SOFTWARE TOOL ARTICLE

Network-based Observability and Controllability Analysis of Dynamical Systems: the NOCAD toolbox [version 1; peer review: 2 approved with reservations]

Dániel Leitold1,2, Ágnes Vathy-Fogarassy1,2, János Abonyi2

1Department of Computer Science and Systems Technology, University of Pannonia, Egyetem u. 10, Veszprém, 8200, Hungary
2MTA-PE Lendulet Complex Systems Monitoring Research Group, University of Pannonia, Egyetem u. 10, POB. 158, Veszprém, 8200, Hungary

Abstract
Network science has become increasingly important in life science over the last decade. The proposed Octave and MATLAB-compatible NOCAD toolbox provides a set of methods which enables the structural controllability and observability analysis of dynamical systems. In this paper, the functionality of the toolbox is presented, and the implemented functions demonstrated.

Keywords
Dynamical systems, Complex networks, Controllability and observability analysis, Robustness, MATLAB toolbox

Open Peer Review

Reviewer Status ✓ ?
Invited Reviewers
1 2

version 2
published
18 Sep 2019

version 1
published
09 May 2019

1 Gilles Didier1, University of Montpellier, Montpellier, France

2 Arthur Montanari1, Federal University of Minas Gerais (UFMG), Belo Horizonte, Brazil
ErcílioMoreira1, Federal University of Minas Gerais (UFMG), Belo Horizonte, Brazil
Luis A. Aguirre1, Federal University of Minas Gerais (UFMG), Belo Horizonte, Brazil

Any reports and responses or comments on the article can be found at the end of the article.
Corresponding author: János Abonyi (abonyij@fmt.uni-pannon.hu)

Author roles: Leitold D: Methodology, Software, Writing – Review & Editing; Vathy-Fogarassy Á: Formal Analysis, Methodology, Writing – Review & Editing; Abonyi J: Conceptualization, Methodology, Software, Writing – Review & Editing

Competing interests: No competing interests were disclosed.

Grant information: This research was supported by the National Research, Development and Innovation Office NKFIH, through the project OTKA-116674 (Process mining and deep learning in the natural sciences and process development) and the EFOP-3.6.1-16-2016-00015 Smart Specialization Strategy (S3) Comprehensive Institutional Development Program. Dániel Leitold was supported by the ÚNKP-18-3 New National Excellence Program of the Ministry of Human Capacities.

The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Copyright: © 2019 Leitold D et al. This is an open access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

How to cite this article: Leitold D, Vathy-Fogarassy Á and Abonyi J. Network-based Observability and Controllability Analysis of Dynamical Systems: the NOCAD toolbox [version 1; peer review: 2 approved with reservations] F1000Research 2019, 8:646 (https://doi.org/10.12688/f1000research.19029.1)

First published: 09 May 2019, 8:646 (https://doi.org/10.12688/f1000research.19029.1)
Introduction

In the life sciences, the determination of driver nodes in networks that play a significant role in the emergence or treatment of diseases is an intensively researched field. In large-scale human liver metabolic networks (HLMN), the driver metabolites have essential functions, and the role of transport reactions and extracellular metabolites in terms of controlling HLMN has revealed the importance of the environment of human liver metabolism with regard to the health of the liver.

In terms of controlling the human signalling network, the role of different proteins was also systematically analysed with the toolset of network controllability in 3 to highlight the role of cancer-associated genes. Target control with objective-guided optimisation (TCO) was introduced to control a set of variables (or targets) of interest while the quantity of drivers and constrained nodes were minimised and maximised, respectively. This method is capable of determining the leading phenotype transitions in biological networks that can be identified as drug targets. Using statistical analysis, a subset of critical control nonprotein-coding RNAs (ncRNAs) enriched by human disease can also be determined. In intra-cellular networks, to understand the information flow, a natural control system was utilised and the robustness of such a control was analysed.

The importance of determining the proper driver nodes in biological networks, more generally in any dynamical system, is unequivocal, and the amount of research concerning network science has increased rapidly. A detailed study about the control principles in biological networks has already been published. The network science-based analysis of dynamical systems has spread rapidly as it provides simple and efficient tools to analyse the structural controllability of any linear or linearised system.

Although considerable research has utilised the method, a flexible software tool which may be used to support the research in this field has yet to be designed. Parallel research has resulted in a collection of applications, toolboxes, plug-ins and scripts that analyse and determine several structural properties of genes, protein-protein interaction or even social or urban networks. Most of these applications only analyse the structural properties of static networks and just a handful of them utilise these structural properties to draw conclusions concerning the dynamics of the system investigated. As our toolbox belongs to the second group, in the following section, the available applications and programs of this group are elaborated.

A brief summary of the available tools with expanded functionalities is given in Table 1. Applications or software packages implemented in Python and capable of analysing the controllability and observability of dynamical systems are: graph-control and WDNfinder. The advantage of Python-based development lies in its widespread use and the countless methods and packages implemented in this language, including the tools developed for network analysis. Although in Python the focus is on developing a broad software package for complex system analysis, this has yet to be fulfilled and all of the available solutions have limitations. The graph-control toolbox only analyses the impact of network topology on the number of inputs and implements the fast matching algorithm. Even though WDNFinder only determines the minimum driver node set (MDS) and classifies nodes based on MDS, it is incapable of facilitating extended analysis.

Additionally, the CytoCtrlAnalyser plug-in for Cytoscape has been developed, which was implemented in Java and offers graphical interfaces for users as well. It evaluates control centrality, control capacity and classifies nodes for biomolecular networks. Furthermore, the Ecological Network Analysis with R software package (enaR) provides some dynamical analysis functions and can generate models to analyse ecological networks in the R environment. As can be seen, both software packages deal with special kinds of networks. The netctrl program can determine the driver nodes and switchboard dynamics model for any complex network. CONTEST is a MATLAB toolbox which can analyse the dynamics of complex systems, but these dynamics do not cover the structural controllability and observability properties of the analysed system. Although the presented software packages ensure the design of a controllable and observable system, they do not provide the opportunity to analyse the designed system exhaustively. These functions are helpful in terms of supporting the work of experts, but are insufficient for the sophisticated analysis of systems.

The contribution of this paper is to provide a novel toolbox, NOCAD, for the comprehensive analysis of linear or linearised systems based on their structural analysis.

Table 1. Toolboxes that implement some functions for dynamical analysis of complex systems based on their structural analysis.

