Constraint-based abstraction of reaction networks to Boolean networks

Athénaïs Vaginay

@Caen, 5th December 2023

Systems Biology

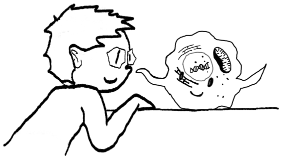
Formal modelling and reasoning about biological systems

A set of species of interest genes, proteins, cells, animals...

Questions

How does the system evolve?

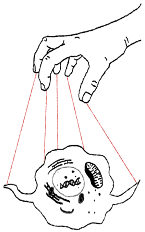
Is the population of some cell type stable over time?



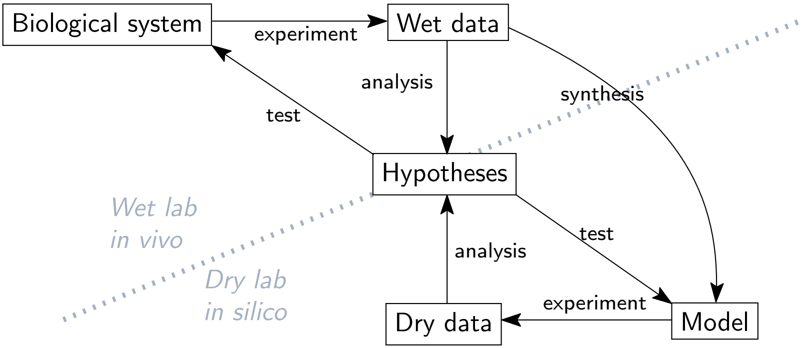
How to control the system?

Cure a pathological system

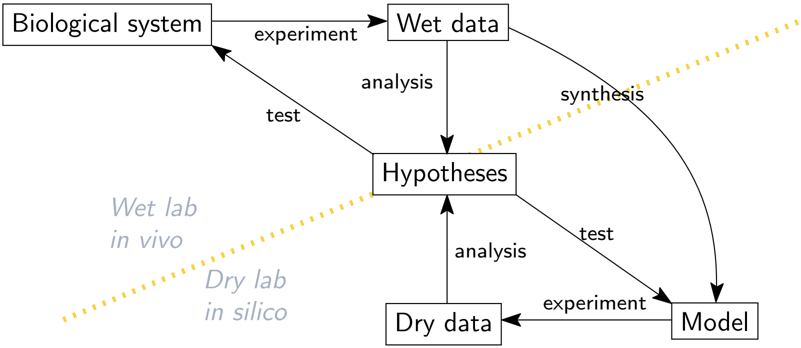
Produce more of some species of interest



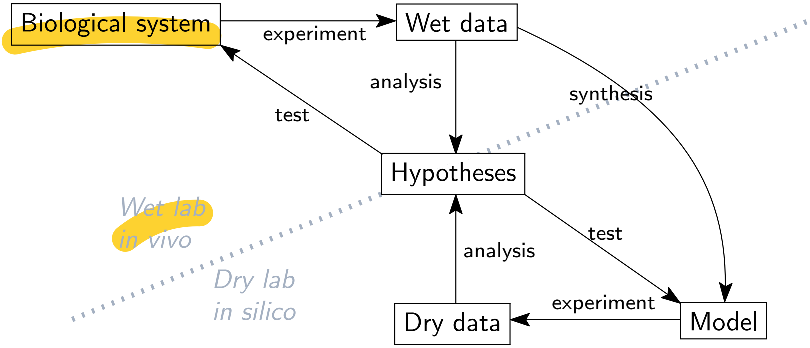
The workflow of system biology [Kohl et al., 2010]



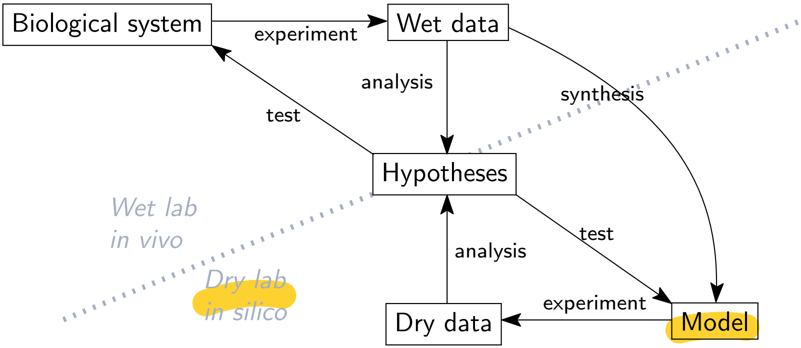
The workflow of system biology [Kohl et al., 2010]



