Peer Community In
Mathematical &
Computational
Biology

Reprogramming of locally-monotone Boolean networks with BoNesis

Sergiu Ivanov *based on peer reviews by* ***Ismail Belgacem*** *and 1 anonymous reviewer*

Loïc Paulevé (2022) Marker and source-marker reprogramming of Most Permissive Boolean networks and ensembles with BoNesis. arXiv, ver. 2, peer-reviewed and recommended by Peer Community in Mathematical and Computational Biology. https://doi.org/10.48550/arXiv.2207.13307

Submitted: 31 August 2022, Recommended: 14 March 2023

Cite this recommendation as:
Ivanov, S. (2023) Reprogramming of locally-monotone Boolean networks with BoNesis. *Peer Community in Mathematical and Computational Biology*, 100183. https://doi.org/10.24072/pci.mcb.100183

Published: 14 March 2023

Reprogramming of cellular networks is a well known challenge in computational biology consisting first of all in properly representing an ensemble of networks having a role in a phenomenon of interest, and secondly in designing strategies to alter the functioning of this ensemble in the desired direction. Important applications involve disease study: a therapy can be seen as a reprogramming strategy, and the disease itself can be considered a result of a series of adversarial reprogramming actions. The origins of this domain go back to the seminal paper by Barabási et al. [1] which formalized the concept of network medicine.

An abstract tool which has gathered considerable success in network medicine and network biology are Boolean networks: sets of Boolean variables, each equipped with a Boolean update function describing how to compute the next value of the variable from the values of the other variables. Despite apparent dissimilarity with the biological systems which involve varying quantities and continuous processes, Boolean networks have been very effective in representing biological networks whose entities are typically seen as being on or off. Particular examples are protein signalling networks as well as gene regulatory networks.

The paper [2] by Loïc Paulevé presents a versatile tool for tackling reprogramming of Boolean networks seen as models of biological networks. The problem of reprogramming is often formulated as the problem of finding a set of perturbations which guarantee some properties on the attractors. The work [2] relies on the most permissive semantics [3], which together with the modelling assumption allows for considerable speed-up in the practically relevant subclass of locally-monotone Boolean networks.

The paper is structured as a tutorial. It starts by introducing the formalism, defining 4 different general variants of reprogramming under the most permissive semantics, and presenting evaluations of their complexity in terms of the polynomial hierarchy. The author then describes the software tool BoNesis which can handle different problems related to Boolean networks, and in particular the 4 reprogramming variants. The presentation includes concrete code examples with their output, which should be very helpful for future users.

The paper [2] introduces a novel scenario: reprogramming of ensembles of Boolean networks delineated by some properties, including for example the property of having a given interaction graph. Ensemble reprogramming looks particularly promising in situations in which the biological knowledge is insufficient to fully determine all the update functions, i.e. in the majority of modelling situations. Finally, the author also shows how BoNesis can be used to deal with sequential reprogramming, which is another promising direction in computational controllability, potentially enabling more efficient therapies [4,5].

References:

- Barabási A-L, Gulbahce N, Loscalzo J (2011) Network medicine: a network-based approach to human disease. Nature Reviews Genetics, 12, 56–68. https://doi.org/10.1038/nrg2918

- Paulevé L (2023) Marker and source-marker reprogramming of Most Permissive Boolean networks and ensembles with BoNesis. arXiv, ver. 2 peer-reviewed and recommended by Peer Community in Mathematical and Computational Biology. https://doi.org/10.48550/arXiv.2207.13307

- Paulevé L, Kolčák J, Chatain T, Haar S (2020) Reconciling qualitative, abstract, and scalable modeling of biological networks. Nature Communications, 11, 4256. https://doi.org/10.1038/s41467-020-18112-5

- Mandon H, Su C, Pang J, Paul S, Haar S, Paulevé L (2019) Algorithms for the Sequential Reprogramming of Boolean Networks. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 16, 1610–1619. https://doi.org/10.1109/TCBB.2019.2914383

- Pardo J, Ivanov S, Delaplace F (2021) Sequential reprogramming of biological network fate. Theoretical Computer Science, 872, 97–116. https://doi.org/10.1016/j.tcs.2021.03.013

Reviews

Evaluation round #2

Reviewed by anonymous reviewer 1, 31 January 2023

The authors has addressed my concerns. I am supportive of publication of this paper.

