




Systems biology

CANTATA—prediction of missing links in Boolean networks using genetic programming

Christoph Müssel[†], Nensi Ikonomi [†], Silke D. Werle[†], Felix M. Weidner ,
Markus Maucher, Julian D. Schwab[‡] and Hans A. Kestler *,[‡]

Institute of Medical Systems Biology, Ulm University, Ulm, Baden-Wuerttemberg 89081, Germany

*To whom correspondence should be addressed.

[†]The authors wish it to be known that, in their opinion, the first three authors should be regarded as Joint First Authors.

[‡]The authors wish it to be known that, in their opinion, the last two authors should be regarded as Joint Last Authors.

Associate Editor: Karsten Borgwardt

Received on March 2, 2022; revised on August 25, 2022; editorial decision on September 8, 2022; accepted on September 9, 2022

Abstract

Motivation: Biological processes are complex systems with distinct behaviour. Despite the growing amount of available data, knowledge is sparse and often insufficient to investigate the complex regulatory behaviour of these systems. Moreover, different cellular phenotypes are possible under varying conditions. Mathematical models attempt to unravel these mechanisms by investigating the dynamics of regulatory networks. Therefore, a major challenge is to combine regulations and phenotypical information as well as the underlying mechanisms. To predict regulatory links in these models, we established an approach called *CANTATA* to support the integration of information into regulatory networks and retrieve potential underlying regulations. This is achieved by optimizing both static and dynamic properties of these networks.

Results: Initial results show that the algorithm predicts missing interactions by recapitulating the known phenotypes while preserving the original topology and optimizing the robustness of the model. The resulting models allow for hypothesizing about the biological impact of certain regulatory dependencies.

Availability and implementation: Source code of the application, example files and results are available at <https://github.com/sysbio-bioinf/Cantata>.

Contact: hans.kestler@uni-ulm.de

Supplementary information: [Supplementary data](#) are available at *Bioinformatics* online.

1 Introduction

Mathematical modelling and simulation have become essential tools for providing a deeper understanding of regulatory processes in biological systems. A particularly popular class of models for the description of biochemical interactions is Boolean networks (Kauffman, 1969, 1993). This qualitative dynamic approach was initially proposed for gene-regulatory networks but can also express a variety of other regulatory systems, such as signal transduction networks. In fact, although this qualitative behaviour constitutes a simplification, Boolean networks approximate the real nature of regulatory processes in biochemical systems well (Albert and Othmer, 2003; Dahlhaus *et al.*, 2016; Davidich and Bornholdt, 2008; Herrmann *et al.*, 2012; Ikonomi *et al.*, 2020; Meyer *et al.*, 2017; Siegle *et al.*, 2018). This modelling framework models compounds as Boolean variables: a compound $x_i \in \mathbb{B}$ can either be active ($x_i = 1$, e.g. a gene is expressed) or inactive ($x_i = 0$, e.g. a gene is not expressed). Boolean functions model regulatory interactions among compounds. A transition function $f_i : \mathbb{B}^n \rightarrow \mathbb{B}$ describes the

regulatory effects on a compound i by the other compounds of the system. Usually, a transition function f_i only depends on a portion of the nodes in the network, such that the function can be simplified to $\tilde{f}_i : \mathbb{B}^{k_i} \rightarrow \mathbb{B}$, $k_i \leq n$. The k_i nodes that determine the value of \tilde{f}_i correspond to the upstream regulators of compound i and are denoted as the *dependencies* or *links* of this compound. The dynamics of Boolean networks are described in discrete time steps. The value of each compound defines the state of a system at a time at this point (see Section 3 for a more detailed mathematical description). In synchronous Boolean networks, a state transition is done by applying all Boolean functions at the same time. The number of states in a Boolean network of the size n is 2^n (Schwab *et al.*, 2020). Due to this deterministic behaviour, after a given number of state transitions, the Boolean network eventually converges to a recurring number of states. These stable states—so-called attractors—represent the long-term behaviour of the Boolean network. For researchers, attractors are of particular importance as they can be linked to biological phenotypes (Kauffman, 1993; Thomas and Kaufman, 2001).

Typically, Boolean networks can be modelled manually based on literature knowledge. Here, natural language statements describing regulatory dependencies can be translated to Boolean functions. Alternatively, a variety of reverse engineering tools has been designed to assist in the construction of these network models (e.g. Akutsu *et al.*, 2000; Lähdesmäki *et al.*, 2003; Liang *et al.*, 1998). When using these algorithms, Boolean functions can be inferred from the binarized series of biomolecular measurements over time (e.g. Hopfensitz *et al.*, 2012; Maucher *et al.*, 2011). With current methods, data and also biological knowledge can vastly be created. Nevertheless, while the function of biological systems and the conditions resulting in various phenotypes are often known, the detailed underlying mechanisms are unexplored. With the growing number of compounds that are identified to be part of a regulatory process, the number of potential interactions also increases. Moreover, measurements of a system are sparse and collect only a few time points, mostly belonging to stable states and not to the progression through time. Consequently, available data are often still incomplete and only lead to partial knowledge about the regulatory mechanisms of the system. Overall, this issue requires strategies to fill the gaps with the available fragmented information.

Various approaches aimed to use Boolean networks to predict missing values in experimental data, which take into account the potential missing information in high throughput data (Crespo *et al.*, 2013; Ogundijo *et al.*, 2016). However, refinement of prior networks by integration of new biological data is still poorly investigated. Only a few studies that aim to predict missing links in Boolean networks using phenotypical data are available. However, these approaches rely on manual investigation (Azpeitia *et al.*, 2010; Liquitaya-Montiel and Mendoza, 2018). This investigation is not only time intensive, but it also tempts to be biased by the experimenters' knowledge (Tanaka *et al.*, 2017). Hence, there is an increasing need for automated and reliable methods to assist the integration of phenotypical data in regulatory models.

To overcome these limitations, we aim to reproduce and optimize these manual procedures by designing a completely data-driven machine intelligence approach. Thus, we propose a novel algorithm, CANTATA ('Computer-Aided Network Transformation According To dynamic Attributes'), that integrates (A) Boolean networks based on incomplete knowledge with (B) the knowledge about the behaviour of the regulatory process, to generate hypotheses about missing interactions in the prior network (see Fig. 1).