<table>
<thead>
<tr>
<th>Software</th>
<th>Language</th>
<th>Applied on</th>
<th>GUI</th>
<th>Ref.</th>
<th>Last updated</th>
</tr>
</thead>
<tbody>
<tr>
<td>netctrl</td>
<td>C++</td>
<td>General networks</td>
<td>No</td>
<td>16</td>
<td>January 8, 2015</td>
</tr>
<tr>
<td>CONTEST</td>
<td>MATLAB</td>
<td>General networks</td>
<td>No</td>
<td>17</td>
<td>February, 2009</td>
</tr>
<tr>
<td>CytoCtrlAnalyser</td>
<td>Java</td>
<td>Biomolecular networks</td>
<td>Yes</td>
<td>13</td>
<td>May 25, 2017</td>
</tr>
<tr>
<td>graph-control</td>
<td>Python</td>
<td>General networks</td>
<td>No</td>
<td>9</td>
<td>December 16, 2015</td>
</tr>
<tr>
<td>WDNfinder</td>
<td>Python</td>
<td>Biological networks</td>
<td>No</td>
<td>10</td>
<td>June 24, 2018</td>
</tr>
<tr>
<td>enaR</td>
<td>R</td>
<td>Ecological networks</td>
<td>No</td>
<td>15</td>
<td>May 18, 2018</td>
</tr>
</tbody>
</table>
dynamical systems based on the approach of network science. In the following section, the implemented functions and measurements are presented through examples of their application.

**Methods**

With the help of the presented Octave- and MATLAB-compatible toolbox, experts can create, analyse and improve any type of dynamical systems. As the structure of the dynamical systems is generally represented by their adjacency matrix and linear dynamical systems can be described by the state-space model that contains the state-transition, input, output and feedthrough matrices, the Octave/MATLAB programming language is a perfect environment to handle these matrices and provide comprehensive functionalities based on them. With the use of NOCAD\(^1\), experts and researchers can effectively determine the input and output matrices of state-space models, calculate system-specific qualitative measurements (e.g. diameter, relative degree, control centrality and robustness of the system, etc.) and improve the system to satisfy the relative degree-based requirements. The workflow of the toolbox can be seen in Figure 1.

The functions of the toolbox can be performed step-by-step given its modular structure. Each module has a specific task and one function from each module calls the others. A system can be analysed by calling the main functions from the modules. The advantage of this structure is its modularity as each module can be expanded easily and further modules also implemented in a simple way. A list of their functions and dependencies on each other is presented in the manual.

**Implementation**

According to the aforementioned approach, the implemented functions of the toolbox were divided into three modules as follows: (1) network mapping module, (2) system characterisation module and (3) improvements and robustness module.

The network mapping module creates a dynamical system from a given network structure, i.e. the necessary matrices of the state-space model are generated for the topology in such a way, that the created system is structurally controllable and structurally observable. The determination of the input and output matrices can be achieved by the path finding and signal sharing methods\(^19\), which modify the result of the maximum matching algorithm.

The system characterisation module performs the calculation of 49 numerical measures to qualify the dynamical system based on its structure. The implemented measures, on the one hand, are well-known static measures (e.g. the number of nodes and edges, closeness and betweenness centralities), and, on the other hand, measures that characterise the dynamics of the system (e.g. structural controllability, observability, control centrality and relative degree). This module can also be used for the purpose of simple network analysis.

The improvement and robustness module integrates two main functions. On the one hand, it enables the input and output configurations of the system to be extended in such a way that the relative degree of the modified system does not exceed the initially defined threshold. For this purpose, this module implements five methods, namely the set covering-based grassroots and retrofit methods\(^20\), the centrality measures-based method\(^20\), the modified Clustering Large Applications based on Simulated Annealing algorithm (mCLASA), and the Geodesic Distance-based Fuzzy c-Medoid Clustering with Simulated Annealing algorithm (GDFCMSA)\(^20,21\). On the other hand, this module allows users to examine the robustness of the extended configurations by removing nodes from the network representation and by checking the structural controllability and structural observability of the damaged system.

The implemented methods are introduced in detail in the cited articles and the manual of the NOCAD toolbox.

**Figure 1. Workflow of the utilisation of the NOCAD toolbox.** The network mapping module provides two methods to create a dynamical system based on the topology of the state variables. The system characterisation module generates more than 49 measures to analyse, classify and characterise the developed system. The improvement and robustness module offers five algorithms to improve the system with additional inputs (observers) as well as outputs (controllers), and can analyse the robustness of the designed system.
Operation
In order to use the NOCAD toolbox\(^1\), installation of Octave or MATLAB is required. Then the directories of the toolbox must be copied into the working directory, or the directories of the toolbox must be added to the paths. The functions were implemented in Octave 5.1.0 and MATLAB R2016a on a Windows 64-bit system. On other operating systems, or with other Octave or MATLAB versions, proper operation is not guaranteed. Our toolbox is independent of other MathWorks toolboxes, it uses only the octave-networks-toolbox\(^2\) and the greedy set covering implementation\(^3\).

Use cases
In this section, the main functionalities of the NOCAD toolbox\(^1\) presented through examples of use cases. Although many biological networks are available from public databases, due to their complex nature, they are unsuitable for such a simple illustration. Therefore, the services of the NOCAD toolbox are presented on simple artificial networks.

The first step in each workflow is to create a state-space model from the adjacency matrix that presents the structural description of the system. This can be achieved by the use of path finding and signal sharing methods implemented in the first module. Both methods are modified versions of the maximum matching algorithm. An example of the application of the path finding method for the creation of a state-space model from the adjacency matrix (A) is shown in Figure 2. In this figure, B denotes the resulting input matrix, C the output matrix, while D stands for representing the direct feedthrough.

As the configuration above is not complex enough to demonstrate the functions of the second module, a more complex configuration of the input and output nodes is used. The sample input and output configurations can be seen in Figure 3, where the input and the output nodes are denoted by blue and red, respectively.

The system presented in Figure 3 consists of 9 state variables and 15 directed connections between them. Quality measures calculated by the System characterisation module of the NOCAD toolbox can be seen in Figure 4, Figure 5, and Figure 6.

In Figure 4, measures qualifying the whole system with one value are presented. The density shows that the number of edges

\[ A = \begin{bmatrix} 0 & a_{12} & a_{13} & 0 & 0 & 0 & 0 & 0 & 0 \\ a_{21} & 0 & 0 & a_{24} & a_{25} & 0 & 0 & 0 & 0 \\ 0 & a_{32} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & a_{45} & a_{46} & 0 & 0 \\ 0 & 0 & 0 & a_{54} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & a_{64} & 0 & 0 & 0 & a_{68} & 0 \\ 0 & 0 & 0 & a_{74} & a_{75} & a_{76} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & a_{97} & 0 & 0 \\ a_{8} & 0 & 0 & 0 & 0 & 0 & 0 & a_{97} & 0 \end{bmatrix}, 

\[ B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, 

\[ C = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, 

\[ D = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. 

