From quantitative SBML models to boolean networks

Athénaïs Vaginay, Taha Boukhobza, Malika Smaïl-Tabbone

From Quantitative SBML Models to Boolean Networks

Athénaïs Vaginay[1,2], Taha Boukhobza[1], and Malika Smaïl-Tabbone[2]

[1] Université de Lorraine, CNRS, CRAN, F-54000 Nancy, France
[2] Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France

Abstract. Modelling complex biological systems is necessary for their study and understanding. SBML is the standard format to represent models of biological systems. Most of the curated models available in the repository Biomodels are quantitative, but in some cases qualitative models—such as Boolean networks—would be better suited. This paper is the first to focus on the automatic transformation of quantitative SBML models to Boolean networks. We propose SBML2BN, a pipeline dedicated to this task. By running SBML2BN on more than 200 quantitative SBML models, we provide evidence that we can automatically construct Boolean networks which are compatible with the structure and the dynamics of a given quantitative SBML model.

Keywords: Boolean Networks · Model Transformation · SBML · Systems Biology

1 Introduction

Life is based on biological systems which are essentially composed of biological components (genes, proteins, metabolites) acted upon by processes. However, they are highly complex, as molecular abundances and interactions change over time in response to external stimuli as well as to dynamical intra-system processes. The biological system that serves as a running example throughout this article is an enzymatic process: first, an enzyme E reversibly binds a molecule of substrate S (reactions $\mathcal{R}_{\text{on}}$ and $\mathcal{R}_{\text{off}}$); together, they form the complex ES; then the substrates are transformed into two molecules of a product P while E returns to its free state (reaction $\mathcal{R}_{\text{cat}}$). The classical chemical notation of this system is:

$$\text{S} + \text{E} \underset{\text{k}_{\text{off}}}{\overset{\text{k}_{\text{on}}}{\rightleftarrows}} \text{ES} \xrightarrow{\text{k}_{\text{cat}}} \text{E} + 2 \cdot \text{P} \tag{1}$$

Each of the three reactions is represented by an arrow from the *reactants* (i.e., components consumed during the reaction) to the *products* (i.e., components created during the reaction). On top (or below) of the arrow is the *speed constant*, which is a proportionality coefficient between the *stoichiometry* (amount) of the reactants and the rate of the reaction.

Through simulations, dynamic models of biological systems are of particular interest because they are useful proxies to understand and predict the behaviour

and the dynamics of biological systems. For example, one can study how the speed of production of the product P is affected by the presence of an inhibitor of the enzyme E. However, the quality of the predictions strongly depends on the quality of the model, which in turn strongly depends on the quality of the data and the depth of the knowledge used to build the model. Moreover, biological models are often hand-crafted and this is error-prone. The repository Biomodels contains a peer-reviewed and curated collection of over a thousand models [20]. Models in Biomodels are described in the Systems Biology Markup Language (SBML), which is the most widely used standard representation language in the field of system biology. Several modelling formalisms exist, ranging from detailed ones (such as differential equations) to the most simple ones (such as Boolean networks). Most SBML models in Biomodels are quantitative models (mostly differential equations). However, in some cases, qualitative models like Boolean networks are more suited [3]. Indeed, their simplicity make them easy to study. They are in particular easily amenable to model checking [16] and control [3], even for large models.

In this paper, we propose SBML2BN, an *automatic* pipeline to synthesise a set of Boolean networks (BNs) modelling a biological system, starting from an SBML representation of the system. First, we introduce the key notions about Boolean networks and the principles of their synthesis starting from the given structure and dynamics of the biological system under study (Sect. 2). Then, we present the pipeline SBML2BN and detail its four steps (Sect. 3): (i) we extract the structure and (ii) the dynamics of the biological system from the SBML model; (iii) we use this information as constraints for the synthesis of the BNs; (iv) we assess the quality of the BNs produced by quantifying how well they fit to the structure and the dynamics of the input SBML model. We also give details about the pipeline implementation, which reuses and extends several published methods and software packages. Finally, we report the evaluation of SBML2BN by running it on more than 200 curated SBML models from the Biomodels database (Sect. 4). We provide evidence that the resulting BNs are in line with the biological system under study. We close the paper with conclusions and a few perspectives.

2 Boolean Networks and Their Synthesis

2.1 Definitions

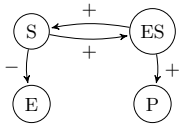
Boolean networks (BNs) were introduced by Kauffman [14] and Thomas [25] to model genetic regulatory networks. Concepts used in BNs are described in a recent review [24]. An example of BN is given in Fig. 1 and used to illustrate the concepts introduced in the following.

