Formal Analysis of Network Motifs Links Structure to Function in Biological Programs

Sara-Jane Dunn, Hillel Kugler, and Boyan Yordanov

Abstract—A recurring set of small sub-networks have been identified as the building blocks of biological networks across diverse organisms. These network motifs are associated with certain dynamic behaviors and define key modules that are important for understanding complex biological programs. Besides studying the properties of motifs in isolation, current algorithms typically evaluate the occurrence frequency of a specific motif in a given biological network compared to that in random networks of similar structure. However, it remains challenging to relate the structure of motifs to the observed and expected behavior of the larger, more complex network they are contained within. This problem is compounded as even the precise structure of most biological networks remains largely unknown. Previously, we developed a formal reasoning approach enabling the synthesis of biological networks capable of reproducing some experimentally observed behavior. Here, we extend this approach to allow reasoning over the requirement for specific network motifs as a way of explaining how these behaviors arise. We illustrate the approach by analyzing the motifs involved in sign-sensitive delay and pulse generation. We demonstrate the scalability and biological relevance of the approach by studying the previously defined networks governing myeloid differentiation, the yeast cell cycle, and naïve pluripotency in mouse embryonic stem cells, revealing the requirement for certain motifs in these systems.

Index Terms—Biological interaction networks, biological programs, network motifs, synthesis, satisfiability modulo theories (SMT), formal reasoning

1 Introduction

Network motifs [2], [29] are basic patterns of interactions that have been observed to recur throughout biological networks more frequently than in random networks with a comparable number of components and interactions. Accordingly, the same small set of network motifs appears to serve as the building blocks of biological networks for diverse organisms [3], [26], [41]. Each network motif can operate as an elementary circuit with a well-defined function, integrated within a larger network, and has a role in performing the required information processing [30]. Since the introduction of the concept of network motifs and the identification and experimental validation of initial instances [2], a wide range of additional motifs with new roles have been uncovered. Network motifs have been identified within transcriptional networks [25], [35], signaling networks [3], neuronal networks [33], and metabolic networks [32]. In addition to biological networks, recurring motifs have been identified in engineered systems, including electronic circuits and the world wide web [29].

The study of network motifs provides an attractive research direction towards understanding complex biological programs, and uncovering modularity and reusable patterns of computation in the design of biological circuits. While many of the associated problems are challenging, especially when dealing with large biological networks, a wide range of computational methods have been developed for motif identification [19], [38]. For example, novel motifs have been found by comparing the occurrence frequency of a sub-network within a known biological network to that in random networks of similar structure, while various graph methods have been applied to scan a network for specific motifs algorithmically. Substantial research effort has been devoted to dealing with the algorithmic challenges of network motif identification and the related graph algorithms [1], [27], [31], [38]. This has led to the development of a number of computational tools, including mfinder [19], MAVisto [34], NeMoFinder [9], FANMOD [39], Grochow-Kellis [15], Kavosh [18], MODA [40], NetMODE [24], Acc-MOTIF [28] and QuateXelero [20] (see also [38] for a review and detailed comparisons).

Verification techniques have also been applied to study network motifs. In [5], certain motifs and their dynamic properties were characterized using temporal logic, and parallel model checking was used to verify properties of networks with around ten components. Temporal logic provides a rich formalism for expressing biological behaviours and has been used for the specification and analysis of biological systems [6], [7], [8], [10], [13], [37]. In [17], approximate methods for analyzing gene regulatory networks were developed utilizing network motifs.

Following the identification of biologically-relevant motifs and the exploration of their dynamic properties in isolation,

- *S.-J. Dunn is with Microsoft Research, Cambridge CB1 2FB, United Kingdom, and also with the Wellcome-MRC Cambridge Stem Cell Institute, University of Cambridge, Cambridge CB2 1TN, United Kingdom. E-mail: sara-jane.dunn@microsoft.com.*
- *H. Kugler is with Bar-Ilan University, Ramat Gan 5290002, Israel. E-mail: hkugler@outlook.com.*
- *B. Yordanov is with Microsoft Research, Cambridge CB1 2FB, United Kingdom. E-mail: yordanov@microsoft.com.*

Manuscript received 14 Mar. 2019; revised 1 Oct. 2019; accepted 10 Oct. 2019. Date of publication 11 Nov. 2019; date of current version 3 Feb. 2021.
(Corresponding authors: Hillel Kugler and Boyan Yordanov.)
Recommended for acceptance by D. Safranek and M. Ceska.
Digital Object Identifier no. 10.1109/TCBB.2019.2948157

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

understanding how their presence or absence within a larger biological network defines that network's behavior becomes a central problem. This problem is compounded by the fact that the precise structure of such biological networks often remains largely unknown, due to noisy and sometimes irreproducible experimental data. This makes it challenging to search for motifs within the network or to explore the connections between a network's structure and its behavior.

Previously, we developed an SMT-based formal reasoning approach enabling the synthesis and analysis of biological networks (e.g., incorporating gene regulation, signaling, etc.) that were only partially known [12], [42]. The method, summarized briefly in Section 2.1, introduced the concept of an Abstract Boolean Network (ABN) as a formalism for describing discrete dynamic models of biological networks where the precise interactions or update rules are unknown. These models can be constrained with specifications of some required behavior, thereby providing a characterization of the set of all networks capable of reproducing experimental observations. Recently an approach that extends the RE:IN framework with temporal logic properties for specifying experimental observations has been developed [14], however the extension does not support any analysis of network motifs.

Here, we extend the approach from [12], [42] to enable automated reasoning about the requirement for specific network motifs as part of a biological network that is only partially known. This allows us to incorporate constraints relating to the structure of the network, represented as logical formulas over the presence or absence of different motifs, together with constraints about the network's dynamic behavior within the same framework. Our reasoning approach then allows us to draw conclusions about certain motifs being essential or disallowed for reproducing the required behavior, thus helping to explain how the observed behaviors arise from various motifs. We illustrate our approach by analyzing the motifs involved in sign-sensitive delay and pulse generation – distinct behaviors that have been associated with certain network structures and observed biological properties [25], [26]. We consider a generic 3-layered network topology that serves as a prototype for a variety of biological programs and find that under the qualitative, Boolean modeling formalism of [12], [42], positive feedback is required to implement both sets of dynamics. We also demonstrate the scalability and biological relevance of the approach by revealing motif requirements in three previously defined biological networks.

This paper extends the preliminary results presented recently in [22] in several ways. First, we generate the complete set of 2 and 3 component motifs as well as all motifs with 4, 5, and 6 components that have a specific feed-forward and feedback topology. Thus, we consider a total of 277 motifs, while only a subset of 15 motifs with 2 and 3 components was considered previously [22]. To handle this large number of motifs, we also extend the implementation of our method to first identify motifs that are always present or missing in a given network topology, before searching for motifs that are required or disallowed for enabling the observed behaviors. Finally, we extend the case studies to examine motifs in the biological programs governing myeloid differentiation and the cell cycle in budding yeast, as well as naïve pluripotency in mouse embryonic stem cells.

We envision that the method proposed in this paper will provide a powerful tool for researchers interested in exploring the structural properties of biological networks, and understanding how different motifs lead to various biological behaviors. In the future, this tool could support both theoretical and experimental studies in which the connections between network structure and function are to be explored, and could, for example, allow researchers to focus on the core, essential modules of biological networks.

2 Methods

In the following, we introduce some notation and summarize the approach from [42], which is implemented in the computational tool RE:IN (Section 2.1) and serves as a foundation for the extensions we propose in this paper (Sections 2.2 and 2.3).

2.1 Abstract Boolean Network Analysis

Following the notation from [42], an *Abstract Boolean Network* is a tuple $\mathcal{A} = (C, I, I^?, r)$, where

- C is the finite set of components,
- I is the set of definite (positive and negative) interactions between the components from C,
- $I^?$ is the set of possible (positive and negative) interactions, and
- r assigns a subset of regulation conditions (possible update functions) to each component from C.