The workflow of system biology [Kohl et al., 2010]



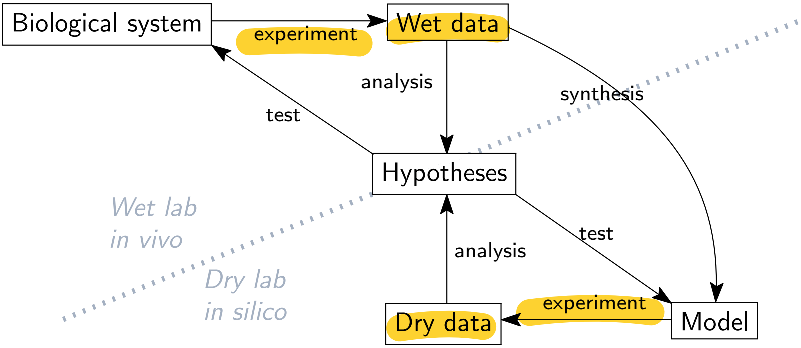
The workflow of system biology [Kohl et al., 2010]



Definition (Model)

Abstract representation (abbreviated and convenient) of the reality (more complex and detailed).

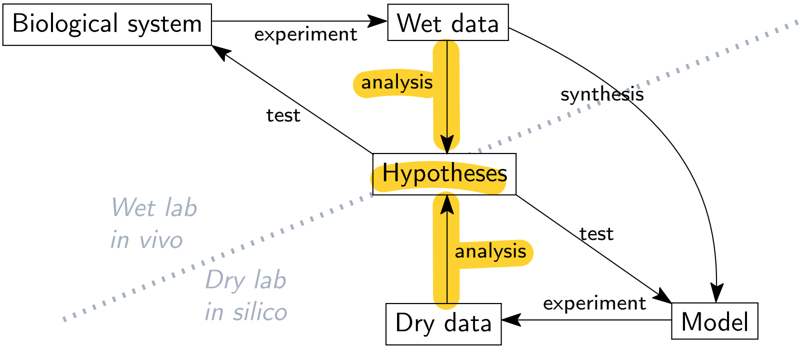
The workflow of system biology [Kohl et al., 2010]



Definition (Model)

Abstract representation (abbreviated and convenient)
of the reality (more complex and detailed).

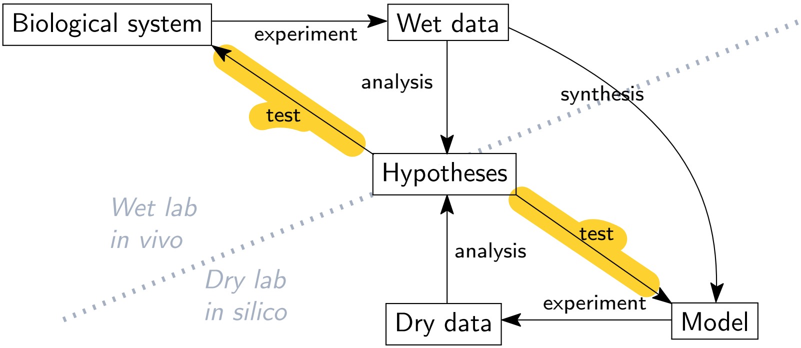
The workflow of system biology [Kohl et al., 2010]



Definition (Model)

Abstract representation (abbreviated and convenient) of the reality (more complex and detailed).

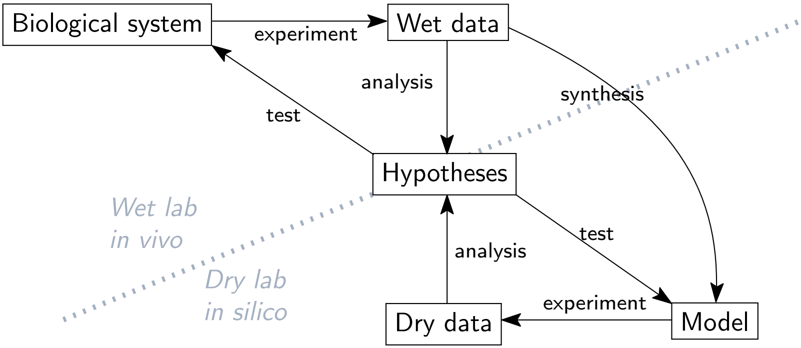
The workflow of system biology [Kohl et al., 2010]



Definition (Model)

Abstract representation (abbreviated and convenient) of the reality (more complex and detailed).