The *components* of a BN are the components of the considered biological system. For example, the BN $\mathcal{B}_1$ (Fig. 1) has four components: S, P, E and ES. A *configuration* of a BN is a vector that associates a Boolean value ($\mathbb{B} = \{0/\text{inactive}; 1/\text{active}\}$) to each of the n components of the BN (in alphabetical order). For example, in the configuration 0000, no components is active,

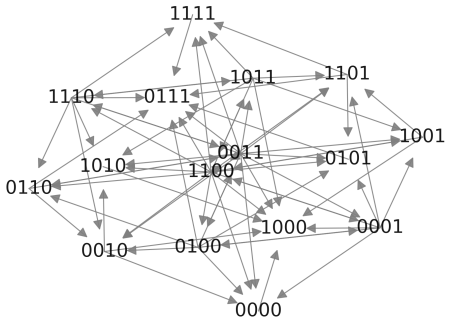
while only E is active in the configuration 1000. A BN with n components has 2^n possible configurations. Each component X has an associated *transition function* $f_X : \mathbb{B}^n \to \mathbb{B}$ that maps the configurations of the BN to the next value of the component. The transition functions are usually written as Boolean expressions. In this paper, these expressions are in Disjunctive Normal Form (DNF), i.e., disjunctions of conjunctions. Moreover, the conjunctions are *satisfiable*, i.e., they do not contain both a literal and its contrary. The operators $\neg$, $\wedge$, $\vee$ represent respectively negation, conjunction and disjunction. The transition function $f_{ES} := (E \wedge \neg S) \vee (\neg E \wedge S)$ states that the value of ES will be 1 if either the value of E or of S was 1 in the previous configuration. Fig. 1a shows examples of transition functions with only one term.

$$\mathcal{B}_1 = \begin{cases} f_E := \neg S \\ f_{ES} := S \\ f_P := ES \\ f_S := ES \end{cases}$$

(a) Transition functions of $\mathcal{B}_1$



(b) Interaction graph of $\mathcal{B}_1$



(c) General Asynchronous State Transition Graph (GA-STG) of $\mathcal{B}_1$

Fig. 1: Example of a possible Boolean network (among others) to model Eq. (1)

The structure of a BN is defined in terms of parent-child relationships between the components. A component P that appears in the transition function of a component X is called a *parent* of X. If the parent P is negated in the DNF associated with X, we say that the *polarity* of the influence of P on X is negative. Conversely, if the parent is not negated, this influence is positive. The *Interaction Graph* (IG) summarises these relationships as a directed graph. The directed edge P $\to$ X is labelled with "+" or "−" depending on the polarity of the influence P has on X. The interaction graph of $\mathcal{B}_1$ (Fig. 1b) contains the edges S $\xrightarrow{-}$ E and S $\xrightarrow{+}$ ES because S appears negatively in the transition function of E and positively in the one of ES. We will see in Sect. 2.2 how the IG is used to define the compatibility of a BN towards a given structure.

The BN *dynamics* is obtained by applying iteratively the transition functions starting from all possible configurations. The order of application of the transition functions is defined by the *update scheme*. The most common are the *synchronous*, *asynchronous* and *general asynchronous*. In the synchronous update scheme, the transition functions are applied all at once, while in the asynchronous update scheme, they are applied one by one (non-deterministically).

In the general asynchronous update scheme, any number of components can be updated at each step. Thus, it includes the updates possibilities of both the synchronous and asynchronous update schemes. The *state transition graph* (STG) is a directed graph whose nodes are the 2^n possible configurations of the BN. It contains a directed edge from c to c' if c' is the result of applying on c the transition function(s) according to the chosen update scheme. Fig. 1c shows the General-Asynchronous STG (GA-STG) of $\mathcal{B}_1$ (Fig. 1a). We will see in Sect. 2.2 how the presence of specific edges in the GA-STG of a BN is used to measure the compatibility of this BN towards a given dynamics.