ABNs are discrete, dynamic models suitable for studying biological systems, when often the existence of interactions between components is hypothesized, but not definitively known [12], [42]. An example of an ABN is illustrated in Fig. 1a. Each component $c \in C$ describes a different chemical signal, protein, gene, etc., that can exist in one of two states: active or inactive. The dynamics of the system are defined by the regulation condition assigned to each component, which serves as an 'update function' that specifies the state of the component at step $k+1$, given the state of all of its regulators (all components $c' \in C$ with interactions to c) at step k. Previously, we defined a set of 18 biologically inspired regulation conditions that correspond to different regulation mechanisms. For example, one regulation condition specifies that a gene is activated only if all its activators and none of its repressors are active, while another condition represents a mechanism where a single activator is sufficient to activate a gene regardless of the repressors. The details and precise definitions of all regulation conditions we consider are available in [12], [42]. Here we consider synchronous updates, such that deterministic trajectories emerge from each initial state, though RE:IN also allows the exploration of systems with asynchronous updates.

ABNs are abstract models because of the uncertainty in the precise network topology and regulation rules for each component. An ABN is transformed into a concrete Boolean Network (BN) by instantiating a subset of the possible interactions, discarding all other optional interactions, and assigning a specific regulation condition for each gene. By virtue of the unique combination of interactions and regulation conditions, different concrete models derived from the same ABN can have different dynamic behaviors.

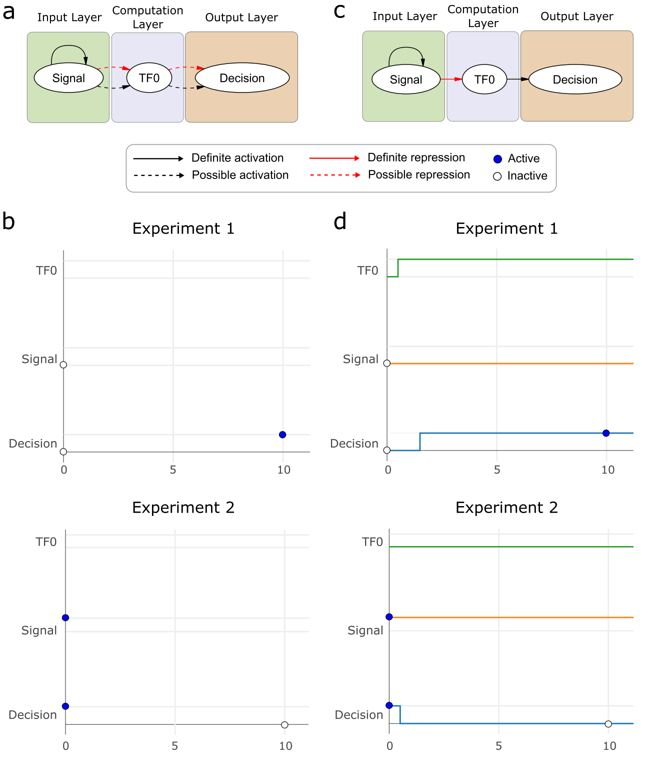

Fig. 1. *ABNs constrained against experimental observations.* **(a)** The generic network architecture we consider is comprised of three layers: input, computation, and output. The input and output layers each include a single component, while the number of computation components, n, can be varied (here $n = 1$). This simple ABN includes four optional interactions and one definite interaction. Formally, the ABN is defined as $\mathcal{A} = (C, I, I^?, r)$, where $C = \{\texttt{Signal}, \texttt{TF0}, \texttt{Decision}\}$, $I = \{(\texttt{Signal}, \texttt{Signal}, +)\}$, $I^? = \{(\texttt{Signal}, \texttt{TF0}, +), (\texttt{Signal}, \texttt{TF0}, -), (\texttt{TF0}, \texttt{Decision}, +), (\texttt{TF0}, \texttt{Decision}, -)\}$, and r allows all regulation conditions for each component. **(b)** Experimental constraints encode expected states along different network trajectories. Here, two experimental constraints are illustrated by specifying the initial state of the signal and decision components, and their state at step 10. **(c)** A single, concrete network that is consistent with the constraints in (b) is generated using our SMT-based approach. **(d)** Experiment trajectories from the concrete model in (c) illustrate how the experimental constraints are satisfied by this network.

The concept of a *Constrained Abstract Boolean Network (cABN)* was introduced in [42] as a formalism for describing a set of concrete BNs that are consistent with some experimentally observed biological behaviors. A cABN is defined in terms of an ABN, together with a set of constraints over the states of the components from C. These constraints encode experimental observations, where separate executions of the system correspond to different biological 'experiments'. For example, the observations encoded in Fig. 1b specify a biological program in which cells make a particular decision only in the absence of some signal. Experiment 1 requires that initially both the 'Signal' and 'Decision' components are inactive and 'Decision' is active at step 10. Similarly, Experiment 2 requires that both the 'Signal' and 'Decision' components are initially active and 'Decision' is inactive at step 10. These constraints limit the feasible assignments of regulation conditions and possible interactions such that all concrete networks from the cABN produce trajectories that reproduce all experimental observations. While the assignment of regulation conditions influences the dynamics of a concrete model and is a key part of cABN analysis, in this work we focus on the network topologies required to reproduce all experimentally observed behaviors.

cABN analysis was solved in [42] by encoding it as a Satisfiability Modulo Theories (SMT) problem. This enables the enumeration of individual concrete models that are consistent with the experimental observations. For example, the concrete BN from Fig. 1c is generated from the ABN in Fig. 1a and is consistent with the constraints from Fig. 1b. This is demonstrated using the trajectories visualized in Fig. 1d. The SMT-based approach from [42] also allows reasoning about hypotheses describing unknown biological behaviors to make novel predictions from all consistent models collectively, without the need to enumerate individual concrete networks.

In addition to the above, the analysis can reveal both required and disallowed interactions of the cABN. An interaction $i \in I^?$ is required if the experimentally observed behavior cannot be reproduced without it (i.e., all concrete models of the cABN include the interaction i). Similarly, an interaction is disallowed if including it in a concrete model means that the observed behavior can no longer be reproduced. The analysis of required and disallowed interactions yields insight into how network structures lead to certain dynamic behaviors, and is the starting point for the extensions we propose here.

In the following, we use set notation to denote the existence or non-existence of interactions in an ABN or cABN. For example, $(c, c', +) \in I$ denotes that a definite, positive interaction exists in $\mathcal{A}$, $(c, c', -) \in I^?$ denotes that an optional, negative interaction exists, and $(c, c', *) \notin I$ denotes that no definite interactions (i.e., the wild card $*$ stands for either $+$ or $-$) exist between c and c'.

2.2 Motif Assignment

Definition 2.1 (Motif). *A motif is a tuple $\mathcal{M} = \{C, I, I^?\}$, where C is the finite set of components, I is the set of definite and $I^?$ the set of possible interactions (similarly to the definition of ABNs).*

Examples of different motifs are illustrated in Fig. 2. In contrast to ABNs, motifs (Definition 2.1) are static networks, without regulation conditions (update functions) to make them dynamical systems. However, because interactions from $I^?$ are uncertain, motifs are abstract – a motif defined as in Definition 2.1 with a non-empty $I^?$ describes a set of $2^{|I^?|}$ concrete, static networks.

Definition 2.2 (Motif Assignment). *Given an ABN $\mathcal{A} = (C_{\mathcal{A}}, I_{\mathcal{A}}, I^?_{\mathcal{A}}, r)$ and a motif $\mathcal{M} = \{C_{\mathcal{M}}, I_{\mathcal{M}}, I^?_{\mathcal{M}}\}$ a motif assignment is a map $\theta : C_{\mathcal{M}} \to C_{\mathcal{A}}$.*

Note that since Definition 2.2 deals only with the topology of an ABN it also applies to cABNs - the additional constraints from the cABN do not affect the motif assignments.

Given an ABN $\mathcal{A} = (C_{\mathcal{A}}, I_{\mathcal{A}}, I^?_{\mathcal{A}}, r)$ and a motif $\mathcal{M} = \{C_{\mathcal{M}}, I_{\mathcal{M}}, I^?_{\mathcal{M}}\}$, let $\bar{I}_{\mathcal{A}} = I_{\mathcal{A}} \cup I^?_{\mathcal{A}}$ and $\bar{I}_{\mathcal{M}} = I_{\mathcal{M}} \cup I^?_{\mathcal{M}}$ denote the set of all interactions (definite and optional) in the ABN and the motif. Given a motif assignment $\theta : C_{\mathcal{M}} \to C_{\mathcal{A}}$, let $I_{\mathcal{A},\theta,\mathcal{M}} = \{(\theta(c), \theta(c'), *) \in I_{\mathcal{A}} \mid c \in C_{\mathcal{M}} \land c' \in C_{\mathcal{M}}\}$ denote the set of definite interactions from the ABN between components that the motif maps to. Similarly, let $I^?_{\mathcal{A},\theta,\mathcal{M}} = \{(\theta(c), \theta(c'), *) \in I^?_{\mathcal{A}} \mid c \in C_{\mathcal{M}} \land c' \in C_{\mathcal{M}}\}$ denote the set of optional interactions from the ABN between components that $\mathcal{M}$ maps to.[1]

[1] While, in general, the motif assignment θ is not invertible, $\theta^{-1}(c)$ and $\theta^{-1}(c')$ can be defined for the interactions $(c, c', *) \in I_{\mathcal{A},\theta,\mathcal{M}}$ and $(c, c', *) \in I^?_{\mathcal{A},\theta,\mathcal{M}}$.