The workflow of system biology [Kohl et al., 2010]

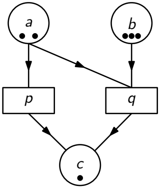


Definition (Model)

Abstract representation (abbreviated and convenient) of the reality (more complex and detailed).

A dichotomic zoo of modelling approaches

Petri net



Process algebra

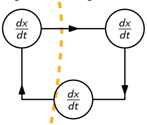
$((b(x, de)[E]) \parallel (B(y, dI)[I]))$

$bh(x, dE)bh(y, dI)(E \parallel I)$

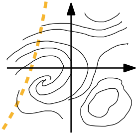
Reaction network

$A + B \to C$

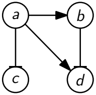
Hybrid system



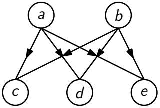
Differential equations



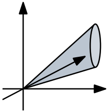
Boolean network



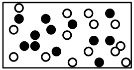
Bayesian network



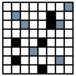
Constraint based model



Agent-based model



Cellular automata



Principles shared across modelling approaches

Synthesis	Usage
▶ from available knowledge and data about the structure and the dynamics ▶ parameter fitting task find models that optimise some criteria	▶ encodes our knowledge, cannot be exact ▶ various analyses simulation, control

Principles shared across modelling approaches

Synthesis

- from available knowledge and data
  about the structure and the dynamics
- parameter fitting task
  find models that optimise some criteria

Usage

- encodes our knowledge, cannot be exact
- various analyses
  simulation, control

Use the simplest model that contains enough information to answer the question at hand [Bornholdt, 2005].

Principles shared across modelling approaches

Synthesis

- from available knowledge and data
  about the structure and the dynamics
- parameter fitting task
  find models that optimise some criteria

Usage

- encodes our knowledge, cannot be exact
- various analyses
  simulation, control

Use the simplest model that contains enough information to answer the question at hand [Bornholdt, 2005].
Boolean networks are simpler than reaction networks.

Principles shared across modelling approaches

Synthesis

- from available knowledge and data
  about the structure and the dynamics
- parameter fitting task
  find models that optimise some criteria

Usage

- encodes our knowledge, cannot be exact
- various analyses
  simulation, control

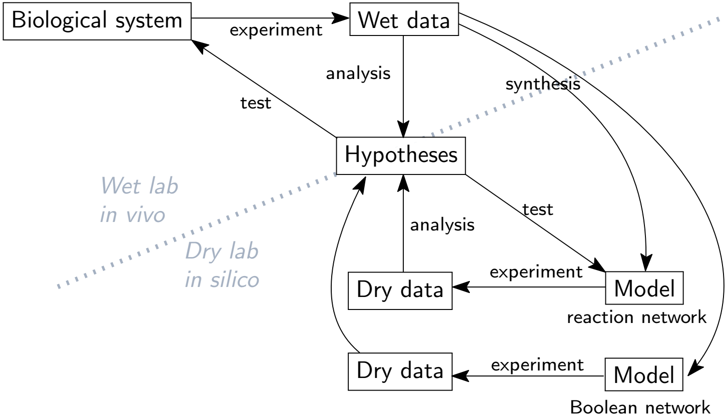
Use the simplest model that contains enough information to answer the question at hand [Bornholdt, 2005].
Boolean networks are simpler than reaction networks.

Problem statement

Automatic transformation (abstraction) of reaction networks to Boolean networks

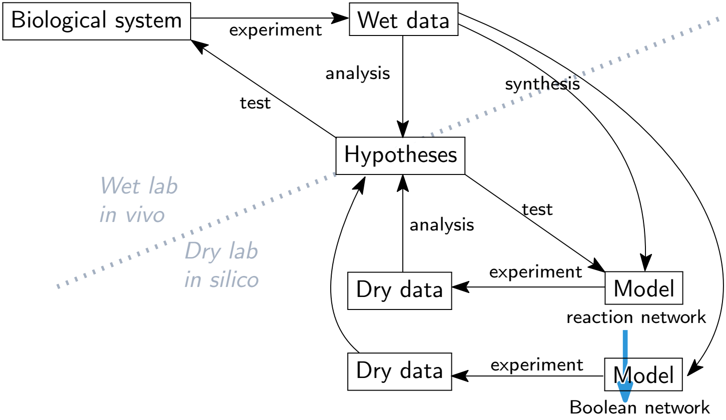
From reactions to Boolean influences with guarantees

Why?



