networks across many differing organisms, suggesting selective pressure specifically for these functional units.8 It follows that biological networks enriched in these motifs have strong support in the way of both functional and evolutionary feasibility. Identifying the number and types of network motifs within a given biological network is therefore a useful metric in characterizing the network as being biologically sound.

Functional motif enrichment in biological networks can be exploited to develop a method for spurious edge removal. A process that removes edges from a biological network to maximally enrich the number of functional motifs in that network works to remove edges that may not contribute to any real process. This theory will be tested in the current work using software developed to both identify all functional motifs in a network and remove edges from said network; calculating several different metrics along the way to ensure proper edge removal and a functionally enriched biological network.

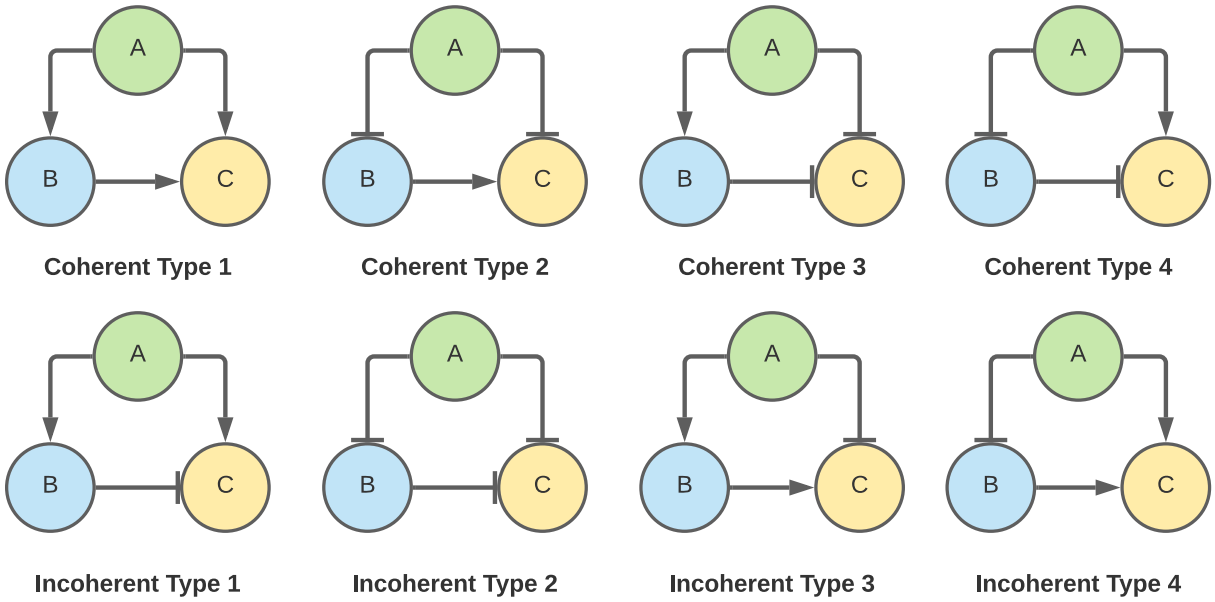


**Figure 1**: Types of feedforward network motifs, arrow represents activation whereas flat bar represents inhibition

## Materials and Methods

### Motif Identification

For the purposes of this project several differing kinds of network motifs will be identified. They include three node feedforward motifs, three node feedback motifs, and two node feedback motifs. Each edge in the network belonging to a motif will be recorded. An individual edge is eligible to belong to multiple differing motifs, shown in figure 3. Finally, there must be an output of each edge in the network with each type and number of motifs it belongs to. There are numerous differing software packages that have been developed to identify motifs in directed network[11,12,13]. However, no single package meets the outlined criteria listed previously. As such, a software package was developed for usage in this project.

This package, provided an edge list, scans the entire network and outputs a comma separated file containing an edge list depicted in table 1.

| Source | Target | C_1 | C_2 | C_3 | C_4 | I_1 | I_2 | I_3 | I_4 | 3_FB | 2_NFB | 2_PFB |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | B | 3 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| Q | R | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| T | A | 5 | 1 | 0 | 0 | 2 | 0 | 0 | 1 | 0 | 0 | 1 |

**Table 1**: Example motif identification output. C1-C4 are represent coherent feedforward motifs 1-4, I1-I4 represent incoherent feedforward motifs 1-4, 3_FB represents three node feedback motifs, 2_NFB represents two node negative feedback motifs and finally 2_PFB represents two node positive feedback motifs

**Identification Algorithms**

The motif identification package utilizes several different algorithms to locate and map every motif to the involved edge(s). Each algorithm was implemented in python 3.6.3 leveraging the Networkx package[14]. Below, each algorithm will be outlined in pseudocode.

**Three node feed forward motifs**:

For every node (a) in the network:

    If the node has two or more outgoing edges:

        Create a set of all nodes connected to these edges: (alpha)

        For every node (b) in (alpha):

            Create a set of all nodes that are connected to the current node in (alpha), named (beta)

            If one member (c) of (beta) is included in (alpha), then (a), (b) and (c) are all classified as belonging to a 3 node feedforward motif

**Three node feedback:**

For every node (a) in the network:

For all outgoing edges of (a), add target nodes to list (alpha):

For all nodes (b) in alpha:

For all outgoing edges of (b), add target nodes to list (beta):

For all nodes (c) in beta:

If node (c) is the same as node (a):

nodes (a),(b) and (c) are a part of a three node feedback motif

**Two Node Feedback:**

For every node (a) in the network:

For every outgoing edge of (a) if edge is bidirectional:

(a) and target node are a part of a two-node motif

**Network Motif Enrichment**

Overall functional motif enrichment will be leveraged to remove spurious edges from an input network. In order to enrich a network, first all motifs will be identified and all edges will be ranked according to the number of motifs they belong to. From this ranking, edges will be removed in an iterative fashion with a rescan of the network and a reranking of edges after every removal to identify functional motifs.