2 Approach

Implementing a machine intelligence approach, CANTATA can support life scientists in network construction and refinement by automatically combining all available knowledge on a system to infer a new global model consistent with this information. In fact, our method imitates the typical process of iterative model refinement that is applied when models are constructed manually by experts: a guided evolutionary process gradually transforms the initial draft on the basis of the expected dynamic behaviour, by changing or inserting new interactions between regulators, while maintaining as much as possible the original topology. First, we applied genetic programming to simulate an evolutionary process, guiding our candidate networks towards the desired phenotypes. To do so, we used a symbolic representation of BNs. This has the main advantage of inserting changes with little impact in the network at each mutation round if compared to truth table-based approaches. The second new concept tackled in our algorithm is the evaluation of the fitness score, which is approached for the first time as a multi-objective problem. Besides considering topology preservation and stability evaluation, we also compared dynamic behaviour. By considering attractors as a series of numeric strings, we solved the comparison problem by dynamic programming. Dynamic programming is a well-known strategy for string comparison. For instance, it is used to compare similarities between DNA sequences (Needleman and Wunsch, 1970). Here, we applied it to compute the similarity between sequences of Boolean network states. The evolutionary selection process finally yields a small set of candidate models that still

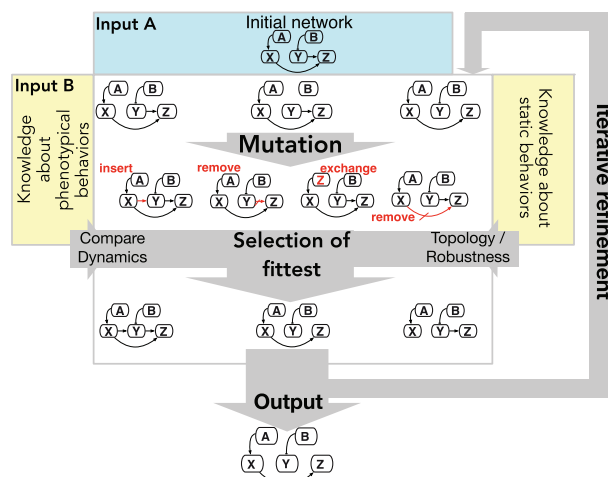


Fig. 1. The workflow followed by CANTATA is depicted. The approach integrated both knowledge about behaviour, either in the form of database information or experimental data, and knowledge about regulations. The latter can again be inferred by databases, literature or experiments and is used to construct an initial network. Finally, by a process of guided evolution, CANTATA iteratively modifies the prior network to better fit the information about systems' behaviour. Hereby, networks are modified based on random mutation. Next, the fittest networks are selected by examining their fitness according to both, phenotypic and static behaviours. Finally, the fittest networks are returned by the algorithm

resemble the model draft closely but show behaviour that coincides with the given ruleset. Through information integration, we can eliminate the vast majority of false candidates occurring in common reconstruction methods that are solely based on experimental data. Not last, we also considered the potential incomplete and variegated nature of biological data. In fact, the prior model parsed to CANTATA can be manually created or reverse engineered from time-series data. The quality of this draft may range from loose assemblies of static regulatory relationships to more refined models describing the detailed interplay of multiple regulators. Our approach can handle incompleteness and defectiveness of the model drafts, such as missing or incorrect information on regulatory dependencies. Similarly, also the knowledge about the phenotypical behaviour used for optimization is described by a highly flexible rule set that covers various types of data, such as time series of measurements, attractors in the wild type, or under perturbation conditions. Incomplete information, such as time series that do not provide information about some regulatory factors in one or more time steps, is still a valid input. Overall, apart from exhibiting a high predictive power, the inferred models are also designed to be intuitively interpretable by human experts.

In the following, we detail our approach of guiding network inference by knowledge integration. Additionally, CANTATA will be employed in three case studies, highlighting the potentiality of our method compared to others. Overall, our results present an automated process that mimics manual procedures for integrating missing information in regulatory networks for the first time.

3 Materials and methods

Our algorithm CANTATA optimizes network models towards a certain behaviour based on a multi-objective genetic programming approach. The algorithm is supplied with a set of rules \mathcal{R} describing the dynamic behaviour of the regulatory process (e.g. time series or attractors) and an initial Boolean network model draft D with n nodes and transition functions. The algorithm maintains a population of m individuals. Each individual represents a candidate network model N , also consisting of n transition functions. To enable the general GP approach for the evolution of Boolean networks, we developed (i) a specific representation of Boolean networks as encoding of the individuals, (ii) a method to mutate Boolean

networks based on this encoding and (iii) a new approach to measure the ability of Boolean networks to recapitulate a certain dynamic behaviour which is used to determine the fitness of the individual networks.

In the following, these three parts of CANTATA will be elaborated in more detail.

3.1 Encoding of Boolean networks

To achieve a high interpretability of the network models and a high similarity to the original model draft, we use a symbolic tree representation for the transition functions instead of a truth table representation (see Fig. 2 for a comparison of the two representations). Each of these trees is composed of

- *literals*, which form the leaves of the tree. A literal is a node or the negation of a node in case of an inhibitory effect.
- the *operators* AND and OR (and their negations), which form the inner nodes of the tree.

Hence, one individual is represented by a list of n Boolean expression trees—one tree for each regulatory function.

3.2 Mutation

To form a new generation of individuals, m^* offspring are created. A certain percentage A (by default $A = 90\%$) of these offspring is created by randomly selecting individuals from the previous generation and applying a mutation operator, i.e. a random change of the expression tree. The remaining offspring are mutated copies of the input network model draft D . This injection of unoptimized drafts aims at maintaining solutions that resemble the original network closely.

For the same reason, we do not employ cross-over operators, as these often disrupt the original structure of a formula entirely.

A network model N is mutated by first selecting a function f_i of the network randomly and then applying a random mutation operator, i.e.