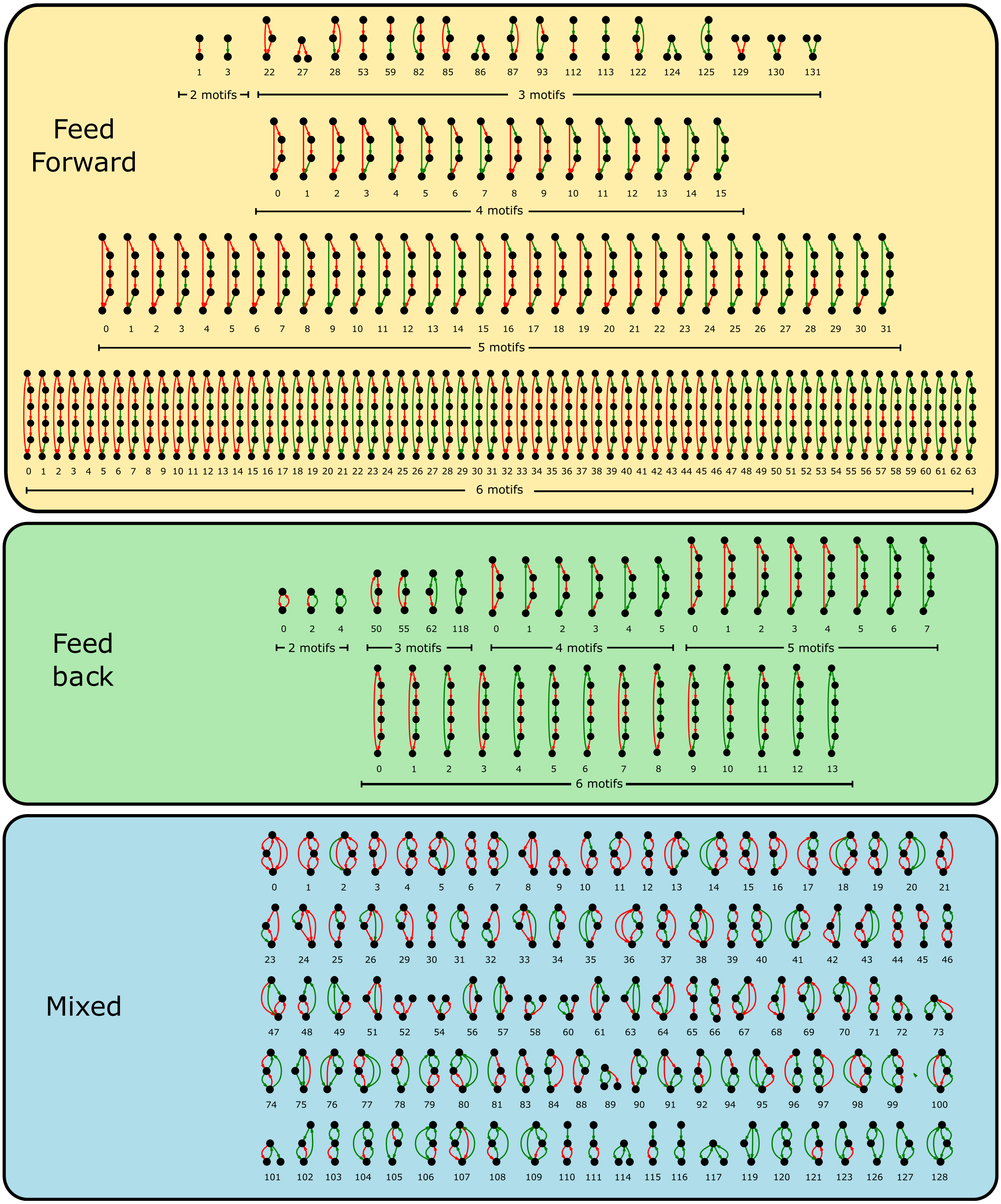

Fig. 2. *The set of network motifs we define and use to analyze biological programs.* The network motifs are sub-divided according to the number of components, and then into feed-forward, feedback, and 'mixed' categories. Red and green arrows respectively indicate negative and positive interactions. Mixed motifs contain both feed-forward and feedback elements. An optional positive and negative interaction from the 'Context' component to every motif component, from every motif component to 'Context', as well as optional positive and negative self regulations at each motif component are also included, but not shown in the figure. This ensures that the motif of interest can be identified in the ABN regardless of how the motif components are connected within the network (see Sec. 2.4). Note that the index of each motif, which we use for referencing, is automatically generated and has no bearing on the order in which they are depicted in this figure. Indexing starts from 0 for motifs that share the same number of components.

Note also that we allow interactions from a component to itself, i.e., $(c, c, +)$, which is termed positive self-regulation and $(c, c, -)$, which is termed negative self-regulation (self-regulation is often also called auto regulation).

Definition 2.3 (Valid Motif Assignment). *A given motif assignment θ between ABN $\mathcal{A}$ and motif $\mathcal{M}$ is valid if and only if*

1) $\forall (c, c', +) \in I_{\mathcal{A},\theta,\mathcal{M}} \,.\, (\theta^{-1}(c), \theta^{-1}(c'), +) \in \bar{I}_{\mathcal{M}}$,

2) $\forall (c, c', -) \in I_{\mathcal{A},\theta,\mathcal{M}} \,.\, (\theta^{-1}(c), \theta^{-1}(c'), -) \in \bar{I}_{\mathcal{M}}$,

3) $\forall (c, c', +) \in I_{\mathcal{M}} \,.\, (\theta(c), \theta(c'), +) \in \bar{I}_{\mathcal{A}}$, *and*

4) $\forall (c, c', -) \in I_{\mathcal{M}} \,.\, (\theta(c), \theta(c'), -) \in \bar{I}_{\mathcal{A}}$.

The conditions from Definitions 2.3.1-2.3.4 ensure that the motif components are assigned to ABN components in such a way that the interactions match. In other words, each definite (positive or negative) interaction in the ABN (between components that the motif maps to) matches an interaction (definite or optional) in the motif (Definition 2.3.1-2.3.2) and each definite interaction in the motif matches an interaction in the ABN (Definition 2.3.3-2.3.4).

Given an optional interaction $(c, c', *) \in I^?_{\mathcal{A}}$, let $\mathcal{I}^*_{c,c'} \in \mathbb{B}$ denote the Boolean choice variable representing whether the interaction is included in a concrete model (see [42] for details of the SMT encoding of ABNs and cABNs). Asserting that $\mathcal{I}^*_{c,c'}$ is true can be interpreted as modifying the ABN such that $(c, c', *) \notin I^?_{\mathcal{A}}$ and $(c, c', *) \in I_{\mathcal{A}}$ (i.e., ensuring that the interaction is definitely present). Similarly, asserting that $\mathcal{I}^*_{c,c'}$ is false can be interpreted as modifying the ABN such that $(c, c', *) \notin I^?_{\mathcal{A}}$ but $(c, c', *) \notin I_{\mathcal{A}}$ (i.e., ensuring that the interaction is definitely absent).

The notion of a valid motif assignment (Definition 2.3) is sufficient to guarantee that the components of the motif are mapped to components of the ABN in such a way that all definite interactions are matched. However, it is possible that optional interactions of the ABN map to definite interactions of the motif or do not match any motif interactions. Therefore, while the interactions of the ABN match that of the motif, it is not possible to guarantee that every concrete network represented by the ABN matches the motif. The additional constraints defined in the following ensure that this is indeed the case.