**Sources, Sinks and Disjoints**

Before the removal of an edge from the input network, it must be decided whether this removal will create a source node, a sink node, or a disconnected graph. A source node is defined as a node with an indegree of 0, whereas a sink node has an outdegree of 0. Finally, a disconnected graph is a graph in which at least two nodes do not have a connecting path to each other. Sources, sinks, and disconnects are modelled in figure 2. If removal of an edge does result in the creation of a source, sink, or disconnect it is simply skipped and not removed.

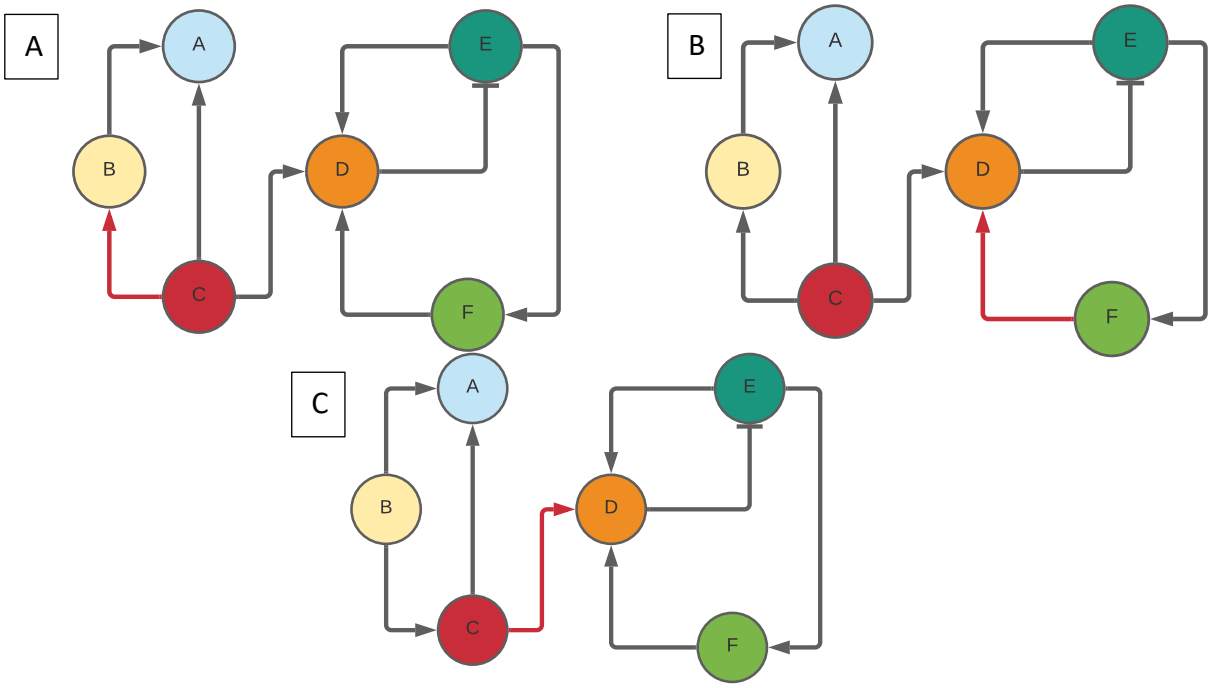


**Figure 2:** Examples of Edges that cannot be removed. In A, removal of edge (C,B) would result in node B becoming a source. In B, removal of edge (F,D) would result in node F becoming a sink. In C, removal of edge (C,D) would create a disconnected graph.

**Enrichment Score**

After each removal, the mean number of motifs that each edge in the network is a part of will be calculated; this value will be referred to as the network's enrichment score. After each of these iterations, the network's enrichment score is saved and compared to the previous scores. Once the enrichment score reaches a maximum, the pruning process is complete, and the input network is considered completely functional enriched. This process is described in Figure 4.

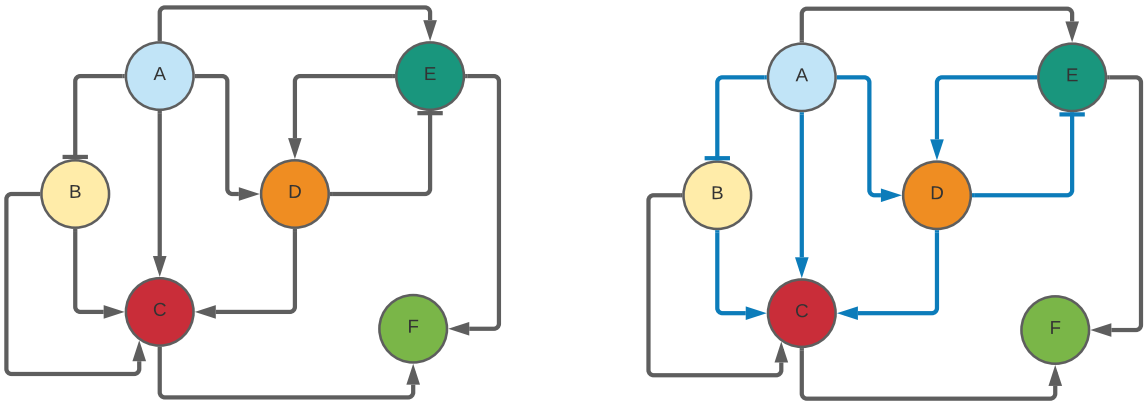


**Figure 3**: Identification of edges involved in functional motifs. Edges involved are depicted in blue, here edges can belong to multiple motifs as seen in edge (A,C)