- random insertion of a new literal into the tree
- insertion of a new operator into the tree
- deletion of a randomly selected subtree
- changing the type of an operator (AND to OR and vice versa)
- negation of a randomly selected subtree

As it is mostly known whether a regulator in the model draft has an activating or inhibitory effect, the negation operation is often implausible. Nevertheless, it sometimes provides the easiest way of changing the dynamics of the model appropriately. The use of this operator is therefore restricted to every i th randomly chosen individual (e.g. $i = 50$). In this way, this operator is still applicable in rare cases where it is needed, but the algorithm will opt for other solutions.

3.3 Computation of the fitness scores

For each of the $m + m^*$ candidate individuals, the fitness is calculated. We employ a multi-objective optimization with a 3D fitness to account for different aspects of the network model. The primary goal of the optimization is to match the desired dynamic behaviour. On top of that, for equally performing networks, the structure and robustness of the model are considered as the second and third objectives. In detail, for an individual (corresponding to a network model N) the following criteria are assessed:

- **Network dynamics:** The first objective measures the concordance of the model dynamics with a set of rules that describes biomolecular measurements—time series of biomolecular measurements or attractors of the network. Each such rule comprises a precondition and a list of expected node states corresponding to time series or attractors in the network model. A rule is matched with the network model N using Dynamic Programming. On a more abstract level, a rule can formalize statements such as: *If Gene A is transcribed and Gene B is not transcribed, a cascade of gene transcriptions is activated*, or: *The presence of Gene C expression initiates the cell cycle*. It is also possible to simulate perturbation and permanent knock-out or stimulation of compounds. This allows for statements like: *If Gene D is knocked out, the cell enters a certain steady state*. We extend the notion of sequences in that the states do not have to be direct successors, which accounts for a possible biological interpretation: Each state specification entry can be thought of as the expression pattern of a certain developmental stage, and there may be unknown and unspecified intermediate states between such stages that were not measured.
- **Network topology:** The second objective rates the structural properties of the network model N . These include the size of the transition functions and the similarity to the input network draft D . Furthermore, this objective punishes setting nodes to constant values.
- **Robustness:** The third objective quantifies the robustness of the network to noise. A set of states is generated randomly, and copies of the states with one randomly flipped bit are created. The network model is robust to noise if the average Hamming distance of the successor states of the original states and their perturbed copies stays small.

These three objectives form a 3D fitness function $F = (f_1, f_2, f_3)^T$. All three objectives are minimization objectives that can take values between 0 and 1.

To create a new generation from the $m + m^*$ candidate models (i.e. the previous generation and the m^* offspring), the best m individuals are chosen for the next generation according to a lexicographical order of the 3D fitness score. Lexicographical ordering here means that there is a hierarchy on the three objectives, i.e.

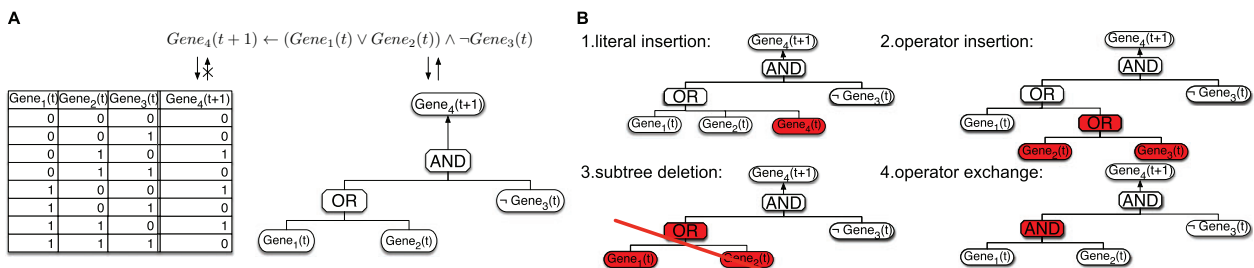


Fig. 2. (A) A truth table representation (left) and symbolic tree representation (right) of the Boolean transition function $Gene_4(t+1) \leftarrow (Gene_1(t) \vee Gene_2(t)) \wedge \neg Gene_3(t)$ for an example of gene regulation. Here, $Gene_1$ and $Gene_2$ activate the transcription of $Gene_4$, while $Gene_3$ inhibits $Gene_4$. In the tree, genes are represented as leaf nodes while operators correspond to inner nodes. While the formula and the tree representation are exchangeable, the conversion of a truth table back to a formula is not unique. (B) The four different type of mutation which are implemented to the CANTATA framework. First, the insertion of literals at a randomly selected position of the tree which represents the corresponding Boolean function (upper left image). Alternatively, the insertion of an operator between two existing literals (upper right). The third case is the deletion of a complete subtree, again, drawn randomly (lower left). Fourth, the exchange of a randomly drawn operator to another one (lower right)

network dynamics takes precedence over topology, and this again over robustness.

In the following, we describe the fitness measure according to network dynamics in more detail. For a detailed description of robustness and topology measurement, see [Supplementary material S1](#).

The desired/expected dynamics of the network model N are described as a set of rules $\mathcal{R} = \{R_1, \dots, R_{n_{\mathcal{R}}}\}$. Each rule $R_k \in \mathcal{R}$ consists of an *initial condition* I and a *state specification list* SL describing either an attractor or a time series of states. Such a rule may express a statement such as: ‘If Gene 1 is expressed and Gene 2 is not expressed, etc., then the model is expected to end up in this attractor’. To rate the quality of a network model, we evaluate how well the dynamic behaviour of the model complies with the expected dynamics. This is done by identifying their best possible matching, which can be done for attractors and sequences of non-attractor states. The initial condition of a rule is a conjunction of literals describing the states of genes prior to reaching the attractor or traversing a time series of states. It may contain only a portion of the genes in the network and leave the remaining genes unspecified. From this condition, a set of initial network states \mathcal{S} can be generated. Furthermore, *CANTATA* allows for perturbed network conditions with knocked-out or overexpressed compounds. For each initial state $s \in \mathcal{S}$, the state specification list SL is matched. The final score $T_{R_i}(N)$ of a network model N for rule $R_i = (I, SL)$ is the average of the subscores for all initial states in \mathcal{S} , i.e.

$$T_{R_i}(N) = \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} S_{SL}^{*(s)}. \quad (1)$$

Here, $S_{SL}^{*(s)}$ denotes the score of an optimal matching of the attractor or sequence of states initiated by s with the corresponding specification list SL . The calculation of these scores is detailed in the following.