Figure 2. An example network with determined input (blue) and output (red) configurations according to the path finding method. The network represents the A state transition matrix. B denotes the input matrix in which the places of the nonzero elements are determined by the controller node allocation algorithm. Similarly, the C output matrix is defined with the observability analysis of the network of the state variables. The D matrix of the direct feedthrough contains only zeros.

Figure 3. The complex configuration of the input and output nodes used for the demonstration of the system characterisation module of the NOCAD toolbox.
Figure 4. The example network with system and node centrality measures.

Figure 5. Measures of node clustering and the representation of the topology of the network.
is almost a fifth of the possible maximum, and the diameter of the system (i.e. the longest shortest path in the network that presents its structure) is 4. The degree variance is 2.67, while the Freeman’s centrality is 0.43. The relative degree of the system is also 4. The Pearson coefficient shows that the in-in and in-out correlations are assortative in nature, while out-out and out-in correlations are likely to be disassortative. The system is controllable and observable. As no loop is present in the network, the percentage of loops relative to edges is 0%. As there are 6 edges that have symmetric edge pairs and the number of connections is 15, the percentage of the symmetric edge pairs relative to the edges is 40%.

Node centrality measures assigned to the state variables of the system are also presented in Figure 4. One of the most important values is the highest degree of the nodes, which belongs to state variable $x_4$. As Scott’s centrality is a normalised degree, the most important node is once again $x_4$. The closeness of node $x_i$ is calculated as the ratio of the number of nodes reachable from $x_i$ to the sum of their distances from $x_i$. The higher value indicates the more central position of the node, and, once again, node $x_4$ is the most central element. The betweenness centrality shows how many shortest paths intercept the given node. If a node has a high value, then it is a critical node in the structure. The highest value belongs to nodes $x_2$ and $x_4$. The PageRank assigns a percentage value for each node, based on their centrality roles if Markov-chains are modelled. The measure referred to as correlation shows the proportion of the number of edges of neighbours’ and the number of neighbours. This information is useful when determining the assortativity of the system. The control centrality and observe centrality measures determine how many state variables can be influenced or observed by the nodes.

In Figure 5, the first vectors (referred to as driver and sensor nodes) show the driver and sensor nodes as logical vectors. The following four vectors classify these nodes as source, external, internal and inaccessible driver and sensor nodes. These types of nodes are introduced in 24 in detail. In the next section of the figure, the controlling and observing matrices are presented. Generally, these matrices are sparse matrices, as only the columns of drivers and sensors contain nonzero values. In Figure 5, we converted them into row vectors for their appropriate visualisation. The values show the number of derivations necessary to influence or observe a state variable in the system. Next, the similarity of the driver and sensor nodes is presented. The similarity of driver nodes $x_4$ and $x_6$ is 0.81. In this case, the reason why it is less than 1 is that although they control the same set of nodes, the numbers of derivations that influence them are different. In terms of sensor similarity, sensor nodes $x_2$ and $x_3$...
observe the same set of nodes and they do this almost simultaneously, so their similarity is 0.91. \( R_e \) and \( R_o \) are the simple reachability matrices. They show which nodes can be controlled or observed by a given node. In \( R_e \), the \( i \)th column shows which nodes can control node \( i \). From the other viewpoint, elements in row \( i \) highlight those nodes which can be controlled by node \( i \). In this example, node \( x_1 \) can influence every node, but it does not guarantee structural controllability. The \( R_o \) matrix can be interpreted analogously with regard to observability.

Finally, measures of edge centrality are seen in Figure 6. The betweenness has the same meaning as in the case of nodes, that is, it yields the number of shortest paths that intercept the edge\(^\text{23} \). From this perspective, the most critical edge is the edge \( a_{25} \) with a value of 10. The endpoint similarity shows how similar the influenced and observed sets of the state variables with regard to the endpoints of edges are. This metric has a high value if the edge is part of a cycle or creates a bridge in the network. As no bridges are present in this network, only cycles can be recognised by this measure. The edge similarity shows how similar the roles of edges are, and it allows redundancies, to be located. In the topology presented, nodes \( x_1, x_2, x_3 \), or nodes \( x_4, x_5, x_6 \) and \( x_7 \) also create parts of the network that possess redundancy.

For the demonstration of the last module, configurations provided by the first module are used again (Figure 2). Results provided by this module can be seen in Figure 7. In this case, five methods were applied to the system to extend the configuration as follows: the required relative degree was set at 2, while the alpha parameter of the cost function was set at 0.5\(^\text{21} \). Results show that all the methods determine the same set of driver nodes for the system, that is, they are sufficient to influence state variables \( x_4 \) and \( x_8 \). The resultant cost is 1.5565, the relative degree is 2 which satisfies the requirements, and the mean of the relative degrees is 1.1111. In this configuration, six different nodes can be identified which can be damaged separately and the system

![Diagram](image)

**Figure 7.** Improvement and robustness analysis of the system.

<table>
<thead>
<tr>
<th>Set Covering grassroot</th>
<th>Set Covering retrofit</th>
<th>Centrality Measures</th>
<th>mCLAS</th>
<th>GFCMSA</th>
</tr>
</thead>
<tbody>
<tr>
<td>driver nodes:</td>
<td>[4, 8]</td>
<td>[4, 8]</td>
<td>[4, 8]</td>
<td>[4, 8]</td>
</tr>
<tr>
<td>size:</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>cost:</td>
<td>1.5556</td>
<td>1.5556</td>
<td>1.5556</td>
<td>1.5556</td>
</tr>
<tr>
<td>max:</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>mean:</td>
<td>1.1111</td>
<td>1.1111</td>
<td>1.1111</td>
<td>1.1111</td>
</tr>
<tr>
<td>robustness:</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>robustness (%):</td>
<td>0.6667</td>
<td>0.6667</td>
<td>0.6667</td>
<td>0.6667</td>
</tr>
<tr>
<td>critical nodes:</td>
<td>[2, 4, 7]</td>
<td>[2, 4, 7]</td>
<td>[2, 4, 7]</td>
<td>[2, 4, 7]</td>
</tr>
<tr>
<td>sensor nodes:</td>
<td>[2, 7, 9]</td>
<td>[1, 4, 9]</td>
<td>[1, 7, 9]</td>
<td>[1, 6, 9]</td>
</tr>
<tr>
<td>size:</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>cost:</td>
<td>1.4444</td>
<td>1.3889</td>
<td>1.3889</td>
<td>1.3889</td>
</tr>
<tr>
<td>max:</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>mean:</td>
<td>0.8889</td>
<td>0.7778</td>
<td>0.7778</td>
<td>0.7778</td>
</tr>
<tr>
<td>robustness:</td>
<td>6</td>
<td>6</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>robustness (%):</td>
<td>0.6667</td>
<td>0.6667</td>
<td>0.7778</td>
<td>0.6667</td>
</tr>
<tr>
<td>critical nodes:</td>
<td>[1, 2, 6]</td>
<td>[1, 6, 9]</td>
<td>[1, 6]</td>
<td>[1, 6, 9]</td>
</tr>
</tbody>
</table>
remains controllable. This is expressed by the value of robustness (66.6%). The most important nodes in terms of controllability are \( x_2, x_3 \), and \( x_4 \). In the case of observability, methods yield different solutions with the exception of the centrality measures-based and mCLASA algorithms which provide the best configuration in this case. Although the cost as well as the maximum and mean of the relative degree were identical in the case of retrofit set covering-based and GDFCMA methods as well, the robustness analysis of these configurations exhibits a higher degree of vulnerability.