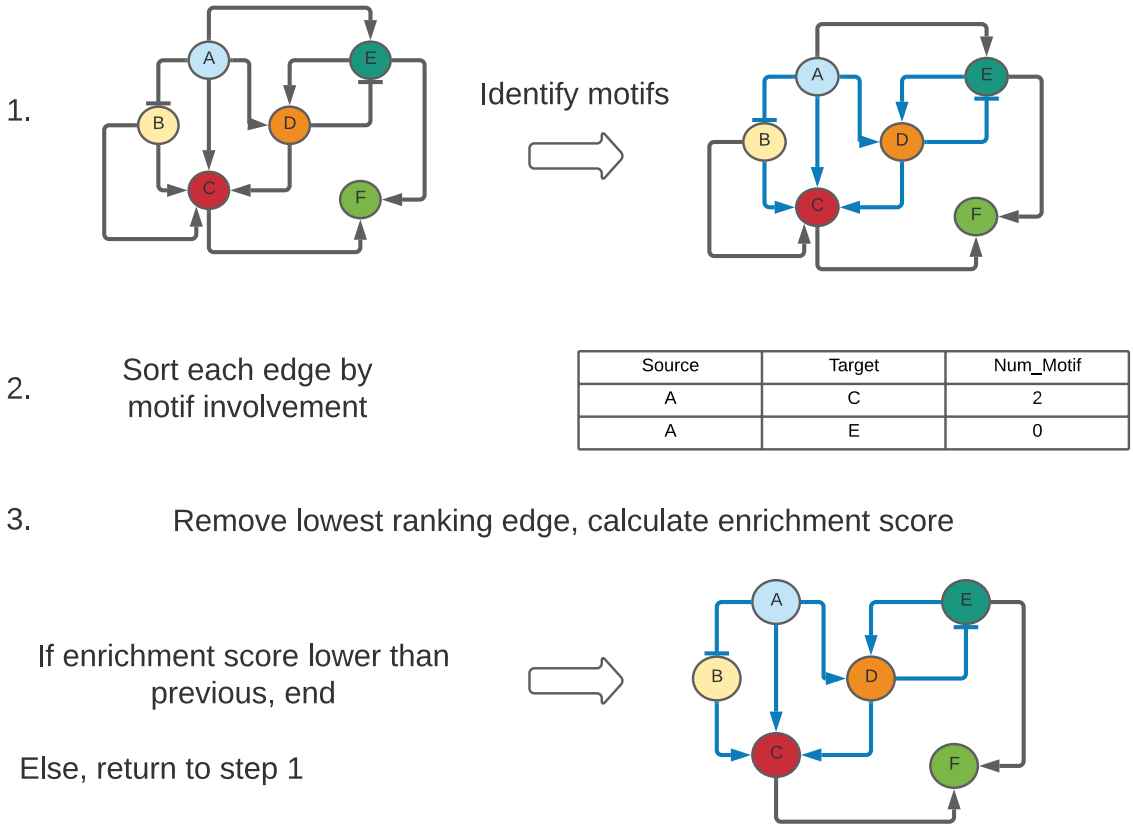


**Figure 4**: Flow chart depicting network motif enrichment. Enrichment score refers to the mean of the number of functional motifs each edge in the network is a part of. Sinks, sources and disconnects are completely avoided.

**Test Network Selection**

To test the aforementioned network motif enrichment algorithm, real world biological networks were required. As such this project will focus on the analysis of two real world clinical protein-protein networks. These networks all model chronic degenerative diseases in humans, and were created through the usage of text-mining software. Since each network contains proprietary information, all node names will be anonymized. They will be referred to as networks A and B.

**Network Metrics**

Networks A is much smaller in size in comparison to network B in terms of both edge and node count. The relative size difference between networks A and B will be useful when estimating the efficacy of the proposed edge pruning algorithm. A depiction of these networks is shown in figure 5, which highlights the extreme size difference between the starting networks. The motif composition of each network will also be analyzed through the usage of the motif identification software described previously.

| Network | Nodes | Edges | Connection Density |
|---|---|---|---|
| A | 53 | 243 | .088 |
| B | 107 | 1656 | .146 |

**Table 2**: Network metrics for networks A and B. Network B has roughly twice as many nodes as A, and roughly ten times as many edges.