Definition 2.4 (Motif Assignment Constraints). *Given a motif assignment θ between ABN $\mathcal{A}$ and motif $\mathcal{M}$, the motif assignment constraints are*

$$\begin{aligned}\mathcal{C}_\theta = {} & \{\mathcal{I}^+_{c,c'} \mid (c, c', +) \in I_M \wedge (\theta(c), \theta(c'), +) \in I^?_{\mathcal{A}}\} \cup \\ & \{\mathcal{I}^-_{c,c'} \mid (c, c', -) \in I_M \wedge (\theta(c), \theta(c'), -) \in I^?_{\mathcal{A}}\} \cup \\ & \{\neg\mathcal{I}^+_{c,c'} \mid (c, c', +) \notin \bar{I}_M \wedge (\theta(c), \theta(c'), +) \in I^?_{\mathcal{A}}\} \cup \\ & \{\neg\mathcal{I}^-_{c,c'} \mid (c, c', -) \notin \bar{I}_M \wedge (\theta(c), \theta(c'), -) \in I^?_{\mathcal{A}}\}.\end{aligned}$$

The additional constraints from (Definition 2.4) assert that an optional interaction in the ABN that matches a definite interaction of the motif is always included. Similarly, an optional ABN interaction that does not match any motif interaction is never included. These additional constraints guarantee that the interactions of all concrete networks of the ABN match those of the motif, under the given motif assignment.

2.3 Motif Constraints

The motif assignment constraints (Definition 2.4) ensure that a given motif $\mathcal{M}$ is implemented in all concrete networks of an ABN $\mathcal{A}$ between the specific components defined by the motif assignment θ. In general, however, we are interested in guaranteeing that motif $\mathcal{M}$ is implemented in the ABN $\mathcal{A}$ by any of its components, rather than the specific set of components specified by θ.

Definition 2.5 (Motif Constraints). *Given an ABN $\mathcal{A}$ and a motif $\mathcal{M}$, the motif constraints $\mathcal{C}_{\mathcal{M},\mathcal{A}}$ are defined in terms of the motif assignment constraints (Definition 2.4) as $\mathcal{C}_{\mathcal{M},\mathcal{A}} = \bigvee_{\theta\in\hat{\Theta}} \mathcal{C}_\theta$, where $\hat{\Theta}$ is the set of valid motif assignments between $\mathcal{A}$ and $\mathcal{M}$ (Definition 2.2).*

The motif constraints from Definition 2.5 guarantee that motif $\mathcal{M}$ is implemented by every concrete network of ABN $\mathcal{A}$, even though the precise components used to implement $\mathcal{M}$ might differ.

Given an ABN $\mathcal{A}$ and a set of motifs, logical formulas (e.g., $\neg\mathcal{M}$, $\mathcal{M} \vee \mathcal{M}'$, $\mathcal{M} \wedge \mathcal{M}'$, etc) could be constructed and interpreted by replacing each motif $\mathcal{M}$ with its corresponding motif constraints $\mathcal{C}_{\mathcal{M},\mathcal{A}}$.

2.4 Implementation

As part of the RE:IN framework [12], [42], a high-level, domain specific language was proposed for describing cABNs by defining the sets of components, interactions, and associated experimental observations. We implement the methods described in Sections 2.2 and 2.3 as an extension of RE:IN, which enables the reasoning about cABNs with additional structural constraints on the presence or absence of various motifs.

Currently, the generation of motif constraints (Definition 2.5) is implemented as a pre-processing step using a straightforward, exhaustive algorithm, where all motif assignments are first generated and then filtered to preserve only the valid ones using the conditions from Definition 2.3. Various cABN analysis problems are then encoded and solved using an SMT solver as shown previously [12], [42], while the additional motif constraints are also incorporated.

Two notable modifications are introduced to the method described in Sections 2.2 and 2.3 for improved usability. First, when the name of a motif component matches the name of a cABN component, no other assignments are considered for that component. This enables the specification of partially known motifs, where the mapping of some of the motif components to the cABN components is given. Second, a dummy 'Context' component is always included within the set of motif components. The Context component represents any network component that is not part of the motif, once the motif is mapped to network components. Given a motif assignment, the 'Context' component matches any cABN component that is not already mapped to by the motif. This provides additional control in specifying how a motif could be implemented as part of the cABN's network. For example, including optional positive and negative interactions from 'Context' to every motif component and vice versa does not impose additional constraints on the motif's implementation. Without any additional 'Context' interactions, on the other hand, the motif can only be fully isolated and disconnected from all other components of the cABN.

2.5 Reasoning About Motifs

Combining the previously-developed SMT-based reasoning strategies [12], [42] with an encoding of the motif constraints from Section 2.3 enables automated reasoning about the structural (motif) properties of a network, together with the requirements about reproducing certain dynamic behavior. Details of the encoding and application of an SMT solver to reason about cABNs can be found in [42] and the encoding of the motif constraints follows directly from Definition 2.5 and the definitions in Section 2.2.

Among the different analysis questions this method could support, in this work we focus specifically on identifying required (essential) and disallowed motifs across all concrete models of the cABN. A motif $\mathcal{M}$ is required if the experimentally observed behavior could not be reproduced without it (i.e., all concrete, consistent networks contain the motif), while it is disallowed if enforcing that the motif be present in the network guarantees the observed behavior can no longer be reproduced (i.e., none of the consistent networks contain the motif). These hypotheses can be tested as follows. Applying the SMT analysis of an ABN, if no concrete models are identified with the constraint $\mathcal{M}$ (the motif is present in the cABN) then the motif is disallowed. If, on the other hand, no concrete models are identified with the constraint $\neg\mathcal{M}$ (the motif is not present in the network), then the motif is essential.

We also distinguish between trivially and non-trivially required (disallowed) motifs, where the former can be identified solely from the topology of the ABN. For example, a motif formed by definite interactions is always present in a given network and will be identified as (trivially) required. Similarly, if a given network does not include suitable definite or optional interactions to implement a given motif, the motif will be (trivially) disallowed.

In addition, it is possible to identify motifs of combinations that are not trivial, but are required or disallowed even when no constraints are imposed about the behaviour of a network. For example, such a situation could arise when including a given (optional) interaction in the network would lead to a motif being implemented by a given set of components, while excluding the same interaction leads to the motif being implemented by different components. Then, the motif is required but this property cannot be inferred directly from definite interactions in the network (i.e., the motif is not trivially required). We refer to such motifs as structurally required or disallowed to highlight the fact that these properties are not imposed by the system's behaviors and to distinguish these motifs from the (functionally) required and disallowed ones.

To deal with these classes of motifs, we extend the implementation of our method to first identify trivially required and disallowed motifs directly from the topology of a given network. Next, we consider a modified cABN, where all constraints are omitted to identify motifs that are required or disallowed structurally, but not due to the system's behaviour. Finally, once all trivial motifs have been discounted, we analyze the cABN together with all constraints to identify motifs that are functionally required and disallowed.

3 Results

To illustrate the analysis method proposed in Section 2, we study the importance of specific motifs for biological networks.

We consider the complete set of 2- and 3- component motifs by generating all non-isomorphic, connected networks. In addition, we include the set of 4-, 5-, and 6- component motifs that have a specific feed-forward and feedback topology. This gives rise to a set of 277 motifs that we consider for all subsequent studies (Fig. 2). To make specification of motifs more flexible, we include an optional positive and negative self loop at each motif component, as well as positive and negative interactions between each motif component and the 'Context' in both directions. This ensures that the motifs can be identified as part of the biological networks regardless of the context or the presence of self-regulation.