**Conclusions**

In this article the Octave- and MATLAB-compatible NOCAD toolbox\(^1\) was proposed to support the network-based controllability and observability analysis of dynamical systems. The toolbox offers two methods to design a structurally controllable and observable system based on the state-transition matrix. The designed system can be analysed by 49 qualitative measures both from structural and dynamical points of view. The toolbox serves five methods to improve the designed system by adding new inputs and outputs to it, thus, its relative degree can be decreased. Then the robustness of the individual designs can also be evaluated. The modular structure of the toolbox supports the facile improvement of the modules by adding new functions and the toolbox can be extended by new modules as well. Even though the modules are built on each other, most of their functions can also be used independently from each other.

Although our goal in this paper is to draw the attention of researchers of life sciences to the services provided by the NOCAD toolbox, it can be utilised in practice in various fields of sciences as well, for example, it enables social networks to be controlled in the economy, transaction networks to be analysed in finance or dynamical systems to be designed in engineering.

**Data availability**

All data underlying the results are available as part of the article and no additional source data are required.

**Software availability**

Source code available from: https://github.com/abonyilab/NOCAD.

Archived source code at time of publication: https://doi.org/10.5281/zenodo.2656674\(^8\)

**License**: GNU General Public License v3.0

**Author contributions**

Dániel Leitold reviewed the literature on network science, developed the algorithms, implemented the Octave and MATLAB functions, designed as well as performed the experiments, and wrote the related sections. Ágnes Vathy-Fogarassy participated in the formalisation of the methodology. János Abonyi developed the algorithms, implemented the Octave and MATLAB functions and proofread the paper.

**Grant information**

This research was supported by the National Research, Development and Innovation Office NKFIH, through the project OTKA-116674 (Process mining and deep learning in the natural sciences and process development) and the EFOP-3.6.1-16-2016-00015 Smart Specialization Strategy (S3) Comprehensive Institutional Development Program. Dániel Leitold was supported by the ÚNKP-18-3 New National Excellence Program of the Ministry of Human Capacities.

The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

---

**References**


Open Peer Review

Current Peer Review Status: 

Version 1

Reviewer Report 25 June 2019

https://doi.org/10.5256/f1000research.20856.r48912

© 2019 Aguirre L et al. This is an open access peer review report distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Arthur Montanari
Graduate Program in Electrical Engineering, Federal University of Minas Gerais (UFMG), Belo Horizonte, Brazil

Ercílio Moreira
Graduate Program in Electrical Engineering, Federal University of Minas Gerais (UFMG), Belo Horizonte, Brazil

Luis A. Aguirre
Graduate Program in Electrical Engineering, Federal University of Minas Gerais (UFMG), Belo Horizonte, Brazil

In this paper the authors describe the NOCAD toolbox developed for the analysis of some aspects of networks as, for instance, structural controllability and observability. The topic of the paper is not only very interesting but also timely as issues relating to controllability and observability of networks could be fundamental in a number of practical situations. Hence to have a nice set of tools to analyze dynamical networks is welcome.

The toolbox described is useful in a specific range of problems. Here we focus on the controllability and observability properties of networks, as prompted by the title. The tools presented focus on structural controllability and observability of linear systems. The assumption of linearity is not mentioned in the abstract. As a matter of fact, the opening phrase of the last paragraph in the introduction should be copied to the abstract.

The nonlinear case and other definitions of controllability and observability, such as, dynamical and symbolical are not mentioned in the paper nor are handled by the toolbox. This is an important remark as it is now known that some algorithms underestimate the cardinality of the set of sensor nodes when applied to nonlinear systems.

The paper

In the second paragraph of the Introduction, the authors mention the importance of “determining the proper driver nodes”. We wonder if the average reader would know what that is. In clarifying this point the
authors would like to address first what is “a” proper set of driving nodes (e.g. one that will guarantee full controllability). However, in order to specify “the” proper set, possibly some more detailed measure of controllability should be employed. For instance, suppose two sets of driving nodes S1 and S2, both with the same cardinality, such that the network is fully controllable either from S1 or from S2. Hence S1 is “a” proper set of driving nodes (assuming that by proper the authors are referring to full controllability) and S2 is another one. Now, it could well be that using S1 less energy is required to drive the network from one state \( x(t_i) \) to another \( x(t_f) \) when compared to S2. In that case, S1 and S2 are not totally equivalent. Structural controllability and observability are unable on their own to provide this distinction.

Following the same vein, we wonder if the average reader of F1000Research would know the distinction of static and dynamic networks. A word about this would be profitable, especially because the toolbox refers mainly to the second class.

In some parts of the paper the authors refer to matrix A as the “state transition” matrix. In continuous-time this is incorrect. In discrete-time this is only correct for a transition time of one sampling period. Matrix A is called the “dynamical matrix”. State transition matrix is something else.

**Relevance to the journal**

The MS introduction is written in such a way that it instigates interest from the target audience of this journal (mathematical biology). However, the techniques implemented in the toolboxes are usually found around the network science community, which includes several other fields that range from power systems to social networks. To show more coherence to this journal, the authors should provide some example of application on a “real-world network” under a biological context. This is also interesting to show how the results provided by the toolbox can be useful to draw conclusions under a practical context other than a “toy problem” as presented.