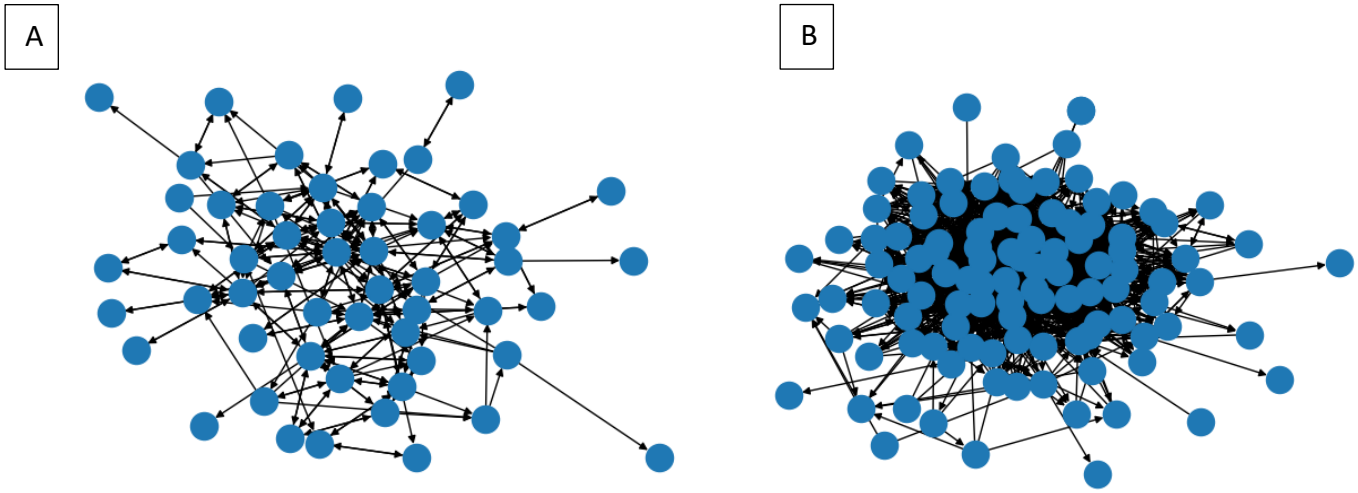


**Figure 5:** Visual Representations of networks A and B. Network B has many more nodes and edges when compared to network A.

**Algorithmic Comparison**

The sample networks, a and b, have been additionally edited using a method centered around the concept of constraint satisfaction[15]. This method results in many differing model solutions to approximate the "true" or most valid network. The output network of the previously mentioned functional enrichment algorithm will be compared to these models using the metric of graph edit distance. This comparison is useful in that the method using constraint satisfaction takes a considerable amount of time compared to the described method.

**Results**

In terms of motif composition, Networks A and B are drastically different. As shown in figures 6 and 7, Network B is dominated by Coherent Type 1 functional motifs. These motifs only have positive polarities associated with their edges, shown in Figure 1. Network B's edges are primarily positive, so the over-representation of Coherent Type 1 motifs is expected. In both networks, three node feedback motifs represent a large amount of the total motifs present. This can be attributed to the fact that three node feedforward motifs are broken down into eight differing types of motifs, whereas three node feedback motifs are counted as one entire category. Other than coherent type 1 motifs and three node feedback motifs, the remaining functional motifs are represented relative equal amounts.

Functional enrichment altered both networks A and B in differing ways. Network A had a larger change in both edges number and connection density, with a 24% loss in edge number and an 23.4% loss in connection density. Network B was altered less in comparison with a 10.6% loss in edge number and an 11% loss in connection density. In both Networks the percentage decrease in edges number and connection density seemed to be correlated. Visually, networks A and B generally look like their pre-functionally enriched states, as shown in figures 5 and 8. However, network A post-enrichment has noticeably less connections; this change in connection density is apparent in Figure 8.

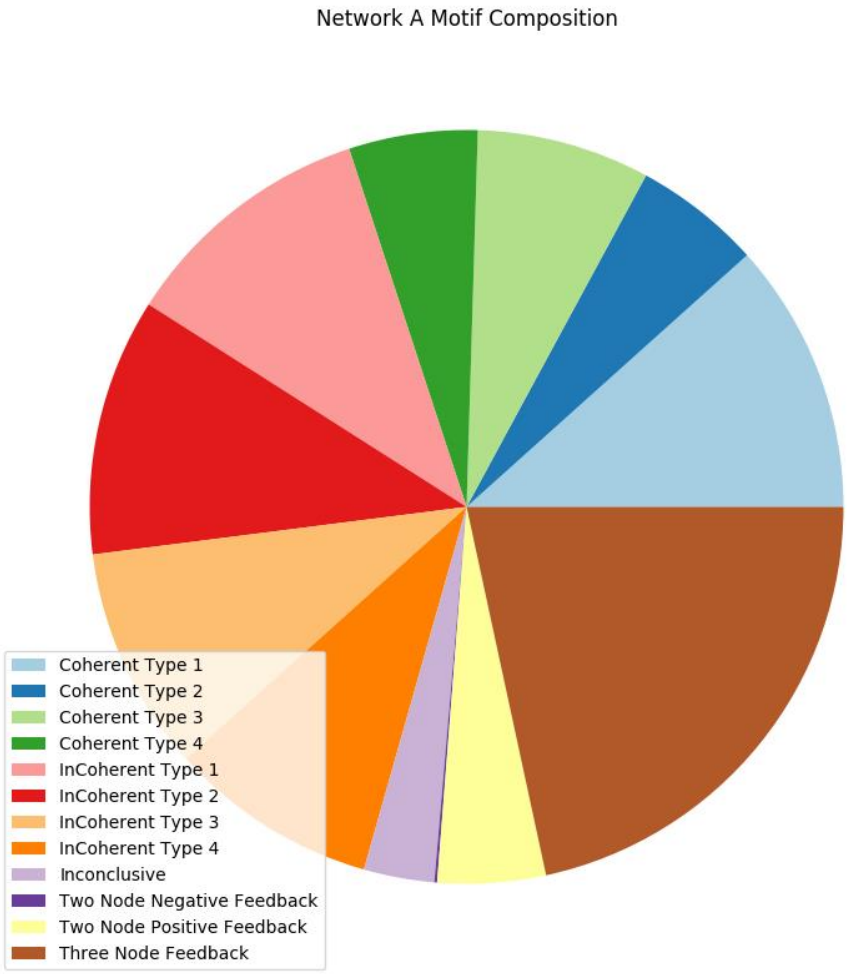


**Figure 6**: Network A Motif Composition previous to enrichment algorithm. Three node feedback motifs represent the greatest ratio of motifs.