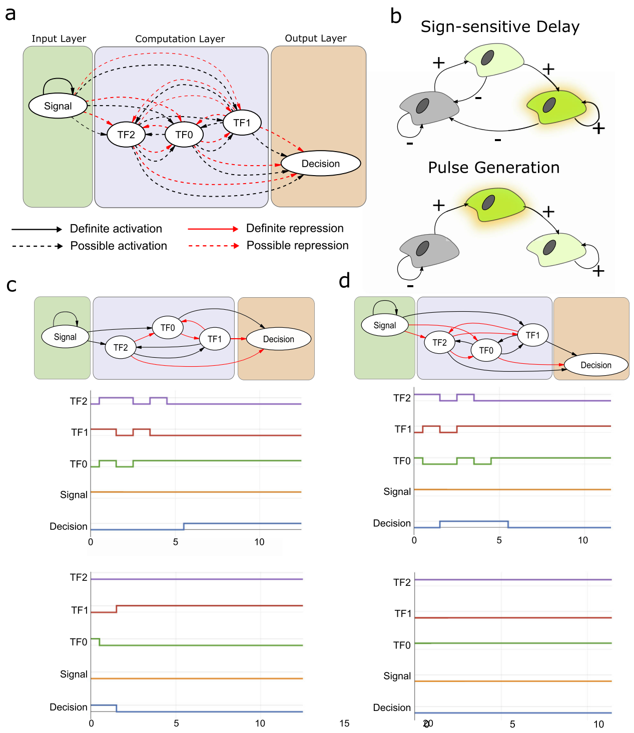

Fig. 3. *A generic network architecture generates biological programs implementing either sign-sensitive delay or pulse generation.* (**a**) The ABN we consider is comprised of three computation components, with optional positive and negative interactions between them. (**b**) The sign-sensitive delay and pulse generation behaviors are represented graphically as transitions between different cellular states. Signs on the edges indicate the presence (+) or absence (-) of the input signal, while the output is active only in the cell type shown in green (right-most cell). (**c**) An example of a biological program implementing sign-sensitive delay has the characteristic delay during activation (top trajectories) but responds faster during deactivation (bottom trajectories). (**d**) An example of a biological program implementing pulse generation produces a single pulse of the output when the input signal is present (top trajectories), while the output remains inactive when the signal is not present (bottom trajectories).

With the set of motifs from Fig. 2, we first consider a generic network architecture (Fig. 3a) composed of an input, computation, and output layer, which serves as a prototype for many biological programs (Section 3.1). We study the models consistent with this network topology that give rise to two distinct dynamic behaviors (Fig. 3b) and identify the motifs that are required or disallowed in all of these models. Then, in Section 3.2 we apply our motif analysis method to case studies of biological networks [11], [42], demonstrating the scalability and biological relevance of the approach. A summary of the number of (trivially) required and disallowed motifs we identify is presented in Table 1.

3.1 Biological Program Prototype

We construct a simple abstract network topology in order to explore how various motifs give rise to different dynamic behaviors. The network has a single input component that represents a biochemical signal and a single output (readout)

TABLE 1
Summary of the Number of Motifs That are Required or Disallowed in Each Case Study

Model	Present	Absent	Required	Disallowed	Inconclusive
Myeloid	13	173	2	8	81
Yeast	1	126	7	60	83
Stem cell	4	97	0	0	176
Pulse generation	0	124	5	118	30
Sign-sensitive delay	0	124	2	92	59

Trivially required motifs are always present in the network and trivially disallowed motifs are always absent.

component that might represent a biochemical signal affecting a downstream process, or a particular cellular decision (e.g., to differentiate, divide, etc). Information processing is performed in the 'computation' layer, which includes a number of components. While all interactions in the network are unknown, we assume that information flows from the input layer through the computation layer into the output layer. As a result, we consider a network with a densely connected computation layer (possible positive and negative interactions between each pair of computation components). The input (signal) component might affect any of the computation components, so possible positive and negative interactions from the input (signal) to all computation components are included. Similarly, possible positive and negative interactions from each computation component to the output (decision) component are included. A definite self-activation is included for the signal to guarantee that once set at the beginning of computation, its value does not change, but no other self-regulation interactions are allowed. The resulting network architecture with $n = 3$ computation components is visualized in Fig. 3a. In all subsequent analysis, we impose the additional constraint that a positive and a negative interaction between the same components in the same direction are never included together in concrete models.

Sign-Sensitive Delay. The first dynamic behavior we consider requires that a system produce an output in response to some input (e.g., the 'Decision' component becomes active if and only if 'Signal' is present in Fig. 3a). However, while the effect on the output is immediate when the signal is withdrawn, there is a delay (with a delay longer than a specified value) on activation of the output when the signal is supplied.[2] Due to the asymmetric response to changes in the input signal, the behavior is called sign-sensitive delay [25], [26] and has been shown to have a role for making decisions based on noisy inputs by filtering out fluctuations in input stimuli [25], [26].

We encode the requirement for a sign-sensitive delay as depicted in Fig. 3b. When no signal is present, the system can stabilize in a state where the output is inactive (shown in gray). When the signal is supplied, a transition to an intermediate state occurs (light green), although the output is still not activated. Withdrawing the signal at this point resets the system to the initial state, while continuous application of the signal causes a transition to a state where the output is active (green). This active state is stable as long as the signal is present, but withdrawal of the signal leads quickly back to the initial, inactive state. The number of steps between activation of the signal and activation of the output is the delay.

We find that for delays greater than a single step, a network with at least $n = 3$ computation components is required, and with $n = 3$, delays of up to 4 steps can be produced. An example of a concrete network implementing this behavior is shown in Fig. 3c, but many such networks consistent with the cABN from Fig. 3a exist, involving a variety of network motifs. To investigate further the network structures capable of producing sign-sensitive delays, we defined a number of feed-forward, feed-back and mixed motifs shown in Fig. 2.

We found that 92 different motifs were disallowed (possibly due to the limited number of computation nodes), and two motifs (2-motifs *1* and *3*) were required for producing a sign-sensitive delay of 4 steps (see S-XI, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TCBB.2019.2948157 for detailed results). The required motifs correspond to simple feed-forward elements consisting of a single positive or negative interaction. To identify more complex structural patterns, we then considered pairs of motifs (e.g., by testing $\mathcal{M} \wedge \mathcal{M}'$ and $\neg\mathcal{M} \wedge \neg\mathcal{M}'$), which were individually neither required nor disallowed (e.g., results were inconclusive). We found that, while many pairs were jointly disallowed, four pairs (2-motifs (*0*, *4*), and 3-motifs (*112*, *53*), (*112*, *59*), and (*113*, *53*)) were jointly required (see Table S-XXVI, available online). In particular, the required pair of 2-motifs (*0*, *4*) corresponds to a pair of positive feedback loops. This indicates that some form of positive feedback (either 2-motif *0* or *4*) is essential for implementing a 4-step sign-sensitive delay in a network with $n = 3$ computation components, but one of these motifs can be substituted for the other.

Pulse Generation. The second dynamic behavior we consider requires that a system produce a transient output pulse in response to some input. We encode this behavior as depicted in Fig. 3b. While no signal is present, the system remains stably in a state where the output is inactive (shown in gray). When the signal is supplied, a transition to an intermediate state occurs (green) where the output is activated. Further application of the signal causes a transition to a state where the output is no longer active (light green). Currently, we do not consider the case where the system 'resets' (i.e., withdrawal of the signal causing a transition to the initial state). The number of steps during which the output is active upon supplying the signal is the pulse width.

We find that for pulse widths greater than a single step, a network with at least $n = 3$ computation components is required, and with $n = 3$, pulse widths up to 4 steps can be produced. An example of a concrete network implementing this behavior is shown in Fig. 3d.

[2] Depending on the exact implementation, the delay could instead be observed when the signal switches from active to inactive, but this variation of a sign-sensitive delay is not considered here.