2.2 Synthesis of BNs Compatible with a Structure and a Dynamics

In general, a Boolean network that models a biological system has to satisfy two categories of *constraints*. On one hand, its structure has to comply with what is known on the system's structure. This knowledge concerns the list of components (genes, proteins...) involved and how they influence each others. Influences have a *polarity*: activation (polarity "+") or inhibition (polarity "−"). The *parents* of a component X are the components which are known to influence X. A *Prior Knowledge Network* (PKN) encodes such knowledge. The nodes of the network are the components of the system, and directed edges parent → child are labelled "+" or "−" according to the polarity of the influences. Fig. 2b shows an example of PKN of the enzymatic reaction Eq. (1). In this PKN, S, ES and E are the parents of E with polarities "−", "+" and "−". The PKN is used to constrain the structure of the synthesised BNs: a BN is *compatible* with a given PKN if its interaction graph is a spanning subgraph of the PKN. In other words, the interaction graph of a BN compatible with a given PKN is formed from the nodes and a subset of the edges of the PKN. This results in constraining which components can appear as variables in each transition function and the polarity of those variables. Hence, a component P is allowed in the transition function of a component X with a polarity σ if the PKN contains an edge $\mathrm{P} \xrightarrow{\sigma} \mathrm{X}$. For example, $\mathcal{B}_1$ (Fig. 1a) is compatible with the PKN given in Fig. 2b. On the contrary, a Boolean network having the transition function $f_{\mathrm{E}} := \neg\mathrm{S} \vee \neg\mathrm{ES}$ is not compatible. Indeed, despite ES being a possible parent of E, the negative polarity is not allowed since $\mathrm{ES} \xrightarrow{-} \mathrm{E}$ is not in the PKN.

On the other hand, the dynamics of the BN has to comply with what is known on the system's dynamics. Starting from a given multivariate Time Series (TS) of the concentrations of the components over time, we can extract a sequence of configurations by binarising the TS. For example, the sequence of configurations extracted from the binarisation of the multivariate TS given in Fig. 2c is 0011 → 1011 → 1010 → 1000 → 1100 → 0101. Ideally, we would like this sequence to be a *walk* in the General Asynchronous STG (GA-STG) i.e., that the GA-STG contains all the edges appearing in the sequence. In such a case, the *coverage ratio* of the GA-STG towards the configuration sequence (defined as the number of edges present in the graph divided by the number of distinct edges in the sequence) is of 1 and the Boolean network is said to be *fully compatible* with the multivariate TS. However, it is not always possible to retrieve the complete

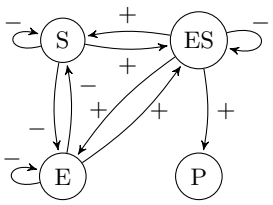
walk in the GA-STG [23]. In this case, the goal is to have the best coverage ratio possible.

All in one, a Boolean network is *compatible* with a Prior Knowledge Network (PKN) if its interaction graph is a subgraph of the PKN, and the compatibility between a Boolean network and a multivariate Time-Series (TS) is quantified using the *coverage ratio*. An ideal Boolean network synthesis method constructs *only* Boolean networks compatible with the given PKN and with *maximal* coverage ratio (of 1) in regard of the given multivariate TS.

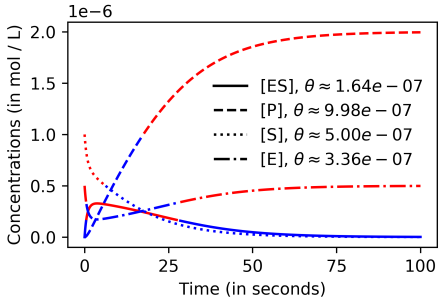
$$
\begin{aligned}
\frac{d[E]}{dt} &= -k_{on}[E][S] + k_{off}[ES] + k_{cat}[ES] \\
\frac{d[ES]}{dt} &= k_{on}[E][S] - k_{off}[ES] - k_{cat}[ES] \\
\frac{d[P]}{dt} &= 2\,k_{cat}[ES] \\
\frac{d[S]}{dt} &= -k_{on}[E][S] + k_{off}[ES]
\end{aligned}
$$

$$
\begin{aligned}
k_{on} &= 10^{6}\,\mathrm{L\,mol^{-1}\,s^{-1}} \\
k_{off} &= 0.2\,\mathrm{s^{-1}} \\
k_{cat} &= 0.1\,\mathrm{s^{-1}} \\
[ES]_0 &= 0\,\mathrm{mol\,L^{-1}} \\
[P]_0 &= 0\,\mathrm{mol\,L^{-1}} \\
[S]_0 &= 1 \times 10^{-6}\,\mathrm{mol\,L^{-1}} \\
[E]_0 &= 5 \times 10^{-7}\,\mathrm{mol\,L^{-1}}
\end{aligned}
$$

(a)



(b)



(c)

Fig. 2: (a) ODE system and its parametrisation, and (b) prior knowledge network for Eq. (1). (c) shows the multivariate time series, binarisation thresholds and resulting binarisation (blue if 0 and red if 1) obtained by simulation of the ODE system.