The complete objective function is made up by the scores of all rules in \mathcal{R} , which may describe different dynamical aspects. It is calculated as

$$f_1(N) = \frac{1}{n_{\mathcal{R}}} \left(n_{\mathcal{R}} - \sum_{i=1}^{n_{\mathcal{R}}} T_{R_i}(N) \right), \quad (2)$$

with $n_{\mathcal{R}}$ being the number of rules and $f_1(N)$ evaluating to zero if all rules match the network perfectly.

In the following, we outline the matching of state specification lists with attractors or state sequences of the network N that yields the above scores $S_{SL}^{*(s)}$ for a specific start state s . We will consider the start state s as fixed and simplify the notation of the score to S_{SL}^* . A state specification list $SL = (c_1, \dots, c_q)$ consists of q conjunctions c_j (the *state specification entries*) describing the states in the sequence of states or the attractor. Each literal in a conjunction c_j specifies the expected value of a gene (0 or 1). The conjunctions do not necessarily specify the values of all genes—unspecified genes are treated as ‘don’t care’ values, i.e. any value of the genes is considered as correct. Below, we describe the attractor matching in more detail. The sequence matching is in line with this approach. For a more detailed description on the sequence matching see [Supplementary material S1](#).

If SL describes an attractor, the network model is expected to end up in a cycle that can be described by the state specification entries when starting from a state s matching the initial condition. We identify the attractor $\mathcal{A} = (a_1, \dots, a_p)$ of the candidate network model N by performing repeated state transitions using s as a starting state: When a previously traversed state is reached again in the repeated transitions, we have entered the attractor, and all states between the two traversals belong to \mathcal{A} . The states a_i of \mathcal{A} and the specification entries c_j are then matched to quantify the agreement. We describe a matching by a function $M : \mathcal{A} \rightarrow SL$ which has the following properties:

- Each state a_i of the attractor \mathcal{A} is associated with exactly one state specification entry c_j , i.e. $\forall a_i \in \mathcal{A} \exists! c_j \in SL : M(a_i) = c_j$.
- If $M(a_{i_1}) = c_{j_1}$, $M(a_{i_2}) = c_{j_2}$, and $i_1 < i_2$, then $j_1 \leq j_2$. That is, the order of the states coincides with the order of the specification entries.

It is not required that all state specification entries are covered by an attractor state a_i . However, if this is not the case, the agreement of the attractor and the specification list is obviously suboptimal. The fraction of literals in the state specification conjunctions c_j is defined as:

$$Sat(a_i, c_j) = \frac{\sum_{g \in c_j} \mathbb{1}_{[c_{jg}=a_{ig}]}}{|c_j|}$$

with g being the index of the literals in c_j and a_i .

Consequently, a good matching has the following properties:

- The fraction of literals in the state specification conjunctions c_j that are satisfied by an associated attractor state a_i , $Sat(a_i, c_j)$, should be as high as possible. In other words, the matching of the attractor states and the specification entries should yield a small error or—in case of a perfect match—no error.
- The number of state specification entries that are not covered by any state of the attractor should be small or zero.

We define the score of a matching as the fraction of fulfilled literals multiplied with the fraction of state specification entries associated with an attractor state, i.e.

$$S_{SL}(M) = \frac{\sum_{i=1}^p Sat(a_i, M(a_i)) \cdot |\{c \in SL \mid \exists a \in \mathcal{A} : M(a) = c\}|}{p \cdot q}$$

This score is 1 if the attractor \mathcal{A} matches the specifications in SL perfectly. It is 0 if none of the literals in the specification entries is fulfilled or if no specification entry is covered by an attractor state. In order to rate the quality of a network model according to an attractor rule, we have to determine the optimal matching M^* that associates the states in \mathcal{A} and the specifications in SL . This is done using a Dynamic Programming approach. The matching algorithm is based on the recursion function $F_o(b, i, j)$ that counts the fractions of satisfied literals for a partial matching of the first i states of attractor \mathcal{A} and the first j entries of specification list SL , given that at most b specification entries are not associated with an attractor state. Furthermore, the attractor sequence \mathcal{A} must be rotated such that each state a_i is once used as the starting state for the matching. This is described by an offset o .

$$F_o(b, i, j) = \max \begin{cases} Sat(a_{(i+o) \bmod p}, c_j) + F_o(b, i-1, j-1) & (1) \\ Sat(a_{(i+o) \bmod p}, c_j) + F_o(b, i-1, j) & (2) \\ F_o(b-1, i, j-1) & \text{if } b > 0 \end{cases} \quad (3)$$

with $F_o(b, i, 0) = -\infty$ for $i = 1, \dots, p$ and $b = 0, \dots, q-1$, $F_o(b, 0, j) = 0$ for $j = 0, \dots, b$ and $b = 0, \dots, q-1$, and $F_o(b, 0, j) = -\infty$ for $j = b+1, \dots, q$ and $b = 0, \dots, q-1$. Ties in the maximum calculation are broken in favour of the first alternative.

The score of the optimal matching M^* , $S_{SL}^* = S_{SL}(M^*)$, is

$$S_{SL}^* = \frac{1}{p \cdot q} \max_{\substack{b=0, \dots, q-1 \\ o=0, \dots, (p-1)}} F_o(b, p, q) \cdot (q-b).$$

That is, we adjust $F_o(b, p, q)$ according to the minimal number of associated specification entries $q-b$ and take the maximum value over all rotation offsets o and all numbers of unassociated entries b . The optimal matching can be reconstructed by tracing the chosen alternatives (see [Supplementary Fig. S.2](#)), but this is not necessary here, as we are only interested in the score itself.

4 Benchmarks

For the evaluation of our method, we created a set of benchmark problems:

- I. We randomly disarranged a given Boolean network of the yeast cell cycle ([Davidich and Bornholdt, 2008](#)) and compared the

reconstructed results to the original networks to measure the prediction accuracy.