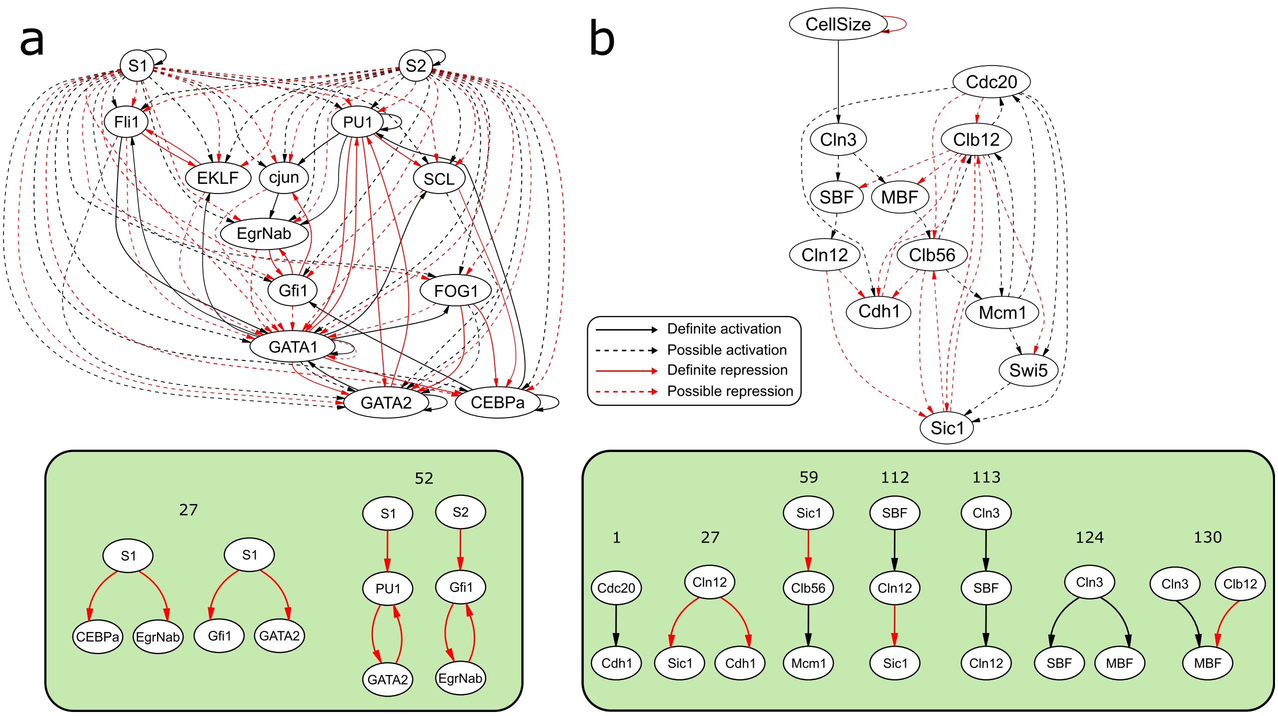

Fig. 4. *Required Network motifs in realistic biological networks.* (**a**) (Top) The ABN downstream of two input signals, S_1 and S_2, hypothesized to govern myeloid differentiation. (Bottom) Two 3-motifs are required, examples of which are illustrated. (**b**) (Top) The ABN hypothesized to govern the yeast cell cycle. (Bottom) Seven motifs are required, examples of which are illustrated.

As in the previous case study, we were interested in exploring how the motifs from Fig. 2 affect the capacity for pulse generation. We found that 118 different motifs were disallowed, while 5 motifs (2-motifs *0*, *1*, and *3*, and 3-motifs *113* and *53*) were required. 2-motif *0* corresponds to a positive feedback loop and the rest of the required motifs correspond to simple feed-forward cascades. This indicates that a positive feedback motif between two components, implemented specifically through two negative interactions, is essential for implementing a generator for pulses of width 4 in a network with $n = 3$ computation components.

3.2 Case Studies: Motifs in Biological Networks

The sign-sensitive delay and pulse generation examples demonstrate the utility of our approach in revealing how different network motifs give rise to different dynamic behaviors within a relatively simple network. Here we apply our analysis to realistic biological networks that have been defined previously: the network governing myeloid differentiation [21], [42], the network governing the yeast cell cycle [23], [42], and the network governing installation and maintenance of naïve pluripotency in mouse embryonic stem cells [11], [12].

Previously, we used RE:IN to study the biological program governing differentiation in hematopoiesis, building on an asynchronous Boolean network of 11 myeloid transcription factors (TFs) first defined and explored by Krumsiek et al. [21]. More recent experimental evidence suggests that cell fate decisions in this system are driven by upstream regulators, such as extracellular signals, and not by the asynchronous interactions of critical TFs [16]. Therefore, rather than assume that the precise order of network updates determines the fate of cell differentiation, we hypothesized that two signals upstream of the myeloid TFs could deterministically specify cell fate (Fig. 4a, top) [42]. We included two potential signals upstream of the Krumsiek et al. network, and found that specification into the four different myeloid lineages was possible under a synchronous update scheme. Later work suggests that this hypothesis has experimental support, as shown in [16]. Here, we asked whether specific motifs are required downstream of the input signals in order to specify the myeloid lineages, and whether these motifs are common to all concrete models that are consistent with the behavioral specifications. It is important to note that there are a number of components in the myeloid ABN that have self-activation loops. Consequently, we assumed that self-loops can occur within the context of each tested motif.

First, we eliminated motifs that were either trivially required or disallowed in the ABN - these are motifs that are either always present given the set of definite interactions, or not feasible given the set of definite and possible interactions (see Tables 1 and S-I, available online). We found that 13 motifs are trivially required, and 173 are trivially disallowed. For example, 3-motif *110* (Fig. 2) is trivially required, as it is instantiated in the definite interactions between CEBPa, Gfi1 and EgrNab, while 3-motif *117* is disallowed as this pair of feedback loops is not present the topology of either definite or possible interactions.

We identified two motifs that are non-trivially required, each comprising three components: *27* and *52* (Fig. 2). Moreover, we could identify that these motifs arise in the concrete models only between certain sets of components. Two examples of where these required motifs arise are shown in Fig. 4a (bottom). Furthermore, we identified that 8 motifs are non-trivially disallowed (Table S-I, available online). The remaining 81 motifs will appear in at least one, but not all, concrete models that are consistent with the behavioral specifications (‘inconclusive’). This analysis is powerful in that it reveals candidate targets of putative inputs, and how the presence / absence of these signals can ultimately determine cell state.

We similarly investigated the requirement for specific motifs in the network governing the cell cycle in budding yeast [23]. This network is required to execute a trajectory through the temporally ordered phases of the cell cycle if it is perturbed from the stationary G1 phase. In [42], we constructed an ABN between 11 components downstream of a cell size input signal to investigate the concrete BNs that are consistent with the expected cell cycle dynamics (Fig. 4b, top). This ABN has 1 trivially required motif, *3*, which is simply the activation of Cln3 by the cell size input, and 126 trivially disallowed motifs (see Tables 1 and S-XXI, available online). Our motif analysis revealed that 7 motifs are non-trivially required across the set of concrete BNs. Examples of where these motifs are instantiated are shown in Fig. 4b, bottom. Furthermore, we learned that 60 motifs are non-trivially disallowed, while the remaining motifs are present in some, but not all, concrete networks.

The final case study we explored was the biological program governing naïve pluripotency: the property uniquely exhibited by embryonic stem cells (ESCs) to generate all adult cell types as well as the germline. This program was first considered in [12], where a cABN was derived that was shown to be consistent with observations of maintenance of the naïve state. We explored required and disallowed motifs in this cABN in [22]. However, given that this program was further refined in a biological study in [11], where it was found also to explain the dynamics of installation of naïve pluripotency, we chose to investigate the refined cABN. While a number of motifs are trivially required or disallowed, we found that there were no non-trivially required or disallowed motifs (see Tables 1 and S-XVI, available online).

These examples demonstrate how our reasoning approach scales to complex networks that explain critical aspects of cellular decision-making, and can reveal essential elements of biological programs that are required to explain observed behavior. Further analysis could reveal whether the motifs we have identified are consistent with additional experimental observations for each cell decision under study, when available, or if further constraints could expose additional requirements for motif structures. Lastly, it would be of interest to study whether the required motifs are found to repeat throughout networks governing similar biological behaviors, revealing elements of biological computation that are conserved across biology.

4 Discussion

The SMT-based formal reasoning approach from [12], [42] allows us to encode dynamic Boolean models of biological networks where the precise set of interactions and regulation conditions for each component are unknown. These models can be constrained against specifications of experimental observations in order to identify concrete models capable of reproducing the required behavior, or to test novel hypotheses without selecting a particular concrete model from the set of models consistent with the experiments.

Certain structural constraints can also be handled directly in this reasoning approach. For example, assigning an interaction as definite guarantees that only concrete models that incorporate that interaction are considered. Similarly, removing an interaction as either optional or definite guarantees that none of the considered models incorporate this interaction. However, expressing more complex structural properties, such as those required for reasoning about motifs, is challenging using the approach from [12], [42]. The extensions we propose in this work allow for complex properties describing the presence or absence of arbitrary motifs to be specified and tested, even when the network being studied is partially unknown. This leads to a natural framework for incorporating rich structural constraints and jointly reasoning about motifs and dynamic properties.