3 Description of the SBML2BN Pipeline

We propose SBML2BN, a pipeline for the automatic synthesis of Boolean networks starting from an existing quantitative SBML description of a biological system. All the necessary concepts about SBML are described in Sect. 3.1. The structure (PKN) and the dynamics (TS) of the biological system under study

are extracted from the given SBML (Sects. 3.2 and 3.3). In the BN synthesis step (Sect. 3.4), the former is used to hard constrain the structure of the resulting BNs, while the latter acts as soft constraints. The pipeline finishes with the evaluation of the *set* of achieved BNs (Sect. 3.5).

3.1 Complete Quantitative SBML Models in a Nutshell

The Systems Biology Markup Language (SBML) [15] is an XML markup language. The SBML file representing the biological system from Eq. (1) is given in the GitLab repository associated to this paper†. The SBML standard[3] specifies how an SBML file is structured and how the different elements are named. This paper focuses on a subset of SBML models which contains all the necessary information for the SBML2BN pipeline to process the model. We refer to these SBML models as *complete quantitative SBML models*. We describe the content of such models as follows.

The biological components involved in the biological system (referred to as *species* in SBML) are supplied, as well as their initial concentration. The reactions taking place in the biological system are described. The definition of a reaction $\mathcal{R}$ is composed of a list of *reactants*, a list of *products*, and a *kinetic law* $e_{\mathcal{R}}$ (i.e., a mathematical expression which gives the rate of the reaction $\mathcal{R}$). For each species X involved in a reaction $\mathcal{R}$, the *net stoichiometry* $\nu_{\mathcal{R}}^{\mathrm{X}}$ of X in $\mathcal{R}$ is the amount of X as a product minus its amount as a reactant. If $\nu_{\mathcal{R}}^{\mathrm{X}} > 0$ (resp. < 0), X is *effectively* produced (resp. consumed) by the reaction $\mathcal{R}$. Sometimes, a reaction involves *modifiers*, i.e., species which influence the speed of the reaction without having their amount modified. A modifier which increases (resp. decreases) the speed of the reaction is an *activator* (resp. *inhibitor*). Finally, all the supplementary kinetic parameters and their values are specified.

3.2 Extraction of the PKN from the SBML Model

This first step consists in the construction of the PKN (noted $\mathcal{G}$). Fig. 2b is the PKN constructed by SBML2BN for Eq. (1). The PKN we obtain in this step corresponds to the Syntactical Influence Graph (SIG) of an SBML model [11]. The nodes of the PKN are the SBML species of the SBML model. As for the edges, they are obtained by applying the following rules on each reaction of the SBML model:

- **If** X is a reactant or an activator and Y disappears **then** $\mathrm{X} \xrightarrow{-} \mathrm{Y} \in \mathcal{G}$
- **If** X is an inhibitor and Y appears **then** $\mathrm{X} \xrightarrow{-} \mathrm{Y} \in \mathcal{G}$
- **If** X is a reactant or an activator and Y appears **then** $\mathrm{X} \xrightarrow{+} \mathrm{Y} \in \mathcal{G}$
- **If** X is an inhibitor and Y disappears **then** $\mathrm{X} \xrightarrow{+} \mathrm{Y} \in \mathcal{G}$

As detailed in Sect. 3.1, a modifier can be either an activator or an inhibitor (or both). In some SBML models, specific annotations (using the System Biology Ontology [7]) indicate the exact role of the modifiers. When such annotations are missing, the modifier is considered as both activator and inhibitor.

[3] http://sbml.org/Documents/Specifications

3.3 Extraction of the Time-Series from the SBML Model

The goal of this step is to retrieve the concentrations of the species over time. Since the processed SBML model is *complete*, it contains all the necessary information to construct a working Ordinary Differential Equations system (ODE). To do so, an expression representing the overall rate of change of the amount of each species is constructed as the sum of the contributions of all the *relevant* reactions (i.e., reaction in which a given species is involved as a product or a reactant). For example, in the running example, the species ES is involved as a product in reaction $\mathcal{R}_{\text{on}}$ and as a reactant in reactions $\mathcal{R}_{\text{cat}}$ and $\mathcal{R}_{\text{off}}$. Hence, the overall rate of change of ES is: $\frac{\text{dES}}{\text{d}t} = \nu^{\text{ES}}_{\mathcal{R}_{\text{on}}} \cdot e_{\mathcal{R}_{\text{on}}} + \nu^{\text{ES}}_{\mathcal{R}_{\text{off}}} \cdot e_{\mathcal{R}_{\text{off}}} + \nu^{\text{ES}}_{\mathcal{R}_{\text{cat}}} \cdot e_{\mathcal{R}_{\text{cat}}}$ with $\nu^{\text{ES}}_{\mathcal{R}_{\text{on}}} = 1$, and both $\nu^{\text{ES}}_{\mathcal{R}_{\text{off}}}$ and $\nu^{\text{ES}}_{\mathcal{R}_{\text{cat}}} = -1$. The SBML representation[†] of Eq. (1) indicates that the speed of each reaction is proportional (with a factor $k_{\mathcal{R}}$) to the product of the amount of reactants. Fig. 2a shows the ODE system, parameterisation and initial conditions retrieved from the SBML file[†].