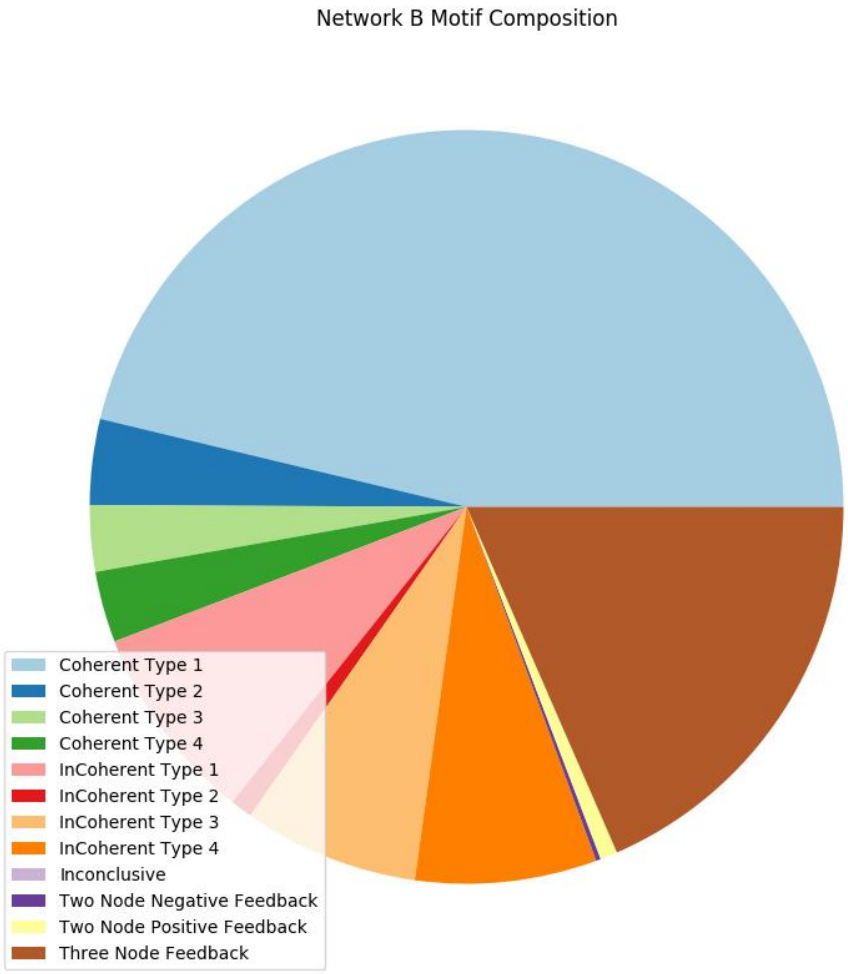


**Figure 7**: Network B motif composition previous to enrichment algorithm. Coherent Type 1 motifs are the majority

| Network | Nodes | Edges + (Δ)% | Connection Density+ (Δ)% |
|---|---|---|---|
| A (functionally enriched) | 53 | 185 (-23.4%) | .067 (-24%) |
| B (functionally enriched) | 107 | 1480 (-10.6%) | .13 (-11%) |

**Table 3:** Network change after functional enrichment. Network B was altered much less in comparison to network A, with a larger change in both connection density and edge number

| Network | Mean GED between Models | GED std. between Models | Mean GED from Pruned Network to Models | GED std. from Pruned Network to Models | GED from original network to functionally enriched network | Mean GED from original network to Models |
|---|---|---|---|---|---|---|
| B | 30.78 | 8.06 | 1501 | 3.81 | 138 | 1602 |
| A | 0.236 | 0.42 | 100.20 | 0.34 | 55 | 100.8 |

**Table 4:** Summarization of within group variation of graph edit distance in models created by constraint satisfaction compared to the graph edit distance of functionally enriched networks to each of the models

| Network | Nodes + (Δ)% | Edges + (Δ)% | Connection Density+ (Δ)% |
|---|---|---|---|
| A (constraint satisfaction) | 50.25 (-5.7%) | 142.01 (-41.5%) | .055-(37.5%) |
| B (constraint satisfaction) | 33.6 (-69%) | 127.4 (-92.3%) | .113(-11%) |

**Table 5:** Network change after constraint satisfaction. Here, nodes can be removed from the original network as shown by the Nodes + (Δ)% column