The degree to which a network is constrained, either by reducing the number of optional interactions or by specifying additional behaviors that must be reproduced, limits the possibility of implementing motifs between different components and could lead to more predictions about essential or disallowed motifs. In contrast, larger, less-constrained biological networks can achieve the same behavior in different ways and certain motifs would no longer be required. How precise our knowledge of an ABN is (i.e., how constrained the network is) also affects the distinction between trivial and non-trivial motifs. If we have few optional interactions and many definite interactions, a greater number of motifs can typically be identified as trivial, which does not mean that they are not biologically meaningful. Thus, an alternative interpretation to the distinction between trivial / non trivial is the terminology of enforced / inferred.

While the problem of identifying motifs does not scale favorably to large networks [4], [36], in [12] it was demonstrated that relatively small networks of core components can explain a rich set of biologically-relevant behaviors and cellular decisions. Therefore, even the straightforward, exhaustive algorithm implemented currently to generate motif constraints (Section 2.4) proves suitable for examining networks of biological significance. Indeed, for small motifs (e.g., with 2 and 3 components) the computation times for generating motif constraints are negligible compared to the task of verifying whether any concrete models exist that satisfy all constraints (see detailed results in Supplementary Material, available online). All analysis reported in this paper was accomplished on the order of seconds to minutes for simple networks (Section 3.1) as well as for the myeloid differentiation and yeast cell cycle biological networks. Due to the number of experimental observations incorporated as part of the model, motif analysis of the stem cell pluripotency network (Section 3.2) provided a more challenging but still feasible problem (see detailed computation times in Supplementary Material, available online).[3] Many of the algorithmic advances towards more efficient identification of motifs in large networks [19], [27], [38] could be adapted as part of the pre-processing step of motif constraint generation.

A possible limitation of the proposed method for certain applications is that the cABN models being analyzed are qualitative (Boolean) and discrete. Dynamic behaviors associated with motifs or networks that require more detailed modeling assumptions cannot be handled directly at present. Thus,

[3] Analysis was performed on *Standard f2s v2* Azure VMs (3.4GHz Intel Xeon Platinum 8168 CPU, 4GB memory).

certain properties, for example relating to noise propagation and attenuation, or precise timing of signals, are not currently supported by our motif analysis. The models we considered as part of the presented case studies were also deterministic due to the synchronous update semantics we assume, although asynchronous updates are also supported by the method [12], [42]. When analysing models with asynchronous updates, especially without additional constraints guaranteeing that all components are updated fairly, often more behaviours are possible but fewer predictions can be made.

Still, a number of interesting biological questions can be framed in terms of the analysis of structural, motif-based properties of partially known networks with respect to the dynamic behaviors they produce. In this work, we focused specifically on identifying motifs that are essential or disallowed for producing certain behaviors. The proposed approach can also be used for more in-depth studies, for example to identify whether a required motif must always involve specific components in the network, or whether it arises in different locations in alternative models, as illustrated in the case studies from Section 3.2. This is achieved by testing whether concrete models without a given motif involving a particular network component exist. Further detailed studies could also explore not just the presence or absence of motifs, but also how these motifs must be connected to the rest of the network (e.g., for a particular flow of information between components) to achieve the required behavior, by exploring different combinations of optional and definite interactions between motif components and the 'Context'.

It is important to note that the motif analysis proposed here provides additional insight beyond that obtained from the identification of required and disallowed interactions (an analysis approach already possible with the methods from [12], [42]). Intuitively, if a given motif must be implemented by specific components of a network, then all the motif's interactions would be identified as required. However, identifying motifs that are essential but appear in different places in different concrete networks that are consistent with experimental observations, reveals deeper connections between structure and behavior. The approach we propose provides direct insights about network motifs rather than individual interactions. Thus, a motif can also be identified as required right away, rather than requiring additional network analysis in cases when all the motif's interactions are required in the network.

In this paper, we focused on network topologies (i.e., the presence or absence of motifs) as a step towards linking the structure and function of biological programs. Besides topologies, our analysis revealed the regulation conditions for each gene in a network required to reproduce the experimentally observed behaviours of a program. Indeed, the same network topologies and motifs could have different dynamic behaviours when different regulation conditions are assigned to the motifs components. Investigating the role of regulation conditions in addition to topologies, for example by considering the same motif with different update rules, could provide finer understanding of the links between structure and function, and is an interesting direction for future work.

Finally, the case studies presented here illustrate how the proposed methodology can be applied to theoretical studies of the properties and requirements for different motifs (Section 3.1) and demonstrate that the approach scales to, and provides useful insights about, realistic biological networks (Section 3.2). For each model, we identify several motifs as required from the set of two and three component motifs, which we explored exhaustively. In addition, we identify many motifs as disallowed, including some from the set of motifs with up to six components, which we explored only partially. Extending these studies and experimentally validating the motif requirements in these networks is a direction for future research.

5 Summary

To deal with the challenge of studying the connections between the structure and function of biological networks, even when these networks are only partially understood, we extended the SMT-based reasoning approach RE:IN from [12], [42]. The proposed method involves the algorithmic generation of motif constraints, and encoding the requirement that a given motif is present in some cABN. Rich structural requirements can then be incorporated in addition to the functional properties encoded as part of the cABN through logical formulas over such motif constraints. We illustrated the method by predicting that certain motifs are essential or disallowed for producing sign-sensitive delay and pulse generation in a network representing a prototype of biological programs, and in the biological networks governing myeloid differentiation, the yeast cell cycle, and stem cell pluripotency. The proposed method enables the study of network motifs in the context of partially unknown, abstract networks and can support future theoretical and experimental studies, where reasoning about network structure and function in the same framework is essential.

Acknowledgments

The research was partially supported by the Horizon 2020 research and innovation programme for the Bio4Comp project under Grant agreement number 732482 and by the ISRAEL SCIENCE FOUNDATION (grant No. 190/19).

References

[1] N. Alon, P. Dao, I. Hajirasouliha, F. Hormozdiari, and S. C. Sahinalp, "Biomolecular network motif counting and discovery by color coding," *Bioinf.*, vol. 24, no. 13, pp. i241–i249, 2008.

[2] U. Alon, *An Introduction to Systems Biology: Design Principles of Biological Circuits*. Boca Raton, FL, USA: CRC Press, 2006.

[3] I. Amit ***et al.***, "A module of negative feedback regulators defines growth factor signaling," *Nature Genetics*, vol. 39, no. 4, 2007,Art. no. 503.

[4] L. Babai and E. M. Luks, "Canonical labeling of graphs," in *Proc. 15th Annu. ACM Symp. Theory Comput.*, 1983, pp. 171–183.

[5] J. Barnat *et al.*, "From simple regulatory motifs to parallel model checking of complex transcriptional networks," *Pre-proceedings Parallel Distrib. Methods Verification Budapest*, pp. 83–96, 2008.

[6] G. Batt *et al.*, "Validation of qualitative models of genetic regulatory networks by model checking: Analysis of the nutritional stress response in *escherichia coli*," *Bioinf.*, vol. 21, pp. 19–28, 2005.

[7] N. Chabrier and F. Fages, "Symbolic model checking of biochemical networks," in *Proc. 1st Int. Workshop Comput. Methods Syst. Biol.*, 2003, pp. 149–162.

[8] N. Chabrier-Rivier, M. Chiaverini, V. Danos, F. Fages, and V. Schächter, "Modeling and querying biomolecular interaction networks," *Theor. Comput. Sci.*, vol. 325, no. 1, pp. 25–44, 2004.

[9] J. Chen, W. Hsu, M. L. Lee, and S.-K. Ng, "NeMoFinder: Dissecting genome-wide protein-protein interactions with meso-scale network motifs," in *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2006, pp. 106–115.

[10] E. Dubrova, M. Teslenko, and L. Ming, "Finding attractors in synchronous multiple-valued networks using SAT-based bounded model checking," in *Proc. 40th IEEE Int. Symp. Multiple-Valued Logic*, 2010, pp. 144–149.

[11] S.-J. Dunn, M. A. Li, E. Carbognin, A. G. Smith, and G. Martello, "A common molecular logic determines embryonic stem cell self-renewal and reprogramming," *EMBO J.*, vol. 38, 2018, Art. no. e100003.

[12] S.-J. Dunn, G. Martello, B. Yordanov, S. Emmott, and A. Smith, "Defining an essential transcription factor program for naïve pluripotency," *Sci.*, vol. 344, no. 6188, pp. 1156–1160, 2014.