**Background**

The MS content is not sufficient to understand the techniques implemented in the toolbox. We know that the paper does not aim at providing extensive background, but some could be helpful. A reader coming from a control theory background might expect the toolbox to provide results based on Kalman’s definition of controllability and observability. However, the toolbox is based on Liu and coworker’s maximum matching algorithm which is based, in turn, on Lin’s structural definition (1974) (which is not even mentioned in the MS). Which definition of controllability and observability the toolbox is based on should be crystal clear in the main text and abstract. Moreover, the notions of structural controllability and observability should be presented to the reader and how they interplay with the more well-known notion in Kalman’s sense (e.g. Lin’s definition is only a necessary condition for Kalman’s definition). Although we think that the definitions of controllability and observability should be mentioned on the main text, the authors provide some background on the implemented maximum matching algorithms in the toolbox. This, however, should be mentioned explicitly in the MS.

**Examples**

The paper furnishes examples to illustrate some of the features of the new toolbox. The network topology is the same in each example, which facilitates understanding and comparison. In some cases, more discussion would be welcome. For instance, in the last example (Figure 7) not all the indices are clearly defined to the user, for instance the critical nodes are given as \( x_2, x_4 \) and \( x_7 \). From the context it seems that if any of these nodes is lost, the network would become uncontrollable, but this is not directly stated.
In referring to Figure 5 the authors speak in terms of Reachability matrices: Rc and Ro. It is clear from Figure 5 that Rc shows which nodes have a path to node i (the ith column). As for Ro, it seems that we should look row-wise instead of column-wise, is that right? Anyhow, we do not think it is adequate to say that Rc shows which nodes can be controlled from another one. The word “control” is perhaps too general. We would suggest just to say that Rc shows which nodes can be reached from another one.

**Toolbox**

The toolbox is subdivided in three modules. The first one, “network mapping module”, implements the maximum matching algorithm (and related modifications) to return a structurally controllable (observable) network with the smallest set of driver (sensor) nodes. This is a welcome feature, especially for a MATLAB environment, which we are unaware of any alternative.

The second module, “system characterization module”, provides several graph measures that are available on other MATLAB-based toolboxes, but are indeed useful to assess the network controllability and observability properties of a system. Thus, its presence is justifiable.

The third module, “improvements and robustness module”, is a set of functions which specifically implement previous results of the authors (e.g. Refs. 20, 21). It seems quite specific, but nevertheless the toolbox relevance is justifiable in great part for its module one.

The toolbox seems fast and no bugs were found in its implementations as far as the MATLAB environment is concerned. See further comments on some compatibility issues when using Octave.

**The target audience of the toolbox**

The toolbox is applicable to any kind of network (graph), be it directed or undirected, weighted or unweighted, and so on. However, although general, the authors should discuss in the MS what are the kinds of networks where a structural analysis of controllability and observability are more useful. For instance, a linearization of a power system model modelled by interconnected Kuramoto oscillators yield a dynamical matrix “A” whose corresponding adjacency graph is not only highly connected but also undirected. Consequently, the toolbox points out that only one driver node is needed to structurally control the network, independently of its size “n”(which is true in this case according to Lin’s definition). However, this does not give insight to the problem since basically any node can be chosen as a driver node, and only one node being sufficient seems quite unrealistic. The question is: For what kind of networks is the structural approach more interesting (and hence the toolbox)? It could be that the toolbox could be extra-helpful in the context of more sparse and directed networks (with higher hierarchy).

**Some nitpicking:**

Sometimes it is not clear whether the authors refer to the dynamical matrix “A” or the adjacency matrix “A^T”. We recommend that different nomenclatures be used, such as “A” for the dynamical matrix and “A_{Adj}” for the adjacency matrix. This should be changed in the MS and manual.

Regarding the installation of the NOCAD toolbox. We noticed that the NOCAD already comes with a “octave-network-toolbox” folder which does not have all the necessary functions to use Module 2. Thus, we had to download the “octave-network-toolbox-master” folder in Ref. 22 to have access to all needed functions. Is this necessary or is the NOCAD toolbox really missing some functions?
Some statements in Section “Use Cases”, Paragraph 5, are redundant with the information already present in Fig. 4. Instead of repeating the same information, it might be more interesting to make some comments on how useful some of these network measures can be to design better and more robust networks from a control and observation point-of-view.

There are some issues in the Octave version of this toolbox. For instance, function “heatmaps” has some bugs in Octave but works well in MATLAB. This happens because the function “colormap” from Octave accepts the argument “hot” but not “Hot”. The authors should do a careful review of the Octave toolbox and check for further compatibility issues.

References

Is the rationale for developing the new software tool clearly explained?
Partly

Is the description of the software tool technically sound?
Partly

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?
Partly

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?
Partly

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?
Partly

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Control theory, control of networked, nonlinear dynamics.

We confirm that we have read this submission and believe that we have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however we have significant reservations, as outlined above.

Janos Abonyi, University of Pannonia, Veszprém, Hungary
Dear Reviewers,

We are grateful for your useful remarks. In the following, we provide a detailed report about how we improved the paper based on your valuable comments and suggestions. We hope that the modifications have significantly improved the understandability of the paper.

Sincerely yours,
Janos Abonyi

In this paper the authors describe the NOCAD toolbox developed for the analysis of some aspects of networks as, for instance, structural controllability and observability. The topic of the paper is not only very interesting but also timely as issues relating to controllability and observability of networks could be fundamental in a number of practical situations. Hence to have a nice set of tools to analyze dynamical networks is welcome.

The toolbox described is useful in a specific range of problems. Here we focus on the controllability and observability properties of networks, as prompted by the title. The tools presented focus on structural controllability and observability of linear systems. The assumption of linearity is not mentioned in the abstract. As a matter of fact, the opening phrase of the last paragraph in the introduction should be copied to the abstract.

The nonlinear case and other definitions of controllability and observability, such as, dynamical and symbolical are not mentioned in the paper nor are handled by the toolbox. This is an important remark as it is now known that some algorithms underestimate the cardinality of the set of sensor nodes when applied to nonlinear systems.

Thank you for this invaluable remark, we have emphasised in both the Abstract and Introduction that the toolbox is applicable to the analysis of linear and linearized systems rather than nonlinear systems for which incorrect results may be suggested.