We then run a deterministic numerical time integration from $t = 0$ to $t_{\max}$ of the ODE system. Fig. 2c shows the multivariate TS obtained by simulating Fig. 2a for $t_{\max} = 100$ seconds (chosen arbitrarily).

3.4 Boolean Networks Synthesis

At this stage, the goal is to construct automatically a set of BNs compatible with the PKN and the multivariate TS. The synthesis problem is largely underspecified, since only one multivariate TS is provided. Several methods have been dedicated to this task [18, 19, 22]. They exploit various strategies, in particular regarding the fitting of the transition functions to given multivariate TS.

In [26], we introduced ASKeD-BN and showed that it is the best synthesis method available in the case of signed PKN and complete multivariate TS (i.e., without missing time steps). ASKeD-BN exhaustively synthesises BNs compatible with a given PKN and multivariate TS with respect to two criteria that correspond closely to the notion of *compatibility* defined in Sect. 2.2:

1. The interaction graph of the synthesised BNs are compatible with the given PKN (i.e., be a subgraph of the PKN), and they have the smallest number of edges possible.
2. The dynamics of each component minimises the mean absolute error with regard to the multivariate TS.

The choice of the binarisation might be crucial for the outcome. ASKeD-BN uses the simplest procedure possible: a threshold θ_{X} is chosen for each component X as $\min + (\max - \min)/2$, where min and max are the observed minimum and maximum of X in the time series. With x_t the value of the concentration of the species X at time t, the binarised value of X at time t is 1 if $x_t \geqslant \theta_{\text{X}}$ and 0 otherwise. In the multivariate TS (Fig. 2c), the red (resp. blue) parts of the lines correspond to concentration values which are bigger (resp. small) than the corresponding threshold, hence will result in 1 (resp. 0). After binarisation, we can extract the following configuration sequence: $0011 \to 1011 \to 1010 \to 1000$

$\rightarrow 1100 \rightarrow 0101$. ASKeD-BN fits the transitions functions by using the mean absolute error to penalise the candidates for each transition they cannot explain.

Other methods such as caspo-TS [22] work on explaining the reachability of the configurations. Hence, wildcard are added to the configurations sequence: $0011 \rightarrow * \rightarrow 1011 \rightarrow * \rightarrow 1010 \rightarrow \ldots$ This feature is an asset in the case of missing time points, but here, the multivariate TS are complete, and this feature is not necessary (and even counter-productive [26]).

3.5 Evaluation of Synthesised Boolean Networks

In this last step, we evaluate the compatibility (such as defined in Sect. 2.2) of all the Boolean networks synthesised by the SBML2BN for the input SBML model. Since they are compatible with the PKN (by construction), our quality check focuses on the compatibility with the multivariate TS: we compute the *coverage ratio* of each BN, and we aggregate the individual coverage ratios using the median and standard deviation. Ideally, the pipeline would return *only* BNs with *maximal* coverage ratios.

3.6 Pipeline Implementation

We have made a point of supporting reproducibility and facilitating the installation of the different tools. All the tools developed and reused are open-source, well documented and freely available. The pipeline is managed using Snakemake [21] (which ensures each step is ran properly and in the correct order) and installed using Conda [1] (which simplifies the management of library dependencies and avoid version conflicts). We use the parser libSBML [4] to retrieve the PKN, COPASI [13] to retrieve the multivariate TS, and PyBoolNet [17] to compute the AG-STG of the BNs. For the BN synthesis step, we used a declarative implementation of ASKeD-BN [26], which uses Answer-Set Programming [12].

4 Evaluation of the SBML2BN Pipeline

4.1 Evaluation on the running example Eq. (1)

We apply SBML2BN on the SBML file[†] modelling Eq. (1). The Boolean network $\mathcal{B}_1$ (Fig. 1a) is the only solution we obtain. Its interaction graph (Fig. 1b) is a spanning subgraph of the PKN. and its GA-STG (Fig. 1c) covers all the 5 transitions extracted from the binarised TS. Its coverage ratio is thus 1, and the coverage median and standard deviation of this singleton of solutions are obviously 1 and 0 respectively, making SBML2BN plainly successful.