[13] S. Eker, M. Knapp, K. Laderoute, P. Lincoln, J. Meseguer, and K. Sonmez, "Pathway logic: Symbolic analysis of biological signaling," in *Biocomputing 2002*. Singapore: World Scientific, 2001, pp. 400–412.

[14] J. Goldfeder and H. Kugler, "Temporal logic based synthesis of experimentally constrained interaction networks," in *Proc. 1st Int. Symp. Mol. Logic Comput. Synthetic Biol.*, 2018, pp. 89–104.

[15] J. A. Grochow and M. Kellis, "Network motif discovery using subgraph enumeration and symmetry-breaking," in *Proc. 11th Annu. Int. Conf. Res. Comput. Mol. Biol.*, 2007, vol. 4453, pp. 92–106.

[16] P. S. Hoppe *et al.*, "Early myeloid lineage choice is not initiated by random PU.1 to GATA1 protein ratios," *Nature*, vol. 535, pp. 299–302, 2016.

[17] S. Ito, T. Ichinose, M. Shimakawa, N. Izumi, S. Hagihara, and N. Yonezaki, "Formal analysis of gene networks using network motifs," in *Proc. Int. Joint Conf. Biomed. Eng. Syst. Technol.*, 2013, pp. 131–146.

[18] Z. R. M. Kashani *et al.*, "Kavosh: A new algorithm for finding network motifs," *BMC Bioinf.*, vol. 10, no. 1, 2009, Art. no. 318.

[19] N. Kashtan, S. Itzkovitz, R. Milo, and U. Alon, "Efficient sampling algorithm for estimating subgraph concentrations and detecting network motifs," *Bioinf.*, vol. 20, no. 11, pp. 1746–1758, 2004.

[20] S. Khakabimamaghani, I. Sharafuddin, N. Dichter, I. Koch, and A. Masoudi-Nejad, "QuateXelero: An accelerated exact network motif detection algorithm," *PloS One*, vol. 8, no. 7, 2013, Art. no. e68073.

[21] J. Krumsiek, C. Marr, T. Schroeder, and F. J. Theis, "Hierarchical differentiation of myeloid progenitors is encoded in the transcription factor network," *PLoS One*, vol. 6, 2011, Art. no. e22649.

[22] H. Kugler, S.-J. Dunn, and B. Yordanov, "Formal analysis of network motifs," in *Proc. Int. Conf. Comput. Methods Syst. Biol.*, 2018, pp. 111–128.

[23] F. Li, T. Long, Y. Lu, Q. Ouyang, and C. Tang, "The yeast cell-cycle network is robustly designed," *Proc. Yeast Cell-Cycle Netw. Robustly Des.*, vol. 101, pp. 4781–4786, 2004.

[24] X. Li, D. S. Stones, H. Wang, H. Deng, X. Liu, and G. Wang, "NetMODE: Network motif detection without nauty," *PloS One*, vol. 7, no. 12, 2012, Art. no. e50093.

[25] S. Mangan and U. Alon, "Structure and function of the feed-forward loop network motif," *Proc. Nat. Acad. Sci.*, vol. 100, no. 21, pp. 11980–11985, 2003.

[26] S. Mangan, A. Zaslaver, and U. Alon, "The coherent feedforward loop serves as a sign-sensitive delay element in transcription networks," *J. Mol. Biol.*, vol. 334, no. 2, pp. 197–204, 2003.

[27] B. McKay, "Practical graph isomorphism," *Congr, Numerantium*, vol. 30, pp. 45–87, 1981.

[28] L. A. Meira, V. R. Máximo, Á. L. Fazenda, and A. F. Da Conceição, "Acc-Motif: Accelerated network motif detection," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 11, no. 5, pp. 853–862, Sep.-Oct. 2014.

[29] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon, "Network motifs: Simple building blocks of complex networks," *Sci.*, vol. 298, no. 5594, pp. 824–827, 2002.

[30] P. Nurse, "Life, logic and information," *Nature*, vol. 454, no. 7203, pp. 424–426, 2008.

[31] N. Pržulj, "Biological network comparison using graphlet degree distribution," *Bioinf.*, vol. 23, no. 2, pp. e177–e183, 2007.

[32] E. Ravasz, A. L. Somera, D. A. Mongru, Z. N. Oltvai, and A.-L. Barabási, "Hierarchical organization of modularity in metabolic networks," *Sci.*, vol. 297, no. 5586, pp. 1551–1555, 2002.

[33] M. Reigl, U. Alon, and D. B. Chklovskii, "Search for computational modules in the c. elegans brain," *BMC Biol.*, vol. 2, no. 1, 2004, Art. no. 25.

[34] F. Schreiber and H. Schwöbbermeyer, "MAVisto: A tool for the exploration of network motifs," *Bioinf.*, vol. 21, no. 17, pp. 3572–3574, 2005.

[35] S. S. Shen-Orr, R. Milo, S. Mangan, and U. Alon, "Network motifs in the transcriptional regulation network of escherichia coli," *Nature Genetics*, vol. 31, no. 1, 2002, Art. no. 64.

[36] N. Shervashidze, S. Vishwanathan, T. Petri, K. Mehlhorn, and K. Borgwardt, "Efficient graphlet kernels for large graph comparison," in *Proc. 12th Int. Conf. Artif. Intell. Statist.*, 2009, pp. 488–495.

[37] A. Tiwari, C. Talcott, M. Knapp, P. Lincoln, and K. Laderoute, "Analyzing pathways using SAT-based approaches," in *Proc. 2nd Int. Conf. Algebraic Biol.*, 2007, pp. 155–169.

[38] N. T. L. Tran, S. Mohan, Z. Xu, and C.-H. Huang, "Current innovations and future challenges of network motif detection," *Briefings Bioinf.*, vol. 16, no. 3, pp. 497–525, 2015.

[39] S. Wernicke and F. Rasche, "FANMOD: A tool for fast network motif detection," *Bioinf.*, vol. 22, no. 9, pp. 1152–1153, 2006.

[40] E. Wong, B. Baur, S. Quader, and C.-H. Huang, "Biological network motif detection: Principles and practice," *Briefings Bioinf.*, vol. 13, no. 2, pp. 202–215, 2011.

[41] E. Yeger-Lotem *et al.*, "Network motifs in integrated cellular networks of transcription–regulation and protein–protein interaction," *Proc. Nat. Acad. Sci. United States Am.*, vol. 101, no. 16, pp. 5934–5939, 2004.

[42] B. Yordanov, S.-J. Dunn, H. Kugler, A. Smith, G. Martello, and S. Emmott, "A method to identify and analyze biological programs through automated reasoning," *NPJ Syst. Biol. Appl.*, vol. 2, no. 16010, 2016.

Sara-Jane Dunn received the degree in mathematics from the University of Oxford, in 2007. She is a senior scientist with Microsoft Research. She remained in Oxford for her doctoral research as part of the Computational Biology Group at the Department of Computer Science. In 2012, she joined Microsoft Research as a postdoctoral researcher, before transitioning to a permanent scientist role in 2014. In 2016, she was invited to become an affiliate researcher of the Wellcome-MRC Cambridge Stem Cell Institute. Her research interests include uncovering the fundamental principles of biological information-processing, particularly investigating decision-making in development.

Hillel Kugler received the PhD degree from the Weizmann Institute of Science in Israel. He has been a faculty member with the Faculty of Engineering, Bar-Ilan University in Israel since 2015. His research interests include modeling and analyzing complex systems using formal reasoning and synthesis methods. His research interests also include the application of visual languages to model the behavior of reactive systems, the development of new computational methods, and tools towards enabling a deeper understanding of biological computation and biological devices. Before joining the Faculty of Engineering at Bar-Ilan, he was a researcher with Microsoft Research in Cambridge. Previously, he was a postdoc with the Biology Department, New York University, and a member of the the Analysis of Computer Systems Group, Department of Computer Science, Courant Institute, New York University.

Boyan Yordanov received the BA degrees in biochemistry and computer science from Clark University, in 2005 and the PhD degree in biomedical engineering from Boston University, in 2011. He was a postdoctoral researcher with the Mechanical Engineering Department, Boston University, in 2011. He is a senior scientist with Microsoft Research. He joined the Biological Computation Group, Microsoft Research, as a postdoctoral scientist and became a Microsoft research scientist in 2014. His research interests include accelerating the design and construction of biochemical circuits and improving the understanding of biological computation through computational methods for the analysis, verification and synthesis of dynamical systems.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/csdl.