The paper

In the second paragraph of the Introduction, the authors mention the importance of “determining the proper driver nodes”. We wonder if the average reader would know what that is. In clarifying this point the authors would like to address first what is “a” proper set of driving nodes (e.g. one that will guarantee full controllability). However, in order to specify “the” proper set, possibly some more detailed measure of controllability should be employed. For instance, suppose two sets of driving nodes S1 and S2, both with the same cardinality, such that the network is fully controllable either from S1 or from S2. Hence S1 is “a” proper set of driving nodes (assuming that by proper the authors are referring to full controllability) and S2 is another one. Now, it could well be that using S1 less energy is required to drive the network from one state x(ti) to another x(tf) when compared to S2. In that case, S1 and S2 are not totally equivalent. Structural controllability and observability are unable on their own to provide this distinction.

Thank you for your valuable remark, we have expanded the relevant part of the paper and emphasized that during optimisation the cardinality and energy demand should be minimised, while the configurations should provide structural controllability and observability.
Following the same vein, we wonder if the average reader of F1000Research would know the distinction of static and dynamic networks. A word about this would be profitable, especially because the toolbox refers mainly to the second class.

- We are grateful for this suggestion, in the Introduction we have clarified the nomenclature as well as system and network classes used in the manuscript.

In some parts of the paper the authors refer to matrix A as the “state transition” matrix. In continuous-time this is incorrect. In discrete-time this is only correct for a transition time of one sampling period. Matrix A is called the “dynamical matrix”. State transition matrix is something else.

- Thank you for this invaluable remark, we have corrected the incorrect nomenclature.

Relevance to the journal

The MS introduction is written in such a way that it instigates interest from the target audience of this journal (mathematical biology). However, the techniques implemented in the toolboxes are usually found around the network science community, which includes several other fields that range from power systems to social networks. To show more coherence to this journal, the authors should provide some example of application on a “real-world network” under a biological context. This is also interesting to show how the results provided by the toolbox can be useful to draw conclusions under a practical context other than a “toy problem” as presented.

- We are grateful for this useful suggestion. We have improved the Introduction and, with the aid of references, introduced the networks that are already utilised in network-based structural controllability and observability analysis. In addition, the example problem has been replaced by the frontal neural network of C. elegans, a well-known standard biological dataset.

Background

The MS content is not sufficient to understand the techniques implemented in the toolbox. We know that the paper does not aim at providing extensive background, but some could be helpful. A reader coming from a control theory background might expect the toolbox to provide results based on Kalman’s definition of controllability and observability. However, the toolbox is based on Liu and coworker’s maximum matching algorithm which is based, in turn, on Lin’s structural definition (1974) (which is not even mentioned in the MS). Which definition of controllability and observability the toolbox is based on should be crystal clear in the main text and abstract. Moreover, the notions of structural controllability and observability should be presented to the reader and how they interplay with the more well-known notion in Kalman’s sense (e.g. Lin’s definition is only a necessary condition for Kalman’s definition).

Although we think that the definitions of controllability and observability should be mentioned on the main text, the authors provide some background on the implemented maximum matching algorithms in the toolbox. This, however, should be mentioned explicitly in the MS.

- We are grateful for your useful remarks and have improved the section entitled Methods, moreover, extended it with the theoretical background to dynamical systems, definitions of structural controllability as well as observability, and the applied methods. The proposed works from the literature have been cited in the relevant sections and for this suggestion we would like to express our sincere gratitude.
Examples

The paper furnishes examples to illustrate some of the features of the new toolbox. The network topology is the same in each example, which facilitates understanding and comparison. In some cases, more discussion would be welcome. For instance, in the last example (Figure 7) not all the indices are clearly defined to the user, for instance the critical nodes are given as $x_2$, $x_4$ and $x_7$. From the context it seems that if any of these nodes is lost, the network would become uncontrollable, but this is not directly stated.

In referring to Figure 5 the authors speak in terms of Reachability matrices: $R_c$ and $R_o$. It is clear from Figure 5 that $R_c$ shows which nodes have a path to node $i$ (the $i$th column). As for $R_o$, it seems that we should look row-wise instead of column-wise, is that right? Anyhow, we do not think it is adequate to say that $R_c$ shows which nodes can be controlled from another one. The word “control” is perhaps too general. We would suggest just to say that $R_c$ shows which nodes can be reached from another one.

- We are grateful for your valuable remark and have expressed the results provided by the toolbox as well as inserted a more thorough introduction to the results. According to the reachability matrices, control and reach derived from several dynamical systems overlap to some extent when the problem with regard to controllability was reduced to one concerning reachability.

Toolbox

The toolbox is subdivided in three modules. The first one, “network mapping module”, implements the maximum matching algorithm (and related modifications) to return a structurally controllable (observable) network with the smallest set of driver (sensor) nodes. This is a welcome feature, especially for a MATLAB environment, which we are unaware of any alternative.

The second module, “system characterization module”, provides several graph measures that are available on other MATLAB-based toolboxes, but are indeed useful to assess the network controllability and observability properties of a system. Thus, its presence is justifiable.

The third module, “improvements and robustness module”, is a set of functions which specifically implement previous results of the authors (e.g. Refs. 201, 212). It seems quite specific, but nevertheless the toolbox relevance is justifiable in great part for its module one.

The toolbox seems fast and no bugs were found in its implementations as far as the MATLAB environment is concerned. See further comments on some compatibility issues when using Octave.

- We are grateful for your useful remarks and have highlighted in the section entitled Implementation the importance of the third module for dynamical systems in general and biological systems specifically.

The target audience of the toolbox

The toolbox is applicable to any kind of network (graph), be it directed or undirected, weighted or
unweighted, and so on. However, although general, the authors should discuss in the MS what are the kinds of networks where a structural analysis of controllability and observability are more useful. For instance, a linearization of a power system model modelled by interconnected Kuramoto oscillators yield a dynamical matrix “A” whose corresponding adjacency graph is not only highly connected but also undirected. Consequently, the toolbox points out that only one driver node is needed to structurally control the network, independently of its size “n”(which is true in this case according to Lin’s definition). However, this does not give insight to the problem since basically any node can be chosen as a driver node, and only one node being sufficient seems quite unrealistic. The question is: For what kind of networks is the structural approach more interesting (and hence the toolbox)? It could be that the toolbox could be extra-helpful in the context of more sparse and directed networks (with higher hierarchy).

• Thank you for your valuable remark. We believe no “better” types of networks for structural analysis are known since the importance of a node in a complex system from a structural point of view can be excessive or critical with regard to reliability, independent of the directedness in the network. The hardest aspect of these analyses is the evaluation of the results that should be provided by experts, moreover, this may cause undirected or directed and sparser or denser networks to be preferred.

Some nitpicking:

Sometimes it is not clear whether the authors refer to the dynamical matrix “A” or the adjacency matrix “A^T”. We recommend that different nomenclatures be used, such as “A” for the dynamical matrix and “A_{Adj}” for the adjacency matrix. This should be changed in the MS and manual.