4.2 Evaluation on SBML Models from BioModels

BioModels [20] is a repository of models of biological and biomedical systems, including metabolic networks, signalling networks, gene regulatory networks and

infectious diseases. All models stored in the `curated` branch of BioModels are encoded in SBML and have passed a drastic manual curation process, which asserts that the simulations from the paper in which the model was originally published are reproducible by the SBML model. The latest available release of Biomodels[4] contains 640 SBML curated models, including 369 complete quantitative SBML (i.e., models for which SBML2BN is able to extract a PKN and a multivariate TS). However, the complexity of the BN synthesis problem increases exponentially with the number of parents for each component. Indeed, the number of possible transition functions for a component with p parents is 2^{2^p}. We assume the problem is not tractable if a component has more than 10 parents, hence we only make the evaluation onto the 209 SBML models that have all their components with less than 10 parents. The number of components in these models ranges from 2 to 60, but bigger models would not have been a problem since ASKeD-BN is not directly impacted by the number of components. For each SBML model, the length of simulation ($t_{\max}$) is extracted by hand from the curation reports of Biomodels.

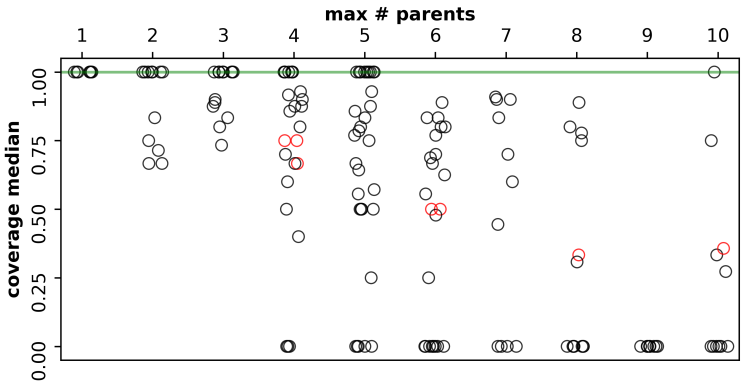


Fig. 3: Coverage evaluation for the BNs synthesised by SBML2BN for 155 SBML models. Each dot represents the set of BNs returned for a given SBML model. Its coordinates are the coverage ratio median (ordinate) and the number maximal of parents for the nodes of the SBML model (abscissa). Sets of BNs having a coverage variance strictly positive are represented by red dots. Green line shows where are the dots when the pipeline only returns BNs with a perfect coverage.

The pipeline processes about three fourth (155) of the models in less than 30 hours (median of 36 minutes—data not shown). In the following, we report the results for these models. In average, the pipeline synthesises 6.5 Boolean networks per SBML. This number masks a strong disparity, since a single BN was synthesised for more than half (106) SBML models. Fig. 3 summarises the coverage evaluation. As said before, all the BNs returned by the pipeline for a

[4] release 31 ftp://ftp.ebi.ac.uk/pub/databases/biomodels/releases/2017-06-26/

given SBML model would ideally have a perfect coverage ratio, hence with a median of 1 and a standard deviation of 0. The pipeline synthesises only perfect BNs for one fourth (39) SBML models. The median and standard deviation of the median coverage ratios of the BNs synthesised for a given SBML model are 0.77 and 0 respectively. They are only 12 models (in red in Fig. 3) for which the standard deviation is not 0 (range 0.07; 0.25). Overall, the pipeline is efficient at finding Boolean networks with good coverage median and small standard deviation. Nevertheless, we can significantly correlate a loss of performance to the maximum number of parents in the systems (Kendall τ value of -0.43, p-value of 1.51e$-$13). We are currently investigating reasons of this correlation. One reason could simply be that Boolean networks cannot explain all phenomena (Sect. 2.2): in some cases, the maximum achievable coverage ratio is smaller than 1, but our quality evaluation of the synthesised BNs does not take this fact into account. We could use Boolean networks with the most-permissive semantics [5] to overcome this limitation, but no implementation is available for BNs having non-monotonous transition functions (such as the ones our pipeline is likely to produce). Another reason could be that the specification of SBML leaves open the possibility for a model to contain contradictory information. It has been showed in [9] that more than 60% of the SBML models tested in 2012 were containing contradictions. Among the contradictory models they identified, the model n°44[5] has reactions with components used in the kinetics which are not listed as reactants nor modifiers. This has a bad impact on the construction of the PKN by our pipeline (Sect. 3.2), since potential parents of some components are not identified as such. For this model, one BN was generated, with a not so good coverage of 0.55. We are planning to investigate how to remove beforehand the contradictions from these models [9].