Reviewed by Ismail Belgacem, 20 February 2023

The authors addressed all my concerns. In particular, the paper is significantly better structured and the contribution have been clarified and improved. The author clearly answers all my previous comments, I appreciated the example added. I have no major issues with the revised manuscript. I have no further comments, and, in my opinion, the paper could now be published like that.

Evaluation round #1

DOI or URL of the preprint: https://doi.org/10.48550/arXiv.2207.13307
Version of the preprint: 1

Authors' reply, 24 January 2023

I thank the reviewers for their positive feedback and valuable suggestions. Please find below a point by point reply.

Review 1

> *Major:*
> *1. For the Most Permissive update regime, what is the biological/physical meaning of the any(*) state? Is it due to the >*
> *Boolean nature of the model, as the level of a node is changing you can intepret as either 1 or 0? What's the difference between the MP updage regime with a tri-state model(where the middle level is representing the transition from 0 to 1)?*

There is maybe a confusion on the meaning of the "*" value. In this paper, "*" is used to denote components that can take both 0 and 1 value in a sub-hypercube.
In the context of attractor, they specify components that can always oscillate between 0 and 1 in the MP attractor. We added this clarification in the "MP attractors are minimal trap spaces" section:
"Therefore, a component with a \$\$ value in an MP attractor \$A\$ indicates that the component that can always oscillate between 0 and 1 in the (cyclic) attractor."*

Regarding the encoding of MP, it depends on what the reviewer means by tri-state model. In case of classical multi-valued logical networks, they cannot model directly MP as we also need the information of whether the node is increasing or decreasing. And with 4 states (increasing/decreasing), they cannot be encoded in the classical neither as we need non-deterministic update functions and the 4 states cannot be ordered. This is addressed in the Nature communicating paper introducing MP [https://doi.org/10.1038/s41467-020-18112-5]
On another hand, in the Nature communication paper, we give an encoding of MP using 4-states automata networks with non-deterministic functions. But their semantics do not match with usual René Thomas/multi-valued logical models of the literature in systems biology.
This is a rather technical point that is out of the scope of this paper, and which has been discussed in the mentioned reference.

> *2. The authors shall discuss what kind of biological systems are locally monotone and what are not? So the biological communities can have a sense when the methodology here is relevant.*

Indeed, this information was missing.

We added in the introduction, after mentioning locally-monotone BNs:
"Locally-monotone BNs cover all the models where it assumed that a node cannot be both an activator and inhibitor of a same other node, which is a common assumption when modeling biological system."

and in the section "BoNesis":
*"We emphasize that *BoNesis* is currently restricted to locally monotone BNs only for which efficient logical encoding of domains of models are possible. Whereas it is a common assumption when modeling of biological systems (a node cannot be both an activator and inhibitor of a same other node), non-monotone BNs are also employed, and cannot be addressed with the current implementation."*

Minor

1. notorious in the first sentence. I think notorious is typically used for bad quality and deed.

Indeed, thank you very much for spotting this ”false friend”. It has been replaced with ”pertinent”.

2. For all python code, it would be preferred if Typing is provided. This will make the package user's life easier.

This has been added, thank you for this suggestion.

3. The paper blended the python snippt with the theoretical discussion. The format is quite noval. Some journals will consolidate these into an appendix. I don't have a preference. As long as the editor is happy with the format, I am supportive the current format.

The idea was to provide theoretical specification with practical implementation side by side, where code can be seen as equations. Moreover, because we distribute the paper as an interactive notebook, this allows to easily modify the examples to better understand the method.

4. I think this paper ”Target Control in Logical Models Using the Domain of Influence of Nodes” is relevant when you discuss BN modeling, application and control/reprogramming.

Indeed, the reference has been added in the introduction.

Review 2

The author takes at the beginning several examples (with small dimensions) that exhibit sustained oscillations behaviors. The oscillations accrued because negative feedback loops are included in models. Then, the reprogramming was the perturbations of the expression of at most k genes to suppress sustained oscillations of all the selected markers (stabilize the markers oscillating to desired states 1 or 0). See, for example, these papers:

I. I. Belgacem, J.-L. Gouzé, R. Edwards, Control of negative feedback loops in genetic networks, in: 2020 59th IEEE Conference on Decision and Control (CDC), IEEE, 2020, pp. 5098–5105.