• Thank you for this invaluable remark, we have used the nomenclature “A^T” to emphasise the transposition, i.e. the connection between the dynamical and adjacency matrices.

Regarding the installation of the NOCAD toolbox. We noticed that the NOCAD already comes with a “octave-network-toolbox” folder which does not have all the necessary functions to use Module 2. Thus, we had to download the “octave-network-toolbox-master” folder in Ref. 22 to have access to all needed functions. Is this necessary or is the NOCAD toolbox really missing some functions?

• We are grateful for the thorough testing of the toolbox. Unfortunately, some function from the octave-networks-toolbox was excluded during its upload, but this has been corrected.

Some statements in Section “Use Cases”, Paragraph 5, are redundant with the information already present in Fig. 4. Instead of repeating the same information, it might be more interesting to make some comments on how useful some of these network measures can be to design better and more robust networks from a control and observation point-of-view.

• Thank you for your remark, we have replaced the example network, therefore, the diagrams have been removed and the relevant parts improved. We hope that this problem has been resolved.

There are some issues in the Octave version of this toolbox. For instance, function “heatmaps” has some bugs in Octave but works well in MATLAB. This happens because the function “colormap” from Octave accepts the argument “hot” but not “Hot”. The authors should do a careful review of the Octave toolbox and check for further compatibility issues.
Thank you for your remark. Interestingly, our system did not alert us to this problem. We have corrected the source code as suggested since both parameters were applied to our configurations, however, we kindly ask you to check your version of Octave as the problem can also be caused by the use of an older version of this program.

**Competing Interests:** No competing interests were disclosed.

---

Reviewer Report 28 May 2019

https://doi.org/10.5256/f1000research.20856.r48269

© 2019 Didier G. This is an open access peer review report distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Gilles Didier
Institut Montpelliérain Alexander Grothendieck (IMAG), CNRS, University of Montpellier, Montpellier, France

This software tool article presents the Octave/MATLAB toolbox "NOCAD" for networks analysis. NOCAD implements various standard measures on networks and their nodes (density, betweenness, centrality...). Its main novelty stands in the "dynamical-systems-based" tools to analyse (static) networks that it provides. This article can be seen as a companion paper of the reference 19 from the same authors (It could be stated somewhere in the text).

The ideas implemented in the toolbox are interesting, but I feel that their presentation has to be improved. First, I suggest you get your text proof-read by a native speaker. Second, many concepts used in the manuscript should be at least briefly recalled. Third, the article lacks a detailed description of the input (in the general sense, including user interaction) of the toolbox. The basic input seems to be the incidence matrix of a directed network but, if I well understood, the toolbox allows to set input and output nodes and has network design capabilities.

Another major concern is that though one of the aims of the article is to draw attention of life science researchers, it does not present clearly how the toolbox can be used on biological networks and on which kind of biological networks it can be applied. Real biological examples should be actually more than welcome in the "Use cases" section, which contains only toy examples. Without such examples, I am afraid that the article fails to draw attention of its expected audience. In particular, I am wondering in what extent the toolbox can be used on standard biological datasets, e.g., PPI networks can have thousands vertices.

Please find below some detailed comments:

**Abstract**
The abstract has to be extended in order to give a better idea of the content of the article.

**Introduction**
The introduction does not reflect well what follows and the topic of the paper. You should start by briefly introducing the general ideas of dynamical analysis of networks, precise in what sense you use the term
"driver node", which is sometimes used in a biological context etc. First mention of dynamical system comes at the end of the second paragraph and could be developed (e.g. by briefly recalling ideas of [7]²) while biological examples may not need to be as much detailed.
The description of previous software sounds good. It may take place a dedicated section.

**Method**
The method section should recall the notions of controllability and observability and the general principles of `dynamical analysis` (it can be brief, something like the two first subsections of the section "Results" of [19]¹). In the current version, I don't think that this part can be understood without reading your reference [19]¹.
You should state explicitly which are the inputs of your toolbox, the accepted formats etc.
I found the paragraph about the modularity of the toolbox is not very useful. It could be shortened or even discarded.
Subsection "Operation" ("Installation"?) is OK.

**Use cases**
If I well understood, the input is the incidence matrix A from which are computed the matrices B, C and D as illustrated with the first example. It should be interesting to apply the approach on a biological network, for instance a protein-protein interaction network or a regulatory network, in order to illustrate the interest of the results obtained in this context.
In the second example (Figure 3), input and output nodes seem to be set arbitrarily in order to get a system complex-enough to show the function of the second module of the toolbox (incidence matrix is the same as that of the first example). Is it possible to illustrate the toolbox with an example in which the input and outputs are not fixed? I am actually concerned with the fact that the use cases do not really show how the toolbox can be used on a real dataset, which may not have natural input and outputs.
Figures 4, 5 and 6 displays the results of the toolbox on the second example. These results are briefly discussed in the text which also recalls the definitions of some of the measures. Applying the toolbox on a real dataset would make the discussion of the results of the toolbox much easier to interpret. Moreover, a real dataset could provide evidence of the relevance of the results obtained by relying on the nature of nodes distinguished by the toolbox.
The different types of nodes displayed in Figure 5 should be defined rather than just refer to [24]³.
Could you please give an idea about what kind of biological networks can be studied by the means of your third module? This part is not very intuitive from a biological point of view.

**References**

**Is the rationale for developing the new software tool clearly explained?**
Partly

**Is the description of the software tool technically sound?**
Partly
Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?
Partly

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?
No

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?
Partly

**Competing Interests:** No competing interests were disclosed.

**Reviewer Expertise:** My main area of research is applied mathematics. I worked on biological networks (regulatory and interaction networks).

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

---

**Reader Comment 10 Sep 2019**

Janos Abonyi, University of Pannonia, Veszprém, Hungary

Dear Reviewer,

We are grateful for your useful remarks.
In the following, we provide a detailed report about how we improved the paper based on your valuable comments and suggestions.
We hope that the modifications have significantly improved the understandability of the paper.

Sincerely yours,
Janos Abonyi

This software tool article presents the Octave/MATLAB toolbox "NOCAD" for networks analysis. NOCAD implements various standard measures on networks and their nodes (density, betweenness, centrality...). Its main novelty stands in the "dynamical-systems-based" tools to analyse (static) networks that it provides. This article can be seen as a companion paper of the reference 19 from the same authors (It could be stated somewhere in the text).