5 Conclusion and Perspectives

In this paper, we presented SBML2BN, a pipeline for the automatic transformation of a *complete* quantitative SBML model into a set of compatible Boolean networks. The transformation of biological models from a formalism to another has been investigated in several papers [2, 10] in particular from ODE system to Boolean networks [8]. Yet, our study is the first to be dedicated to the *automatic* transformation from a complete quantitative SBML model to Boolean networks. As a complete and automatic process, our pipeline reduces the risk of errors and saves effort and time of biologists. Our results show that SBML2BN succeeds most of the time at recovering small sets of BNs compatible with both the structure and dynamics extracted from the input SBML model.

Overall, SBML2BN is an important building block on which we can build upon. We are investigating strategies to make the pipeline even more efficient, and on more complex models (i.e., for models with more than 10 parents for a component). To go beyond, we plan to take benefit of the *set* of BNs synthesised for a given SBML model by combining and simulating them together, as recently

[5] https://www.ebi.ac.uk/biomodels/BIOMD0000000044

proposed in [6]. We are also investigating how to validate and then aggregate BNs from several SBML models when they concern the same biological system.

† *Availability* All data and programs needed to reproduce the presented results are accessible at https://gitlab.inria.fr/avaginay/CNA2021.

Acknowledgements We thank Hans-Jörg Schurr for his valuable comments and suggestions, and Laurine Hubert for helpful comments on an early draft.

References

1. Conda. Anaconda Software Distribution (Sep 2021)
2. Aghamiri, S.S., Singh, V., Naldi, A., Helikar, T., Soliman, S., Niarakis, A.: Automated inference of Boolean models from molecular interaction maps using CaSQ. Bioinformatics 36(16), 4473–4482 (Aug 2020)
3. Biane, C., Delaplace, F., Melliti, T.: Abductive network action inference for targeted therapy discovery. Electronic Notes in Theoretical Computer Science 335, 3–25 (Apr 2018)
4. Bornstein, B.J., Keating, S.M., Jouraku, A., Hucka, M.: LibSBML: An API Library for SBML. Bioinformatics 24(6), 880–881 (Mar 2008)
5. Chatain, T., Haar, S., Kolčák, J., Paulevé, L.: Most Permissive Semantics of Boolean Networks. Research Report, Univ. Bordeaux, Bordeaux INP, CNRS, LaBRI, UMR5800, F-33400 Talence, France ; LSV, ENS Cachan, CNRS, INRIA, Université Paris-Saclay, Cachan (France) (2020)
6. Chevalier, S., Noël, V., Calzone, L., Zinovyev, A., Paulevé, L.: Synthesis and Simulation of Ensembles of Boolean Networks for Cell Fate Decision. In: Abate, A., Petrov, T., Wolf, V. (eds.) Computational Methods in Systems Biology. pp. 193–209. Lecture Notes in Computer Science, Springer International Publishing, Cham (2020)
7. Courtot, M., Juty, N., Knüpfer, C., Waltemath, D., Zhukova, A., Dräger, A., Dumontier, M., Finney, A., Golebiewski, M., Hastings, J., Hoops, S., Keating, S., Kell, D.B., Kerrien, S., Lawson, J., Lister, A., Lu, J., Machne, R., Mendes, P., Pocock, M., Rodriguez, N., Villeger, A., Wilkinson, D.J., Wimalaratne, S., Laibe, C., Hucka, M., Novère, N.L.: Controlled vocabularies and semantics in systems biology. Molecular Systems Biology 7(1), 543 (Jan 2011)
8. Davidich, M., Bornholdt, S.: The transition from differential equations to boolean networks: A case study in simplifying a regulatory network model. Journal of Theoretical Biology 255(3), 269–277 (Dec 2008)
9. Fages, F., Gay, S., Soliman, S.: Automatic Curation of SBML Models based on their ODE Semantics. Research Report RR-8014, INRIA (Jul 2012)
10. Fages, F., Soliman, S.: Abstract interpretation and types for systems biology. Theoretical Computer Science 403(1), 52–70 (Aug 2008)
11. Fages, F., Soliman, S.: From Reaction Models to Influence Graphs and Back: A Theorem. In: Fisher, J. (ed.) Formal Methods in Systems Biology. pp. 90–102. Lecture Notes in Computer Science, Springer Berlin Heidelberg (2008)
12. Gebser, M., Kaminski, R., Kaufmann, B., Schaub, T.: Answer Set Solving in Practice. Morgan & Claypool Publishers (2012)
13. Hoops, S., Sahle, S., Gauges, R., Lee, C., Pahle, J., Simus, N., Singhal, M., Xu, L., Mendes, P., Kummer, U.: COPASI—a COmplex PAthway SImulator. Bioinformatics 22(24), 3067–3074 (Dec 2006)