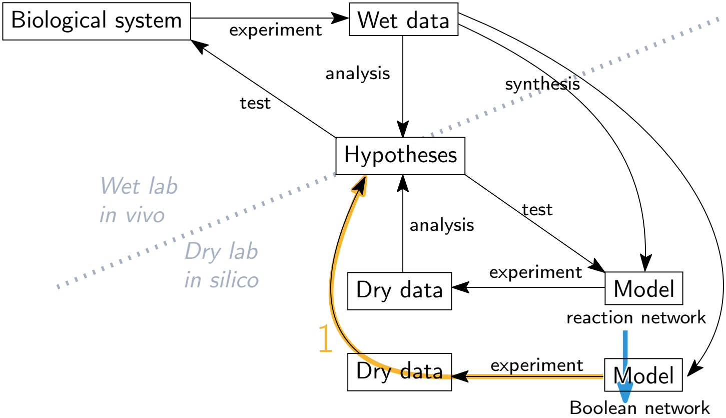
From reactions to Boolean influences with guarantees

Why?



From reactions to Boolean influences with guarantees

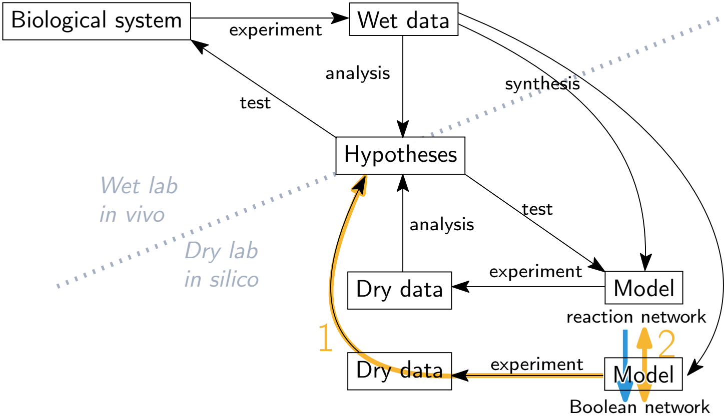
Why?



1. Use BNs to facilitate some analyses

From reactions to Boolean influences with guarantees

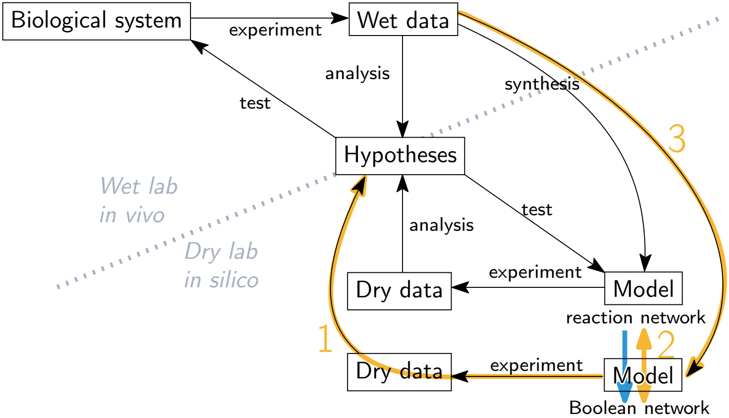
Why?



1. Use BNs to facilitate some analyses
2. Explore the formal relationship between RN and BN

From reactions to Boolean influences with guarantees

Why?