- II. We applied our approach to perturbation data of a signalling network of hepatocytes from the DREAM challenge (Prill *et al.*, 2011).
- III. As a comparison to another approach, we applied CANTATA to predict missing links in a Boolean network of *Arabidopsis thaliana* stem cells, similar to a method by Azpeitia *et al.* (2010).

In the following, we discuss the first approach in more detail. Benchmarks (II) and (III) are discussed in the [Supplementary materials Sections S1 and S2](#).

5 Results

To investigate our method's ability of identifying crucial regulatory dependencies, we devised a simulation study based on a Boolean model of gene regulation in the fission yeast cell cycle. By randomly scrambling and removing parts of the networks, we simulate incomplete model drafts. We then reconstruct networks from the scrambled model drafts and time series information and compare the reconstructed models to the true biological model. The computer-intensive simulation studies assess the quality of reconstructed models by averaging over many reconstruction scenarios that reflect realistic settings. The design of the studies is depicted in [Supplementary Figure S.1](#). We employ a well-known Boolean model of the cell cycle sequence of fission yeast (Davidich and Bornholdt, 2008) as the ground truth to which the reconstructed results are compared. For the 10 genes in the network, we transformed the original perceptron representation into a set of minimal Boolean transition functions in disjunctive normal form.

Artificial measurements are generated by simulating the network model. The network has 13 attractors (12 steady states and 1 attractor with three states). The basins of attraction of two of the attractors—a steady state modelling the G1 state and the three-state cycle—cover around 95% of the states of the network. The cell cycle itself is described as a sequence of 10 states traversing the phases G1, S, G2 and M and ending up in the G1 steady state. We supplied our method only with this time series and the two most relevant attractors.

Additionally, fragmentary model drafts are imitated by generating 100 randomly scrambled copies of the true network model. Each of the copies is subject to c random changes:

- With a probability of 0.75, a change operation randomly removes a dependency from the model. That is, all occurrences of literals corresponding to a certain transcription factor are deleted from a randomly chosen transition function. This corresponds to incomplete knowledge on genetic interactions in the model draft.
- With a probability of 0.25, a change operation randomly adds a false link to the network. This corresponds to the (presumably less frequent) case of incorrect assumptions about dependencies between genes.

In two substudies, we generate two sets of 100 scrambled models by applying $c = 5$ and $c = 7$ random changes to the original network. These studies are called Study A and Study B, respectively. For each scrambled model draft, we then reconstruct network models by supplying our algorithm with the draft and the simulated measurements from the true network. Each simulation study thus summarizes over 100 separate reconstruction processes. The reconstructed models are compared to the corresponding true model. CANTATA's parameters were set to $n_g = 1000$ generations of $m = 100$ individuals, $m^* = 200$ offspring and $n_s = 5$ restarts. Our method can identify candidates with perfect fitness based on the expected dynamics for 98 of the 100 network drafts in simulation Study A and for 91 of the 100

drafts in Study B. The average number of matching candidate networks returned by CANTATA in a successful reconstruction is 8.7 for simulation Study A and 8 mutations Study B. Among the reconstructed networks with perfect fitness, we further evaluated the reconstruction performance. To measure the goodness of the reconstruction: (i) We measured the mean accuracy of all the interactions in the reconstructed networks compared to the original network to measure the complete overlap. (ii) We calculated the mean accuracy when comparing only the interactions that were changed between original and scrambled networks with the reconstructed results. While (i) focuses on the overall similarity of the reconstructed and original networks, with (ii) we measure how precise the set of changes between original and scrambled networks was reconstructed. Given multiple equally performing results by CANTATA, their respective interactions were integrated into one solution before computing the accuracy. Here, results in Study A show a mean accuracy across all reconstructed networks of (i) 0.979 (SD = 0.0161) and (ii) 0.622 (SD = 0.13). Vice versa, the results in Study B show a mean accuracy of (i) 0.975 (SD = 0.015) and (ii) 0.631 (SD = 0.133).

Looking at the individual interactions, [Figure 3](#) provides details on the dependencies of the reconstructed network models. Here, each cell of a table denotes a single dependency (with the targets in the rows and the transcription factors in the columns). The top-level numbers in the cells correspond to the percentage of reconstructed models exhibiting the dependency. If the associated dependency is a true dependency of the original mammalian cell cycle model, the cell is coloured in different shades of green. If the associated dependency is not present in the model, the cell is coloured in shades of red. The intensity of the colour corresponds to the relative availability of this dependency across all reconstructed networks. If green dominates the table, most reconstructed dependencies are true dependencies of the original model. By contrast, if many cells are coloured in blue, the reconstruction has inserted many links that were not present in the original model. The relative changes of the reconstructed dependencies compared to the original models is supplied in brackets. This difference indicates whether the algorithm tends to insert or remove the dependency. On the basis of the difference, we assess the hypothesis that our algorithm changes the frequency of the link significantly using an exact Fisher test. A significant change (p -value < 0.05) is indicated by an asterisk (*), and a highly significant change (p -value < 0.01) is indicated by two asterisks (**). We applied a Bonferroni correction to account for multiple testing. In both scenarios, our algorithm is able to reconstruct true dependencies reliably. In Study A, eight true dependencies are significantly strengthened by CANTATA; four of them are even present in all networks. The presence of two false links is increased significantly as well: the strongest false link is the dependency of Cdc25 on Wee1/Mik1. The opposing dependency of Wee1/Mik1 on Cdc25 is also present in a high number of networks, although it is not significantly increased in experiment A. There are plausible biological explanations for the insertion of these two links: Wee1 and Cdc25 are opposites in their roles in the cell cycle, with Cdc25 being an activator of mitosis and Wee1 being an inhibitor of mitosis (see e.g. Perry and Kornbluth, 2007). Both have been shown to be co-regulated by a shared set of factors. Although most of these factors are not present in the network model of Davidich and Bornholdt (2008), the transition functions also reflect this antagonistic behaviour. For CANTATA, an inhibition by one of the two genes is thus indistinguishable from an activation by the other gene. Thus, it is plausible to replace missing activating dependencies $Wee1/Mik1 \rightarrow Wee1/Mik1$ or $Cdc25 \rightarrow Cdc25$ in the scrambled drafts by the opposing inhibiting dependencies $Wee1/Mik1 \rightarrow Cdc25$ or $Cdc25 \rightarrow Wee1/Mik1$. The second false link whose presence is increased significantly is the dependency of Cdc25 on Slp1. Here, Slp1 is mostly employed as a replacement of the self-dependency $Cdc25 \rightarrow Cdc25$ which is missing in around 40% of the scrambled drafts. For this link, there seems to be no biological evidence. In Study B, nine true dependencies occur significantly more often than in the drafts. Four of these links are present in 100% of the reconstructed networks. The higher level of noise results in four false links