II. L. Chambon, I. Belgacem, J.-L. Gouzé, Qualitative control of undesired oscillations in a genetic negative feedback loop with uncertain measurements, Automatica 112 (2020) 108642.

where a similar problem is investigated to suppress sustained oscillations for genetic negative feedback loops in general. The authors proposed in these papers a sliding mode control to suppress sustained oscillations for genetic negative feedback loops by controlling only the expression of one gene (which is physically possible).

Here, the author should include a more detailed explanation of the practical applicability of controlling the expression of many genes at the same time or in parallel and shows its (the control) potential interest for the control community.

We emphasized in the discussion that the controls are constant over time and independent of the state of the system.
”This paper focused on permanent perturbations, i.e., enforcing the value of one or several components constantly over time, independently of the state of the system.”

Moreover, we slightly extended the discussion around the sequential reprogramming to link with control theory:

"Sequential reprogramming brings the BN reprogramming settings closer to classical control theory, as the control can depend both on time and state of the system."

> *After, the author for example considers ten models with high dimensions (of up to 75) and for the reprogramming up to 6 simultaneous perturbations (k=6). Many solutions were suggested or found for reprogramming, however which ones have to be more feasible for the control implementation? Could we always find control of each function of some genes to be 0 or 1 (according to each equation)? Are there solutions that have very little practical relevance? Do all the solutions make sense physically?*

We added a precision in the discussion that the computation is based solely on the Boolean dynamics and do not integrate extra information on biological feasibility, when considering models of biological systems:

"It should be noted that the candidate combinations of perturbations are computed solely based on the Boolean dynamics, and do not account for experimental feasibility, e.g., in the scope of models of biological systems. Future work may consider optimization or prioritization of perturbations based on such extra information."

> *The author of this paper also considers only locally-monotone BNs, where each local function is unate. This means that the system conserves the partial ordering of solutions. Monotone systems also have strong properties of convergence towards equilibria. Is the proposition of locally-monotone functions simplified much for reprogramming? Furthermore, BoNesis is only supporting locally-monotone BNs... How about if we take general BNs?*

Locally monotone BNs are different than monotone BNs (which are a particular case). Whereas locally monotone BNs offer a smaller computational complexity than in general, they do not possess the same properties as monotone BNs.

We added clarifications at the end of the "Basic definition" section of the Background:

*"Locally monotone BNs should not be confused with *monotone* BNs where a component appears in *all* local functions with the same sign. Monotone BNs are a particular case of locally monote BNs."*

and in the section "BoNesis":

*"We emphasize that *BoNesis* is currently restricted to locally monotone BNs only for which efficient logical encoding of domains of models are possible. Whereas it is a common assumption when modeling of biological systems (a node cannot be both an activator and inhibitor of a same other node), non-monotone BNs are also employed, and cannot be addressed with the current implementation."*

> *1- The paper has some typos issue to be checked and corrected. For example, in the 13th page: to itrs right configuration, I believe that it should be: to its right configuration.*

This has been corrected, together with several other typos.

> *2- Similarly, in the top of the page 22: for some BNs they will have no fixed points, therefore all their fixed points match with the marker, I believe that the author would like to say for some BNs, they will have no other fixed points,*

The paragraph has been rephrased, with an additional example, to make the point clearer.

Download tracked changes file

Decision by Sergiu Ivanov, posted 15 December 2022, validated 19 December 2022

Minor revision

Dear Author,

The anonymous reviewers have read your submission and provided their comments.

Could you please take into account their corrections and suggestions for improvement?

Thank you for your work!

-

Best regards,
Sergiu

Reviewed by anonymous reviewer 1, 25 October 2022

The paper studied four variants of marker reprogramming problems for Boolean network under Most Permissive update regime and locally monotone Boolean functions. For each type of reprogramming problem, the author discussed the theoretical computational complexity and also provide a python framework to solve it computationally. For P3 and P4, the author solve the problem by solving its complementary problem with exhaustion. The paper is generally well written and would be interesting for the community to read. I am supportive of the publication if the following issues are addressed:

Major:

1. For the Most Permissive update regime, what is the biological/physical meaning of the any(*) state? Is it due to the Boolean nature of the model, as the level of a node is changing you can inteprete as either 1 or 0? What's the difference between the MP updage regime with a tri-state model(where the middle level is representing the transition from 0 to 1)?

2. The authors shall discuss what kind of biological systems are locally monotone and what are not? So the biological communities can have a sense when the methodology here is relevant.

Minor

1. notorious in the first sentence. I think notorious is typically used for bad quality and deed.