- Thank you for your valuable remark. We have cited the works related to the functions of the toolbox. The sections entitled *Introduction* and *Methods* have been rewritten and improved in order to highlight the applicability of the toolbox in the life sciences and provide additional contribution.

The ideas implemented in the toolbox are interesting, but I feel that their presentation has to be improved. First, I suggest you get your text proof-read by a native speaker. Second, many concepts used in the manuscript should be at least briefly recalled. Third, the article lacks a detailed description of the input (in the general sense, including user interaction) of the toolbox.
The basic input seems to be the incidence matrix of a directed network but, if I well understood, the toolbox allows to set input and output nodes and has network design capabilities.

- Thank you for this remark. We used Grammarly and asked a professional proofreader to correct the second version of the manuscript. We have tried our best to minimize the number of grammatical errors.

- At the beginning of the section entitled Methods, the representation of linear systems, definitions of structural controllability as well as observability, and introductions to the methodology in addition to maximum matching were included, therefore, a review of the cited literature was deemed unnecessary. The inputs of the modules were clarified in the section entitled Implementation.

Another major concern is that though one of the aims of the article is to draw attention of life science researchers, it does not present clearly how the toolbox can be used on biological networks and on which kind of biological networks it can be applied. Real biological examples should be actually more than welcome in the "Use cases" section, which contains only toy examples. Without such examples, I am afraid that the article fails to draw attention of its expected audience. In particular, I am wondering in what extent the toolbox can be used on standard biological datasets, e.g., PPI networks can have thousands vertices.

- We are grateful for your valuable remark and have cited the latest articles to highlight the importance of the methodology in the field of the life sciences. In the section entitled Use cases, the example network was replaced by a frontal neural network of C. elegans from the standard biological dataset that has been comprehensively analysed and is sufficiently well-known to be regarded as an interesting example for scientists from the fields of the life as well as network sciences.

Please find below some detailed comments:

**Abstract**
The abstract has to be extended in order to give a better idea of the content of the article.

- Thank you for this remark. We have improved the abstract based on the reviews and extended it with the information provided to develop the introduction with regard to the content and the main goal of the paper, namely the applicability of the methodology to the life sciences.

**Introduction**
The introduction does not reflect well what follows and the topic of the paper. You should start by briefly introducing the general ideas of dynamical analysis of networks, precise in what sense you use the term "driver node", which is sometimes used in a biological context etc. First mention of dynamical system comes at the end of the second paragraph and could be developed (e.g. by briefly recalling ideas of [7][2]) while biological examples may not need to be as much detailed.

- We are grateful for your useful remarks. We have overviewed the Introduction and rewritten it to draw attention to the importance of the methodology in the life sciences. Furthermore, in the section entitled Methods, we introduced the theoretical background to network-based structural controllability and observability analysis.

The description of previous software sounds good. It may take place a dedicated section.

- Thank you for this remark, we decided to dedicate a separate section to the existing software.

**Method**
The method section should recall the notions of controllability and observability and the general
principles of "dynamical analysis” (it can be brief, something like the two first subsections of the section "Results" of [19][1]). In the current version, I don’t think that this part can be understood without reading your reference [19][1].
  • We are grateful for this useful suggestion. As previously mentioned, we have included a theoretical introduction to dynamical systems and their controllability as well as observability in the section entitled Methods.

You should state explicitly which are the inputs of your toolbox, the accepted formats etc.
  • We are grateful for your remark. The introduction to dynamical systems emphasises the input of the toolbox, namely the dynamical matrix of the state-space representation. In addition, in the section entitled Implementation, each of the inputs to the modules was described separately.

I found the paragraph about the modularity of the toolbox is not very useful. It could be shortened or even discarded.
  • We are grateful for your valuable suggestion and have deleted this paragraph.

Subsection “Operation” (“Installation”?) is OK.

Use cases
If I well understood, the input is the incidence matrix A from which are computed the matrices B, C and D as illustrated with the first example. It should be interesting to apply the approach on a biological network, for instance a protein-protein interaction network or a regulatory network, in order to illustrate the interest of the results obtained in this context.
  • Thank you for your valuable remark. We have replaced the example network with the frontal neural network of C. elegans to draw attention to the easily understandable measures rather than the useful results. Therefore, the results from the analysis of the frontal neural network of C. elegans are presented, while in the manual the example network is still available.

In the second example (Figure 3), input and output nodes seem to be set arbitrarily in order to get a system complex-enough to show the function of the second module of the toolbox (incidence matrix is the same as that of the first example). Is it possible to illustrate the toolbox with an example in which the input and outputs are not fixed? I am actually concerned with the fact that the use cases do not really show how the toolbox can be used on a real dataset, which may not have natural input and outputs.
  • Since we replaced the example network with the frontal neural network of C. elegans, it is not necessary to apply the fixed inputs and outputs, therefore, we hope that this problem has been resolved.

Figures 4, 5 and 6 displays the results of the toolbox on the second example. These results are briefly discussed in the text which also recalls the definitions of some of the measures. Applying the toolbox on a real dataset would make the discussion of the results of the toolbox much easier to interpret. Moreover, a real dataset could provide evidence of the relevance of the results obtained by relying on the nature of nodes distinguished by the toolbox.
  • As the example network was changed, we hope this remark has been acted on.

The different types of nodes displayed in Figure 5 should be defined rather than just refer to [24][3].
  • Thank you for this invaluable remark. We have expanded this part of the paper and, with references, mentioned the importance of these types in biological networks.

Could you please give an idea about what kind of biological networks can be studied by the means of your third module? This part is not very intuitive from a biological point of view.
  • We are grateful for this useful suggestion. We have dedicated a paragraph to the importance of this module in the section entitled Implementation. Given the improved theoretical background, we hope that the presence of this module is more suitable.
Is the rationale for developing the new software tool clearly explained?
Partly
- As no widespread and approved tool for the network-based structural controllability and observability analysis of dynamical systems is known, we propose our toolbox for this purpose, and its applicability is introduced with the aid of a real network from a standard biological dataset.

Is the description of the software tool technically sound?
Partly
- We have improved the introduction to the theoretical background, therefore, the paper can be regarded as an independent work that introduces the advantages of the methodology and implemented toolbox.

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?
Partly
- We have stated the input used in this article and the toolbox, moreover, a well-known example was used in order of maximal reproducibility.

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?
No
- We have included a more detailed introduction to the theoretical background in order to provide sufficient information concerning the expected outputs and results.

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?
Partly
- We hope that the frontal neural network of C. elegans and its results represent well the applicability of the toolbox, and believe in the importance of such a novel toolbox, even though numerous publications utilise this methodology.

Competing Interests: No competing interests were disclosed.