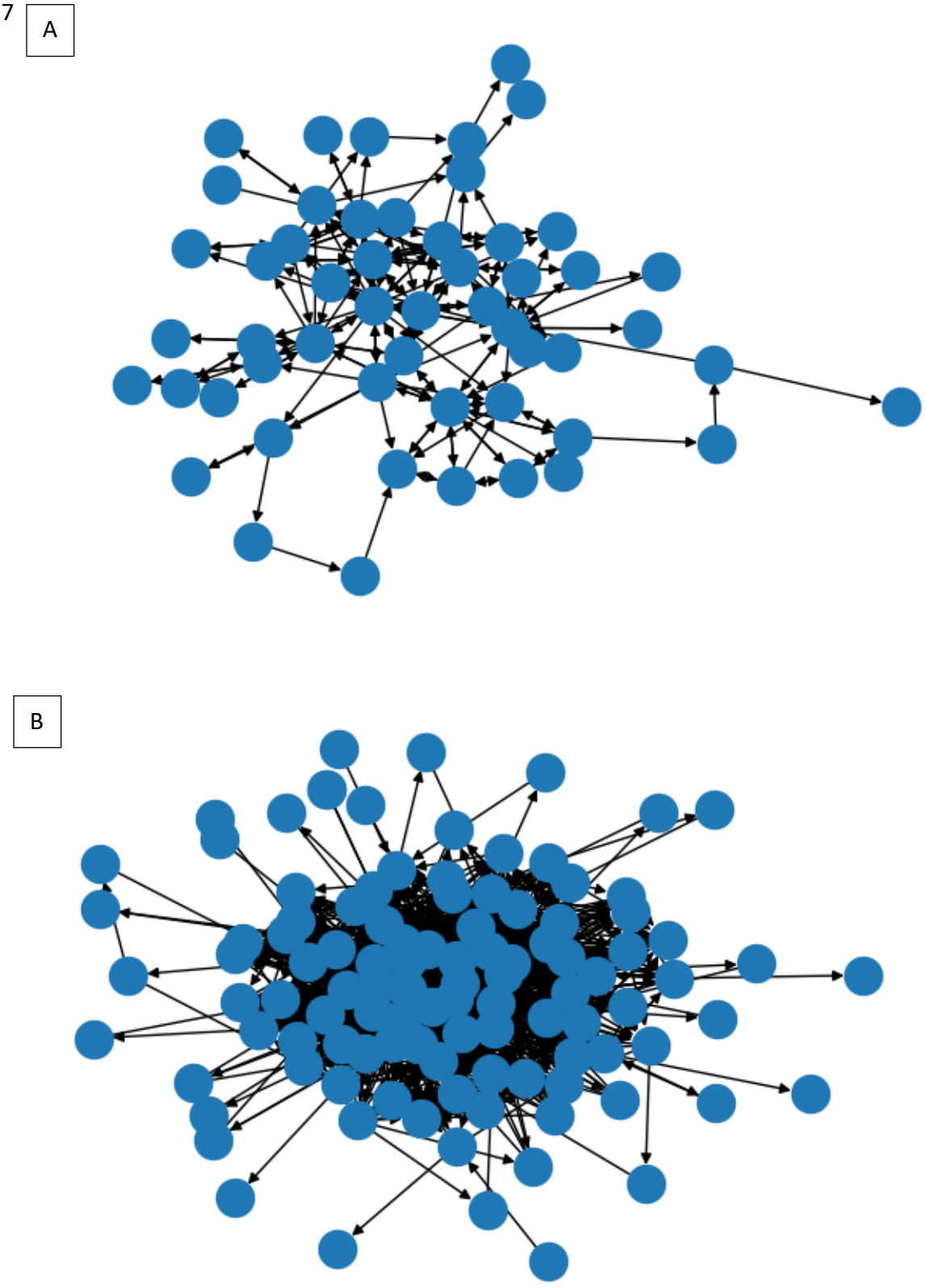


**Figure 8:** Graph visualization of networks A and B after functional enrichment algorithm.

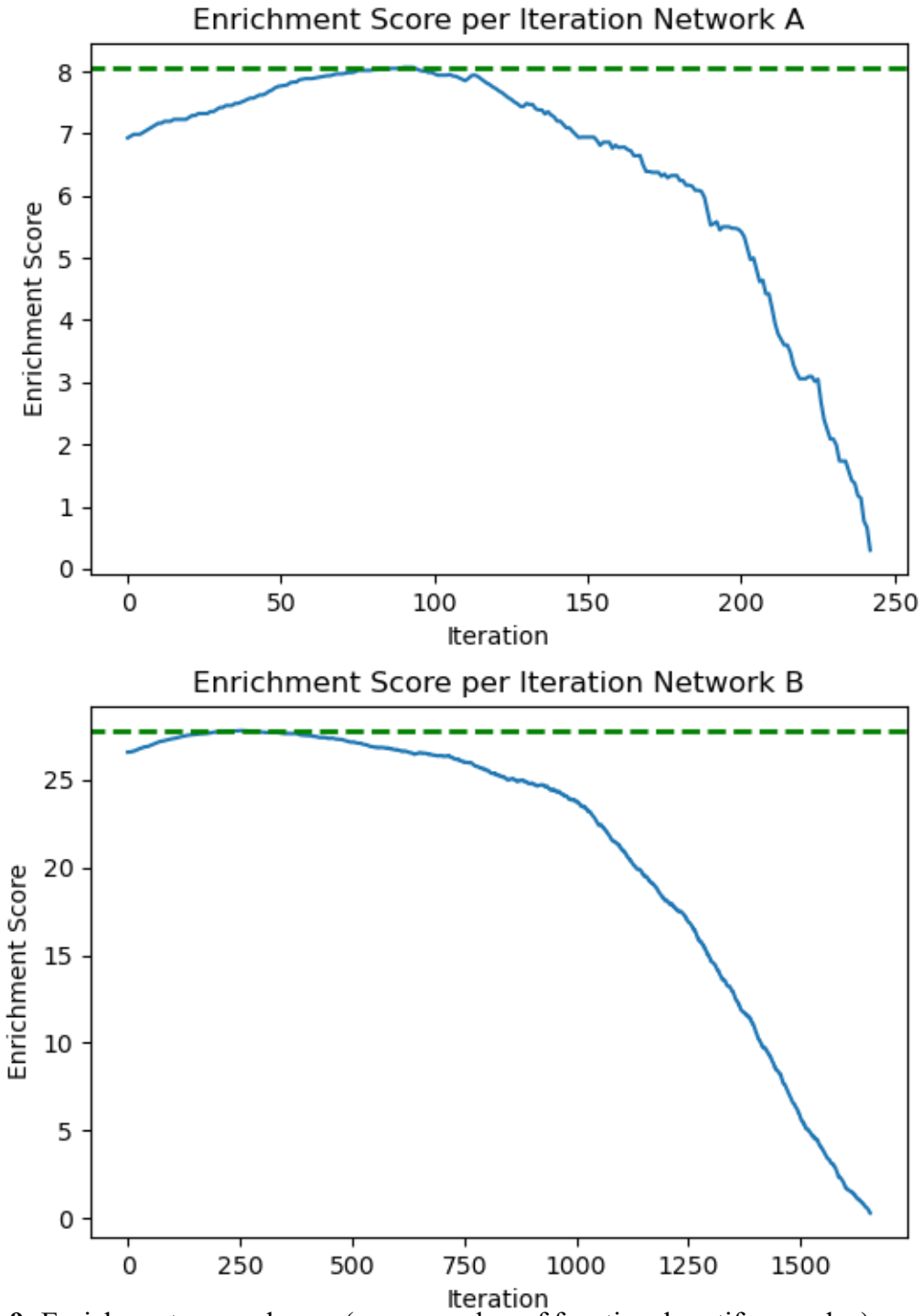


**Figure 9:** Enrichment score change (mean number of functional motifs per edge) per iteration of enrichment algorithm. The dotted green line identifies the maximum enrichment score, where the algorithm stops removing edges from the input network.

The process of functional enrichment is modelled in Figure 9. Here, after each removal of the lowest ranking edge, in terms of enrichment score, the enrichment score is recalculated and graphed. Each network is slowly enriched in functional motifs after each iteration, until a maximum is reached. After this, there is a plateau in enrichment score, then a steep drop, shown in Figure 9. Network A seems to reach maximum enrichment score before network B with respect to iteration number.

When comparing the method of spurious edge removal presented in this paper to the previously mentioned constraint satisfaction method[15] there are very different results in the final network produced by each method. The functional enrichment algorithm presented in this work does not remove any nodes in the original network, whereas the method of constraint satisfaction removed 5.7% and 69% of nodes from networks A and B respectively. As far as the other network metrics are concerned, both algorithms had similar connection densities and edge number in network A, shown in table 3 and 5. However, the metrics in network B were completely differing with the constraint satisfaction algorithm removing 92% of edges from the original network.

**Discussion**

A novel method of spurious edge removal through the process of functional enrichment was shown in this work. The innovation in this project was the consideration of an *edge's* involvement in functional motifs, where each edge could be involved in multiple functional motifs. Wang and coworkers have utilized node membership in motifs to calculate what they deemed "node importance" but did not use this measurement to remove or change the network analyzed.[16] Other works have used the total number of functional motifs in a network as a metric to calculate a sort of "motif centrality" measure to discover the "backbone" of a network.[17] However, no current method utilizes the motif membership of each *edge* as an indicator of spurious edge removal.

As a method of spurious edge removal the algorithm outlined in this work had differing results when applied to larger networks, such as network B, than when performed on smaller networks such as network A. The algorithm reached a maximum enrichment score and stopped removing edges much sooner when applied to the larger network than when applied to the smaller network. This effect could be attributed to the underlying difference in connection density from the starting networks, the sheer difference in network size, or simply the presence of more functionally involved edges in the larger network. In terms of the comparison of spurious edge removal through functional enrichment and the method of constraint satisfaction mentioned previously, both methods removed a similar number of edges and reduced connection density in a similar fashion in network A.

The larger difference between methods occurs in the analysis of network B. The method utilizing constraint satisfaction removed a considerable number of edges (92%), whereas the