1. Use BNs to facilitate some analyses
2. Explore the formal relationship between RN and BN
3. Improve the BN synthesis methods

Outline

1. Preliminaries on reaction networks and Boolean networks
2. My method and its guarantees (where constraints pop in)
3. Evaluation of the approach
4. Conclusion and perspectives

Outline

1. Preliminaries on reaction networks and Boolean networks
2. My method and its guarantees (where constraints pop in)
3. Evaluation of the approach
4. Conclusion and perspectives

Preliminaries

Reaction networks, structure and dynamics

$\mathcal{R} = \{\mathcal{R}_i : R_i \xrightarrow{e_i} P_i\}_{i=1\ldots m}$

reaction, reactants, products, kinetics

Example

$$\mathcal{S} = \{\mathsf{A}, \mathsf{B}, \mathsf{C}\}$$

$$\mathcal{R}_1 : \mathsf{A} + \mathsf{B} \xrightarrow{e_1} 2 \times \mathsf{C}$$

$$\mathcal{R}_2 : \mathsf{A} + \mathsf{C} \xrightarrow{e_2} \mathsf{A} + \mathsf{B}$$

Reaction networks, structure and dynamics

$\mathcal{R} = \{\mathcal{R}_i : R_i \xrightarrow{e_i} P_i\}_{i=1...m}$
reaction, reactants, products, kinetics

Example

$$\mathcal{S} = \{\mathsf{A}, \mathsf{B}, \mathsf{C}\}$$

$$\mathcal{R}_1 : \mathsf{A} + \mathsf{B} \xrightarrow{e_1} 2 \times \mathsf{C}$$

$$\mathcal{R}_2 : \mathsf{A} + \mathsf{C} \xrightarrow{e_2} \mathsf{A} + \mathsf{B}$$

Reaction networks, structure and dynamics

$\mathcal{R} = \{\mathcal{R}_i : R_i \xrightarrow{e_i} P_i\}_{i=1\ldots m}$

reaction, reactants, products, kinetics

Example

$$\mathcal{S} = \{\mathsf{A}, \mathsf{B}, \mathsf{C}\}$$

$$\mathcal{R}_1 : \mathsf{A} + \mathsf{B} \xrightarrow{e_1} 2 \times \mathsf{C}$$

$$\mathcal{R}_2 : \mathsf{A} + \mathsf{C} \xrightarrow{e_2} \mathsf{A} + \mathsf{B}$$

Reaction networks, structure and dynamics

$$\mathcal{R} = \{\mathcal{R}_i : R_i \xrightarrow{e_i} P_i\}_{i=1\ldots m}$$

reaction, reactants, products, kinetics

Example

$$\mathcal{S} = \{\mathsf{A}, \mathsf{B}, \mathsf{C}\}$$

$$\mathcal{R}_1 : \mathsf{A} + \mathsf{B} \xrightarrow{e_1} 2 \times \mathsf{C}$$

$$\mathcal{R}_2 : \mathsf{A} + \mathsf{C} \xrightarrow{e_2} \mathsf{A} + \mathsf{B}$$

Reaction networks, structure and dynamics

$\mathcal{R} = \{\mathcal{R}_i : R_i \xrightarrow{e_i} P_i\}_{i=1\ldots m}$
reaction, reactants, products, kinetics

Example

$$\mathcal{S} = \{\mathsf{A}, \mathsf{B}, \mathsf{C}\}$$

$$\mathcal{R}_1 : \mathsf{A} + \mathsf{B} \xrightarrow{e_1} 2 \times \mathsf{C}$$

$$\mathcal{R}_2 : \mathsf{A} + \mathsf{C} \xrightarrow{e_2} \mathsf{A} + \mathsf{B}$$

Reaction networks, structure and dynamics

$\mathcal{R} = \{\mathcal{R}_i : R_i \xrightarrow{e_i} P_i\}_{i=1\dots m}$
reaction, reactants, products, kinetics

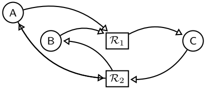
Reaction graph
$(\mathcal{S} \cup \mathcal{R}, E \subseteq (\mathcal{S} \times \mathcal{R}) \cup (\mathcal{R} \times \mathcal{S}))$

Example

$\mathcal{S} = \{\mathsf{A}, \mathsf{B}, \mathsf{C}\}$

$\mathcal{R}_1 : \mathsf{A} + \mathsf{B} \xrightarrow{e_1} 2 \times \mathsf{C}$

$\mathcal{R}_2 : \mathsf{A} + \mathsf{C} \xrightarrow{e_2} \mathsf{A} + \mathsf{B}$



Reaction networks, structure and dynamics

$\mathcal{R} = \{\mathcal{R}_i : R_i \xrightarrow{e_i} P_i\}_{i=1\dots m}$
reaction, reactants, products, kinetics

Reaction graph

$(\mathcal{S} \cup \mathcal{R}, E \subseteq (\mathcal{S} \times \mathcal{R}) \cup (\mathcal{R} \times \mathcal{S}))$