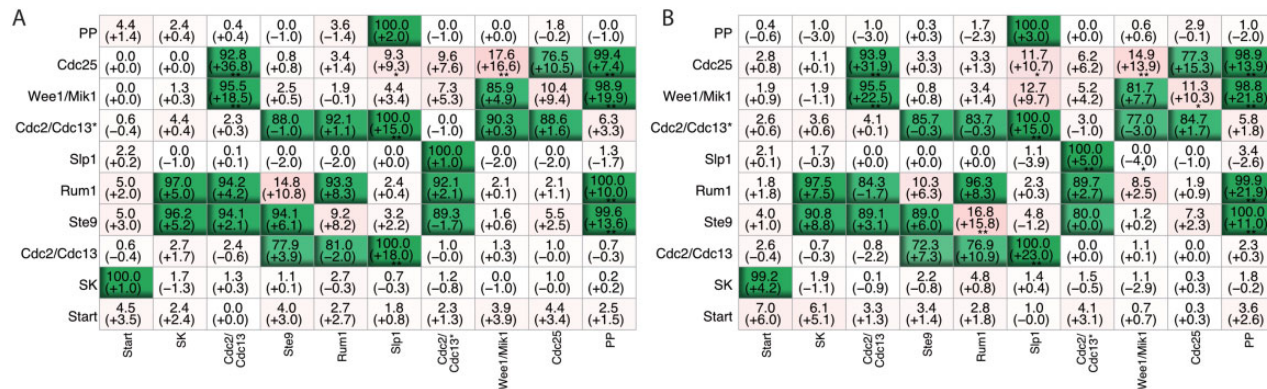


Fig. 3. Overview of the fission yeast cell cycle case study, based on 100 scrambled network drafts. **(A)** Each of the networks was created by applying $c = 5$ random changes to the original network. **(B)** We applied $c = 7$ random changes. These changes can be removal of existing interactions or adding false links. In the tables, rows correspond to target genes, and columns correspond to transcription factors. The numbers at the top of the cells denote the percentage of reconstructed networks exhibiting the corresponding dependency. p -values are calculated using exact Fisher test and are Bonferroni-corrected to account for multiple testing. The numbers in brackets specify the increase or decrease relative to the scrambled input drafts, and the asterisks specify if this change is significant ($*p < 0.05$) or highly significant ($**p < 0.01$). Cells are drawn in shades of green with shadow if the corresponding dependency exists in the true (unperturbed) network. They are drawn in shades of red if the corresponding dependency does not exist in the true network.

with a significant increase in occurrence compared to the true model, among them the opposing interactions of Wee1/Mik1 and Cdc25 as well as the dependency of Cdc25 on Slp1 discussed above. The strongest false link is the dependency of Ste9 on Rum1, which is present in 17% of the reconstructed models. Kitamura *et al.* (1998) state that mutants of these two genes exhibited indistinguishable defects. The similar behaviour is also reflected in the transition functions of both genes in the true model: Apart from self-dependencies $Ste9 \rightarrow Ste9$ and $Rum1 \rightarrow Rum1$, they are equal. Similarly to the case of Wee1/Mik1 and Cdc25, our method can use these functionally identical genes exchangeably if the corresponding dependencies are missing in the scrambled drafts, leading to increased false link rates for $Ste9 \rightarrow Rum1$ and $Rum1 \rightarrow Ste9$. Hence, these ‘false links’ represent a type of redundancy identified by CANTATA rather than true errors.

6 Discussion

Boolean networks have been shown to be a powerful tool to predict the behaviour of biological systems based on modelled regulatory interactions. However, the more these models are used in the context of systems biology, the more it has been realized that our current information on various biological processes is limited, if not incomplete. There is a variety of different approaches to infer interactions or complete Boolean networks from time-series of data (Lähdesmäki *et al.*, 2003; Maucher *et al.*, 2011; Schwab *et al.*, 2021; Shi *et al.*, 2020). Barman and Kwon (2018) used a genetic approach to reconstruct Boolean networks from time series data. In contrast to those algorithms, approaches to adapt existing networks to a desired behaviour are rare. Exemplarily, Azpeitia *et al.* (2010) showed incomplete knowledge to affect the prediction power of their model on *A.thaliana* stem cells. Hence, approaches aimed to integrate and infer unknown interactions in Boolean networks models are of great interest. In this direction, previous theoretical work has been accomplished. Pal *et al.* (2005) investigated how to produce a BN with a definite set of attractors. Later, Zou (2010) provided insights on how ground requirements in obtaining certain attractors when network topology and Boolean functions are partially known. Nevertheless, these work stay prompted from a theoretical perspective, away from a generalized applicable method. Finally, Azpeitia *et al.* (2013) tried to infer missing interactions for their previously published model on *A.thaliana* stems. Here, the authors showed different approaches to introduce new interactions in their model. However, their work is not formalized, case-specific and still requires manual preparation steps to be performed before starting the algorithm. Moreover, the authors propose six different

processes to be applied to infer interactions that might lead to a laborious and not rigorous experimental procedure. On the other hand, our novel approach implemented in CANTATA incorporates knowledge in the form of fragmentary model drafts from regulatory knowledge and combines this topological information with behavioural information. In accordance when comparing our approach to the one of Azpeitia *et al.* (2013), we could show that CANTATA could extensively recapitulate their results, with the additional advantage of retrieving all the desired attractors and eliminating additional cyclic ones (see Supplementary material S2). Here, in fact, the authors aimed to predict a set of missing links in a prior Boolean network so that it matches nine specific cell types specified as attractors. While the procedure by the authors led to a network matching seven out of nine of these attractors, CANTATA led to networks comprising each of the nine attractors.