method presented in this project only removed around 10% of edges. Even the pruned networks, A and B, produced by the two differing methods have vast graph edit distances from each other. The method using constraint satisfaction produced 30 model solutions for Network B, which had a graph edit distance standard deviation of ~8, and a mean graph edit distance of 30.78. The model solutions had therefore small variance between themselves but were very different from the network created from functional enrichment, with a mean graph edit distance of 1501 to each of the model solutions.

One of the main limitations of this study was the lack of testing the proposed algorithm on gold standard networks, or even a larger pool of experimental networks. To prove the efficacy of spurious edge removal using the outlined method of spurious edge removal through functional enrichment, testing needs to be done in trying to recover networks that have been altered through the addition of spurious edges. In order to do so the usage of previously established gold standard biological networks should be used[18]. Furthermore, the outlined algorithm could be tested using software packages that create synthetic biological networks based on common biological network properties[19,20] . In this way, several hundred experiments could be performed as testing will not be bounded by sample size.

There are many other questions to explore relating to this work. There seems to be a relationship between motif membership per edge and network size; a study using synthetic biological networks, or gold standard biological networks could be conducted using the software package written for this project. Furthermore, this algorithm could be combined with other algorithms that consider structural features of a network when removing spurious edges.[9,8] The stopping rule implemented in the third step of the pruning algorithm could also be altered. Instead of stopping the algorithm upon the incidence of a maximum value, the algorithm could

continue until a significantly different value from the maximum is reached. In other words, the maximum value would still be recorded and averaged with every other enrichment score calculated, when the next enrichment value is significantly different compared to this mean, then the algorithm will stop.

The algorithm outlined and implemented in this project could not remove nodes from an existing network. One possible change in the algorithm could be in the removal of nodes. This would allow greater edits to be done to the pre-existing network and could result in an even greater enrichment of functional motifs in each biological network.

**Conclusion**

In this work a software package was developed to identify all three and two node functional motifs in an input network. This package also outputs a csv containing information for each edge and which motifs they participate in. Currently this is the only motif identification software that identifies the extent to which each edge participates in network motifs. Therefore, this package can be used in future projects concerning the membership of edges in functional motifs.

The algorithm to remove spurious edges from biological network developed in this work has been shown enrich the number of functional motifs in a given network. This process removes edges which are non-important in terms of functional involvement and can therefore be used to remove spurious edges from experimental or prior knowledge biological networks. This algorithm needs further testing using gold standard biological networks in order to further prove its validity.