Differential semantics

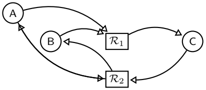
ordinary differential equation (ODE)
$\left\{\dot{\mathsf{X}} = \sum_{i \in 1\dots m} e_i \times (P_i(\mathsf{X}) - R_i(\mathsf{X}))\right\}_{\mathsf{X} \in \mathcal{S}}$

Example

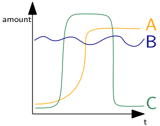
$\mathcal{S} = \{\mathsf{A}, \mathsf{B}, \mathsf{C}\}$

$\mathcal{R}_1 : \mathsf{A} + \mathsf{B} \xrightarrow{e_1} 2 \times \mathsf{C}$

$\mathcal{R}_2 : \mathsf{A} + \mathsf{C} \xrightarrow{e_2} \mathsf{A} + \mathsf{B}$



$$\begin{cases} \dot{\mathsf{A}} &= -1 \times e_1 \\ \dot{\mathsf{B}} &= -1 \times e_1 + 1 \times e_2 \\ \dot{\mathsf{C}} &= 2 \times e_1 + (-1) \times e_2 \end{cases}$$



Boolean network, structure and dynamics

One transition function per species in $\mathcal{S}$:
$\left\{f_\mathsf{X} : \mathbb{B}^{|\mathcal{S}|} \to \mathbb{B}\right\}_{\mathsf{X}\in\mathcal{S}}$ $\qquad \mathbb{B} = \{0, 1\}$

Example

$$\mathcal{S} = \{\mathsf{A}, \mathsf{B}, \mathsf{C}\}$$

$$\begin{aligned} f_\mathsf{A} &:= 0 \\ f_\mathsf{B} &:= (\mathsf{B} \land \neg\mathsf{C}) \lor (\neg\mathsf{B} \land \mathsf{C}) \\ f_\mathsf{C} &:= \neg\mathsf{C} \end{aligned}$$

Boolean network, structure and dynamics

One transition function per species in $\mathcal{S}$:
$\left\{f_{\mathsf{X}} : \mathbb{B}^{|\mathcal{S}|} \to \mathbb{B}\right\}_{\mathsf{X}\in\mathcal{S}}$ $\qquad \mathbb{B} = \{0, 1\}$

Influence graph
$IG = (\mathcal{S}, E \subseteq \mathcal{S} \times \mathcal{S}, \sigma : E \to \{+, -, \pm\})$

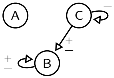
Example

$$\mathcal{S} = \{\mathsf{A}, \mathsf{B}, \mathsf{C}\}$$

$f_{\mathsf{A}} := 0$

$f_{\mathsf{B}} := (\mathsf{B} \wedge \neg\mathsf{C}) \vee (\neg\mathsf{B} \wedge \mathsf{C})$

$f_{\mathsf{C}} := \neg\mathsf{C}$



Boolean network, structure and dynamics

One transition function per species in $\mathcal{S}$:
$\left\{ f_X : \mathbb{B}^{|\mathcal{S}|} \to \mathbb{B} \right\}_{X \in \mathcal{S}}$ $\quad \mathbb{B} = \{0, 1\}$

Influence graph
$IG = (\mathcal{S}, E \subseteq \mathcal{S} \times \mathcal{S}, \sigma : E \to \{+, -, \pm\})$

Transition graph (TG)
$(\mathbb{B}^{|\mathcal{S}|}, E \subseteq \mathbb{B}^{|\mathcal{S}|} \times \mathbb{B}^{|\mathcal{S}|})$
general asynchronous update scheme:
$\mathcal{P}(\mathcal{S}) \setminus \emptyset$

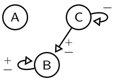
Example

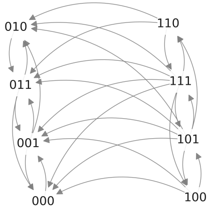
$\mathcal{S} = \{A, B, C\}$

$f_A := 0$

$f_B := (B \wedge \neg C) \vee (\neg B \wedge C)$

$f_C := \neg C$



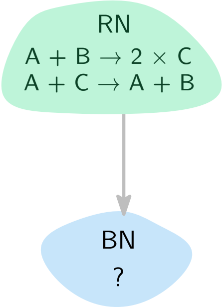


Outline

My method and its guarantees

(where constraints pop in)