In another study, we took an existing model of the fission yeast cell cycle by Davidich and Bornholdt (2008) and scrambled it by randomly adding new interactions or deleting existing interactions. We then ran CANTATA on these scrambled networks to reproduce the behaviour of the original network and to measure how well the changes could be reverted. Results show that CANTATA lead to the original behaviour in 98% of the networks with five changed interactions and in 91% of the networks with seven changes. Furthermore, when comparing the resulting networks with the original networks, we can see the high similarity (mean accuracy of 0.97 for both studies). This shows that CANTATA is leading to results with minimal changes to recapitulate the desired behaviour. When comparing the changed interaction during the reconstruction process to the changes that were applied by scrambling the networks, we observed that a substantial amount of these changes were found and corrected by CANTATA (mean accuracy of 0.62/0.63, Fig. 3A or B). However, results also point out that there are different interactions that lead to the expected results with equal quality. By investigating the biological relevance of the predicted links which differ from the original model, we could connect these to the phenomenon of biological redundancy in regulatory networks. Overall, we show that CANTATA was not only able to revert applied changes but also to suggest other interactions with biological plausibility.

Artificial intelligence approaches have been applied to unravel unknown biological mechanisms (Razaghi-Moghadam and Nikoloski, 2020; Song *et al.*, 2020), such as new functions of transcription factors from -omics-data (Razaghi-Moghadam and Nikoloski, 2020). In CANTATA, genetic programming is elaborated to guide an evolutionary transformation process, yielding network models that resemble the initial model drafts closely while matching the observed dynamic behaviour. The algorithm ensures minimal

interventions in the network drafts by relying on symbolic representation. When modifying a given network model draft, the truth table representation has several drawbacks: as truth tables are not intuitively understandable for humans, models will usually be specified in a symbolic form, i.e. as Boolean expressions. While the conversion from the symbolic level to a truth table representation is unique, there is no distinct way of converting a truth table back to a Boolean expression. Commonly, a conversion is based on normal forms, such as the conjunctive or disjunctive normal form. This means that—even if no modifications occur in a network model—the expressions resulting from the conversion will probably not resemble the input formulae, whose structure reflects the researcher's intentions and assumptions. Furthermore, even the flip of a single bit in the truth table may result in an entirely different Boolean expression. Using the symbolic notation, we aim at the high interpretability of the network models and high similarity to the original model draft.

One main advantage in CANTATA is exactly the possibility to assess directly the fitness of the modified networks from together a topology, robustness and dynamic behaviour perspective. Accordingly, both the topology and the behaviour of the models constructed by our method conform excellently to biological observations. In the *in silico* studies, we applied our method to a well-known model of gene regulation and showed that it reliably identifies relevant regulatory dependencies. By additionally eliminating false links, it extracts the most plausible models from the extremely large set of possible models. Our analysis also indicates that we efficiently identify redundancy among regulatory interactions in a biologically plausible way. In this way, the algorithm generates new hypotheses on potential interactions and coregulations. In our second experiment, an application to perturbation data from a signalling network of hepatocytes (see [Supplementary material S1](#)) confirmed the plausibility of inferred regulatory interactions. Here, the method discovered a well-described regulatory interaction in NfκB signalling. Yet, the main focus of this evaluation was on the dynamic behaviour of the network models: We assessed their ability to predict the effects of upstream interventions that were not known to the inference algorithm. The cross-validation proves that the reconstructed models are able to simulate the behaviour of the underlying system with a very high accuracy.

Our results show novelty and the completeness of our approach. Even other authors attempted to integrate new information on interaction graphs, their analysis was mainly focused on topology and simplified dynamic prediction, such as single inhibitory or activatory interactions ([Melas et al., 2013](#); [Saez-Rodriguez et al., 2009](#)). Dorier and colleagues proposed a reconstruction approach based on prior network knowledge integrated with experimental data ([Dorier et al., 2016](#)). However, in their approach, they compared dynamics based on single perturbations on the original and in optimized networks. In contrast, our fitness evaluation, is based on a multi-objective optimization. In particular, our dynamic assessment is based on a dynamic programming approach that compared attractors of the evolved networks with the expected ones. Furthermore, runtime complexity (see [Supplementary material S3](#)) of our approach shows the scalability to larger networks as increasing network size is no major contributor to runtime expectations.

7 Conclusion

CANTATA supports investigators in generating refined regulatory networks and allowing constant updates and integration of knowledge in biological regulatory networks. Our approach complements the work of different authors that instead tried to complete the missing information in data series by the support of graph knowledge ([Crespo et al., 2013](#); [Ogundijo et al., 2016](#)). From a more general point of view, these results also confirm that Boolean models can capture the essential behaviour of regulatory systems remarkably well despite their inherent simplicity. Hence, our works set a new pillar in the development of machine intelligence methods aimed automatically refine and integrate information from a systems biology perspective in regulator networks.

Acknowledgements

We thank Martin Hopfensitz, Paolo Frasconi, Michael Kühl and Franziska Herrmann for fruitful discussions.

Funding

This work was supported by the German Science Foundation [DFG, 217328187 (SFB 1074), 450627322 (SFB 1506) to H.A.K.]. Furthermore, H.A.K. acknowledges funding from the German Federal Ministry of Education and Research (BMBF) e: MED confirm [id 01ZX1708C] and TRANSCAN VI—PMTR-pNET [id 01KT1901B].

Conflict of Interest: none declared.

Data availability

The data underlying this article are available in <https://github.com/sysbio-bioinf/Cantata>.