14. Kauffman, S.A.: Metabolic stability and epigenesis in randomly constructed genetic nets. Journal of Theoretical Biology 22(3), 437–467 (Mar 1969)
15. Keating, S.M., Waltemath, D., König, M., Zhang, F., Dräger, A., Chaouiya, C., Bergmann, F.T., Finney, A., Gillespie, C.S., Helikar, T., Hoops, S., Malik-Sheriff, R.S., Moodie, S.L., Moraru, I.I., Myers, C.J., Naldi, A., Olivier, B.G., Sahle, S., Schaff, J.C., Smith, L.P., Swat, M.J., Thieffry, D., Watanabe, L., Wilkinson, D.J., Blinov, M.L., Begley, K., Faeder, J.R., Gómez, H.F., Hamm, T.M., Inagaki, Y., Liebermeister, W., Lister, A.L., Lucio, D., Mjolsness, E., Proctor, C.J., Raman, K., Rodriguez, N., Shaffer, C.A., Shapiro, B.E., Stelling, J., Swainston, N., Tanimura, N., Wagner, J., Meier-Schellersheim, M., Sauro, H.M., Palsson, B., Bolouri, H., Kitano, H., Funahashi, A., Hermjakob, H., Doyle, J.C., Hucka, M., SBML Level 3 Community members: SBML Level 3: An extensible format for the exchange and reuse of biological models. Molecular Systems Biology 16(8), e9110 (Aug 2020)
16. Klarner, H., Heinitz, F., Nee, S., Siebert, H.: Basins of Attraction, Commitment Sets, and Phenotypes of Boolean Networks. IEEE/ACM Transactions on Computational Biology and Bioinformatics 17(4), 1115–1124 (Jul 2020)
17. Klarner, H., Streck, A., Siebert, H.: PyBoolNet: A python package for the generation, analysis and visualization of boolean networks. Bioinformatics p. btw682 (Oct 2016)
18. Lähdesmäki, H., Shmulevich, I., Yli-Harja, O.: On learning gene regulatory networks under the boolean network model. Machine Learning 52(1), 147–167 (Jul 2003)
19. Liang, S., Fuhrman, S., Somogyi, R.: REVEAL, a general reverse engineering algorithm for inference of genetic network architectures. Pacific Symposium on Biocomputing. pp. 18–29 (1998)
20. Malik-Sheriff, R.S., Glont, M., Nguyen, T.V.N., Tiwari, K., Roberts, M.G., Xavier, A., Vu, M.T., Men, J., Maire, M., Kananathan, S., Fairbanks, E.L., Meyer, J.P., Arankalle, C., Varusai, T.M., Knight-Schrijver, V., Li, L., Dueñas-Roca, C., Dass, G., Keating, S.M., Park, Y.M., Buso, N., Rodriguez, N., Hucka, M., Hermjakob, H.: BioModels—15 years of sharing computational models in life science. Nucleic Acids Research 48(D1), D407–D415 (Jan 2020)
21. Mölder, F., Jablonski, K.P., Letcher, B., Hall, M.B., Tomkins-Tinch, C.H., Sochat, V., Forster, J., Lee, S., Twardziok, S.O., Kanitz, A., Wilm, A., Holtgrewe, M., Rahmann, S., Nahnsen, S., Köster, J.: Sustainable data analysis with Snakemake. F1000Research 10, 33 (Apr 2021)
22. Ostrowski, M., Paulevé, L., Schaub, T., Siegel, A., Guziolowski, C.: Boolean network identification from perturbation time series data combining dynamics abstraction and logic programming. Biosystems 149, 139–153 (Nov 2016)
23. Paulevé, L., Kolčák, J., Chatain, T., Haar, S.: Reconciling qualitative, abstract, and scalable modeling of biological networks. Nature Communications 11(1), 4256 (Aug 2020)
24. Schwab, J.D., Kühlwein, S.D., Ikonomi, N., Kühl, M., Kestler, H.A.: Concepts in Boolean network modeling: What do they all mean? Computational and Structural Biotechnology Journal 18, 571–582 (Jan 2020)
25. Thomas, R.: Boolean formalization of genetic control circuits. Journal of Theoretical Biology 42(3), 563–585 (Dec 1973)
26. Vaginay, A., Boukhobza, T., Smaïl-Tabbone, M.: Automatic Synthesis of Boolean Networks from Biological Knowledge and Data. In: Dorronsoro, B., Amodeo, L., Pavone, M., Ruiz, P. (eds.) Optimization and Learning. pp. 156–170. Communications in Computer and Information Science, Springer International Publishing, Cham (2021)