2. For all python code, it would be preferred if Typing is provided. This will make the package user's life easier.

3. The paper blended the python snippt with the theoretical discussion. The format is quite noval. Some journals will consolidate these into an appendix. I don't have a preference. As long as the editor is happy with the format, I am supportive the current format.

4. I think this paper ”Target Control in Logical Models Using the Domain of Influence of Nodes” is relevant when you discuss BN modeling, application and control/reprogramming.

Reviewed by Ismail Belgacem, 15 December 2022

The paper shows how to use BoNesis software for controlling Most Permissive (MP) Boolean networks (BNs). In general, the aim is to find parsimonious control inputs such that the system dynamics leave the current disease marker signature to necessarily reach the healthy marker signature (at steady states or attractors). First, note that the software BoNesis was developed by the same author of this paper (see Chevalier et al., 2019) for Most Permissive BNs synthesis. This kind of modeling (using Most Permissive BNs) permits to capture in addition the transient behaviors by considering an uncertainty region where the gene is active and able or not to regulate the gene expression of its targets (an area between off to on) or when the gene is inactive and able or not to control the gene transcription of its targets (a region between on to off).

Here, the author aims to show how BoNesis is also employed for identifying strategies for controlling fixed points and attractors of MP BNs. In particular, the control of fixed points and attractors starting from any initial conditions (globally) or only the fixed points and attractors beginning from a given initial state or configuration. The author considers that the desired targets (fixed points or attractors) are specified by a set of variables with fixed Boolean values (markers). Then, the author presents through examples how we use BoNesis to identify all the possible perturbations for reprogramming the fixed points or the attractors of BNs to be compatible with the selected markers. For each example, the author provides the python codes for the reprogramming.

A perturbation is freezing the local functions of some components (at a maximum number k) at constant Boolean values, which means the control in parallel of the expression of some genes to be is either active or inactive. If there is no attractor, the reprogramming also permits the identification of all the perturbations that will create such attractors and ensure their reachability by considering a maximum number of genes to control in parallel. The reprogramming also ensures that the selected markers are not oscillating if the attractors are stable limit cycles or cyclic attractors. I find the problem interesting. Very pleasant to read.

The author takes at the beginning several examples (with small dimensions) that exhibit sustained oscillations behaviors. The oscillations accrued because negative feedback loops are included in models. Then, the reprogramming was the perturbations of the expression of at most k genes to suppress sustained oscillations of all the selected markers (stabilize the markers oscillating to desired states 1 or 0). See, for example, these papers:

I. I. Belgacem, J.-L. Gouzé, R. Edwards, Control of negative feedback loops in genetic networks, in: 2020 59th IEEE Conference on Decision and Control (CDC), IEEE, 2020, pp. 5098–5105.

II. L. Chambon, I. Belgacem, J.-L. Gouzé, Qualitative control of undesired oscillations in a genetic negative feedback loop with uncertain measurements, Automatica 112 (2020) 108642.

where a similar problem is investigated to suppress sustained oscillations for genetic negative feedback loops in general. The authors proposed in these papers a sliding mode control to suppress sustained oscillations for genetic negative feedback loops by controlling only the expression of one gene (which is physically possible). Here, the author should include a more detailed explanation of the practical applicability of controlling the expression of many genes at the same time or in parallel and shows its (the control) potential interest for the control community. After, the author for example considers ten models with high dimensions (of up to 75) and for the reprogramming up to 6 simultaneous perturbations (k=6). Many solutions were suggested or found for reprogramming, however which ones have to be more feasible for the control implementation? Could we always find control of each function of some genes to be 0 or 1 (according to each equation)? Are there solutions that have very little practical relevance? Do all the solutions make sense physically?

The author of this paper also considers only locally-monotone BNs, where each local function is unate. This means that the system conserves the partial ordering of solutions. Monotone systems also have strong properties of convergence towards equilibria. Is the proposition of locally-monotone functions simplified much for reprogramming? Furthermore, BoNesis is only supporting locally-monotone BNs... How about if we take general BNs?

The paper is well written. My only minor suggestions to the author are as follows.

1- The paper has some typos issue to be checked and corrected. For example, in the 13th page: to itrs right configuration, I believe that it should be: to its right configuration.

2- Similarly, in the top of the page 22: for some BNs they will have no fixed points, therefore all their fixed points match with the marker, I believe that the author would like to say for some BNs, they will have no other fixed points, ... etc.