**Code Availability**

Implementation of the two and three node motif finder, and the functional enrichment algorithm are both available via github in the JMI package:

https://github.com/petrepage/JMI

**References**

(1) Ma, X.; Gao, L. Biological Network Analysis: Insights into Structure and Functions. *Brief. Funct. Genomics* **2012**, *11* (6), 434–442. https://doi.org/10.1093/bfgp/els045.

(2) Directed Graphs - an overview | ScienceDirect Topics https://www.sciencedirect.com/topics/computer-science/directed-graphs (accessed 2021 -11 -19).

(3) Ideker, T.; Sharan, R. Protein Networks in Disease. *Genome Res.* **2008**, *18* (4), 644–652. https://doi.org/10.1101/gr.071852.107.

(4) Lecca, P.; Priami, C. Biological Network Inference for Drug Discovery. *Drug Discov. Today* **2013**, *18* (5), 256–264. https://doi.org/10.1016/j.drudis.2012.11.001.

(5) Lefebvre, C.; Rieckhof, G.; Califano, A. Reverse-Engineering Human Regulatory Networks. *Wiley Interdiscip. Rev. Syst. Biol. Med.* **2012**, *4* (4), 311–325. https://doi.org/10.1002/wsbm.1159.

(6) M. Hendrickx, D.; B. Hendriks, M. M. W.; C. Eilers, P. H.; K. Smilde, A.; J. Hoefsloot, H. C. Reverse Engineering of Metabolic Networks, a Critical Assessment. *Mol. Biosyst.* **2011**, *7* (2), 511–520. https://doi.org/10.1039/C0MB00083C.

(7) De Smet, R.; Marchal, K. Advantages and Limitations of Current Network Inference Methods. *Nat. Rev. Microbiol.* **2010**, *8* (10), 717–729. https://doi.org/10.1038/nrmicro2419.

(8) Pan, L.; Zhou, T.; Lü, L.; Hu, C.-K. Predicting Missing Links and Identifying Spurious Links via Likelihood Analysis. *Sci. Rep.* **2016**, *6* (1), 22955. https://doi.org/10.1038/srep22955.

(9) Guimerà, R.; Sales-Pardo, M. Missing and Spurious Interactions and the Reconstruction of Complex Networks. *Proc. Natl. Acad. Sci. U. S. A.* **2009**, *106* (52), 22073. https://doi.org/10.1073/pnas.0908366106.

(10) Alon, U. Network Motifs: Theory and Experimental Approaches. *Nat. Rev. Genet.* **2007**, *8* (6), 450–461. https://doi.org/10.1038/nrg2102.

(11) Kouhsar, M.; Razaghi-Moghadam, Z.; Mousavian, Z.; Masoudi-Nejad, A. CeFunMO: A Centrality Based Method for Discovering Functional Motifs with Application in Biological Networks. *Comput. Biol. Med.* **2016**, *76*, 154–159. https://doi.org/10.1016/j.compbiomed.2016.07.009.

(12) Tran, N. T. L.; Mohan, S.; Xu, Z.; Huang, C.-H. Current Innovations and Future Challenges of Network Motif Detection. *Brief. Bioinform.* **2015**, *16* (3), 497–525. https://doi.org/10.1093/bib/bbu021.

(13) Fodor, J.; Brand, M.; Stones, R. J.; Buckle, A. M. Intrinsic Limitations in Mainstream Methods of Identifying Network Motifs in Biology. *BMC Bioinformatics* **2020**, *21*. https://doi.org/10.1186/s12859-020-3441-x.

(14) Proceedings of the Python in Science Conference (SciPy): Exploring Network Structure, Dynamics, and Function using NetworkX http://conference.scipy.org/proceedings/SciPy2008/paper_2/ (accessed 2021 -11 -11).

(15) Sedghamiz, H.; Morris, M.; Craddock, T. J. A.; Whitley, D.; Broderick, G. Bio-ModelChecker: Using Bounded Constraint Satisfaction to Seamlessly Integrate Observed Behavior With Prior Knowledge of Biological Networks. *Front. Bioeng. Biotechnol.* **2019**, *7*, 48. https://doi.org/10.3389/fbioe.2019.00048.

(16) Wang, P.; Lü, J.; Yu, X. Identification of Important Nodes in Directed Biological Networks: A Network Motif Approach. *PLoS ONE* **2014**, *9* (8), e106132. https://doi.org/10.1371/journal.pone.0106132.

(17) Cao, J.; Ding, C.; Shi, B. Motif-Based Functional Backbone Extraction of Complex Networks. *Phys. Stat. Mech. Its Appl.* **2019**, *526*, 121123. https://doi.org/10.1016/j.physa.2019.121123.

(18) Khammash, M. Reverse Engineering: The Architecture of Biological Networks. *BioTechniques* **2008**, *44* (3), 323–329. https://doi.org/10.2144/000112772.

(19) Marbach, D.; Schaffter, T.; Mattiussi, C.; Floreano, D. Generating Realistic in Silico Gene Networks for Performance Assessment of Reverse Engineering Methods. *J. Comput. Biol. J. Comput. Mol. Cell Biol.* **2009**, *16* (2), 229–239. https://doi.org/10.1089/cmb.2008.09TT.

(20) Camillo, B. D.; Toffolo, G.; Cobelli, C. A Gene Network Simulator to Assess Reverse Engineering Algorithms. *Ann. N. Y. Acad. Sci.* **2009**, *1158* (1), 125–142. https://doi.org/10.1111/j.1749-6632.2008.03756.x.