References

- Akutsu, T. *et al.* (2000) Inferring qualitative relations in genetic networks and metabolic pathways. *Bioinformatics*, **16**, 727–734.
- Albert, R. and Othmer, H.G. (2003) The topology of the regulatory interactions predicts the expression pattern of the segment polarity genes in *Drosophila melanogaster*. *J. Theor. Biol.*, **223**, 1–18.
- Azpeitia, E. *et al.* (2010) Single-cell and coupled GRN models of cell patterning in the *Arabidopsis thaliana* root stem cell niche. *BMC Syst. Biol.*, **4**, 134.
- Azpeitia, E. *et al.* (2013) Finding missing interactions of the *Arabidopsis thaliana* root stem cell niche gene regulatory network. *Front. Plant Sci.*, **4**, 110.
- Barman, S. and Kwon, Y.-K. (2018) A Boolean network inference from time-series gene expression data using a genetic algorithm. *Bioinformatics*, **34**, i927–i933.
- Crespo, I. *et al.* (2013) Predicting missing expression values in gene regulatory networks using a discrete logic modeling optimization guided by network stable states. *Nucleic Acids Res.*, **41**, e8.
- Dahlhaus, M. *et al.* (2016) Boolean modeling identifies greatwall/MASTL as an important regulator in the AURKA network of neuroblastoma. *Cancer Lett.*, **371**, 79–89.
- Davidich, M.I. and Bornholdt, S. (2008) Boolean network model predicts cell cycle sequence of fission yeast. *PLoS One*, **3**, e1672.
- Dorier, J. *et al.* (2016) Boolean regulatory network reconstruction using literature based knowledge with a genetic algorithm optimization method. *BMC Bioinformatics*, **17**, 410.
- Herrmann, F. *et al.* (2012) A Boolean model of the cardiac gene regulatory network determining first and second heart field identity. *PLoS One*, **7**, e46798.
- Hopfensitz, M. *et al.* (2012) Multiscale binarization of gene expression data for reconstructing Boolean networks. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, **9**, 487–498.
- Ikonomi, N. *et al.* (2020) Awakening the HSC: dynamic modeling of HSC maintenance unravels regulation of the TP53 pathway and quiescence. *Front. Physiol.*, **11**, 848.
- Kauffman, S.A. (1969) Metabolic stability and epigenesis in randomly constructed genetic nets. *J. Theor. Biol.*, **22**, 437–467.
- Kauffman, S.A. (1993) *The Origins of Order: Self-Organization and Selection in Evolution*. Oxford University Press, Oxford (UK).
- Kitamura, K. *et al.* (1998) Fission yeast Ste9, a homolog of Hct1/Cdh1 and fizzy-related, is a novel negative regulator of cell cycle progression during G1-phase. *Mol. Biol. Cell*, **9**, 1065–1080.
- Lähdesmäki, H. *et al.* (2003) On learning gene regulatory networks under the Boolean network model. *Mach. Learn.*, **52**, 147–167.
- Liang, S. *et al.* (1998) REVEAL, a general reverse engineering algorithm for inference of genetic network architectures. In: *Proceedings of the Pacific Symposium on Biocomputing, Proceedings of the Pacific Symposium on Biocomputing*. World Scientific Publishing Co., Singapore, pp. 18–29.
- Liquitaya-Montiel, A.J. and Mendoza, L. (2018) Dynamical analysis of the regulatory network controlling natural killer cells differentiation. *Front. Physiol.*, **9**, 1029.
- Maucher, M. *et al.* (2011) Inferring Boolean network structure via correlation. *Bioinformatics*, **27**, 1529–1536.
- Melas, I.N. *et al.* (2013) Detecting and removing inconsistencies between experimental data and signaling network topologies using integer linear programming on interaction graphs. *PLoS Comput. Biol.*, **9**, e1003204.

- Meyer, P. et al. (2017) A model of the onset of the senescence associated secretory phenotype after DNA damage induced senescence. *PLoS Comput. Biol.*, **13**, e1005741.
- Needleman, S.B. and Wunsch, C.D. (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.*, **48**, 443–453.
- Ogundijo, O.E. et al. (2016) Reverse engineering gene regulatory networks from measurement with missing values. *EURASIP J. Bioinform. Syst. Biol.*, **2017**, 2–11.
- Pal, R. et al. (2005) Generating Boolean networks with a prescribed attractor structure. *Bioinformatics*, **21**, 4021–4025.
- Perry, J.A. and Kornbluth, S. (2007) Cdc25 and Wee1: analogous opposites? *Cell Div.*, **2**, 12.
- Prill, R.J. et al. (2011) Crowdsourcing network inference: the dream predictive signaling network challenge. *Sci. Signal.*, **4**, mr7.
- Razaghi-Moghadam, Z. and Nikoloski, Z. (2020) Supervised learning of gene-regulatory networks based on graph distance profiles of transcriptomics data. *NPJ Syst. Biol. Appl.*, **6**, 21.
- Saez-Rodriguez, J. et al. (2009) Discrete logic modelling as a means to link protein signalling networks with functional analysis of mammalian signal transduction. *Mol. Syst. Biol.*, **5**, 331.
- Schwab, J.D. et al. (2021) Reconstructing Boolean network ensembles from single-cell data for unraveling dynamics in the aging of human hematopoietic stem cells. *Comput. Struct. Biotechnol. J.*, **19**, 5321–5332.
- Schwab, J.D. et al. (2020) Concepts in Boolean network modeling: what do they all mean? *Comput. Struct. Biotechnol. J.*, **18**, 571–582.
- Shi, N. et al. (2020) ATEN: AND/OR tree ensemble for inferring accurate Boolean network topology and dynamics. *Bioinformatics*, **36**, 578–585.
- Siegle, L. et al. (2018) A Boolean network of the crosstalk between IGF and Wnt signaling in aging satellite cells. *PLoS One*, **13**, e0195126.
- Song, Q. et al. (2020) Prediction of condition-specific regulatory genes using machine learning. *Nucleic Acids Res.*, **48**, e62.
- Tanaka, H. et al. (2017) Boolean modelling of mammalian cell cycle and cancer pathways. In: *The 2017 International Conference on Artificial Life and Robotics (ICAROB 2017)*, Seagaia Convention Center, Miyazaki, Japan. ALife Robotics Corp. Ltd, Oita, Japan, pp. 507–510.
- Thomas, R. and Kaufman, M. (2001) Multistationarity, the basis of cell differentiation and memory. II. Logical analysis of regulatory networks in terms of feedback circuits. *Chaos*, **11**, 180–195.
- Zou, Y.M. (2010) Modeling and analyzing complex biological networks incorporating experimental information on both network topology and stable states. *Bioinformatics*, **26**, 2037–2041.