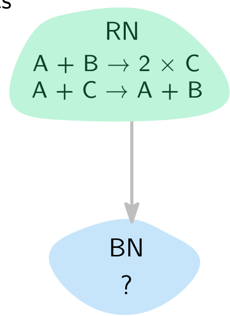
From RN to BN: which constraints



From RN to BN: which constraints

structure constraints

dynamics constraints

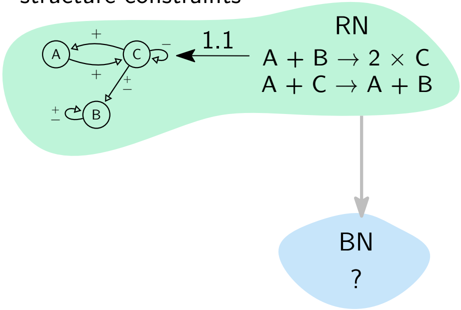


STEP 1: Retrieve constraints from the input RN

From RN to BN: which constraints

structure constraints

dynamics constraints

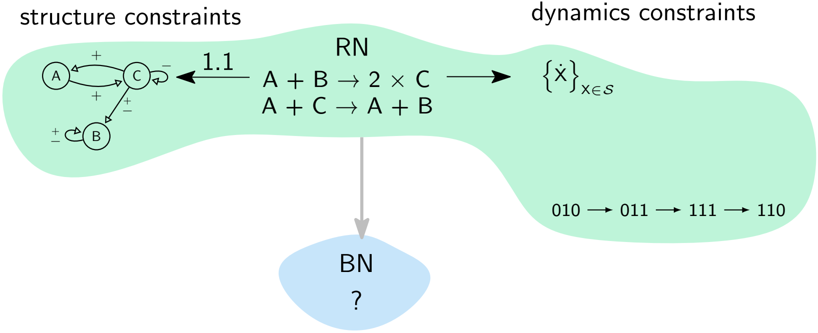


STEP 1: Retrieve constraints from the input RN

Structure: influence graph

1.1: syntactic parsing of the RN

From RN to BN: which constraints

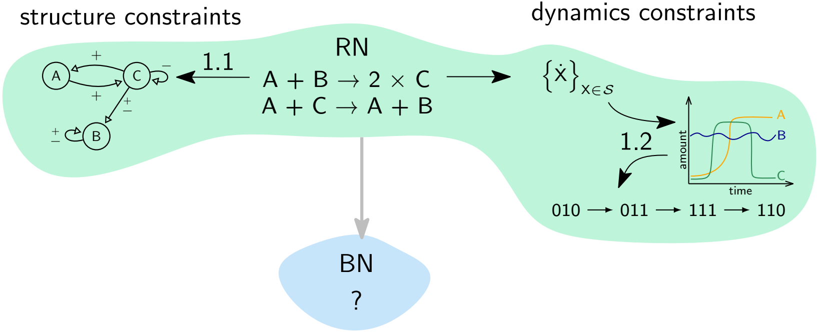


STEP 1: Retrieve constraints from the input RN

Structure: influence graph **Dynamics**: Boolean transitions

1.1: syntactic parsing of the RN

From RN to BN: which constraints



STEP 1: Retrieve constraints from the input RN

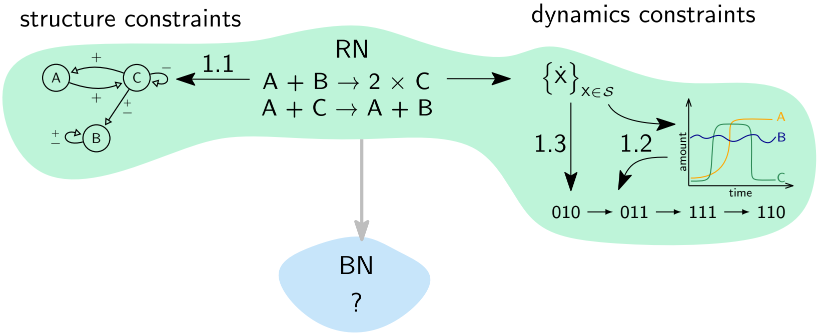
Structure: influence graph

1.1: syntactic parsing of the RN

Dynamics: Boolean transitions

1.2: ODEs simulation + binarisation

From RN to BN: which constraints



STEP 1: Retrieve constraints from the input RN

Structure: influence graph

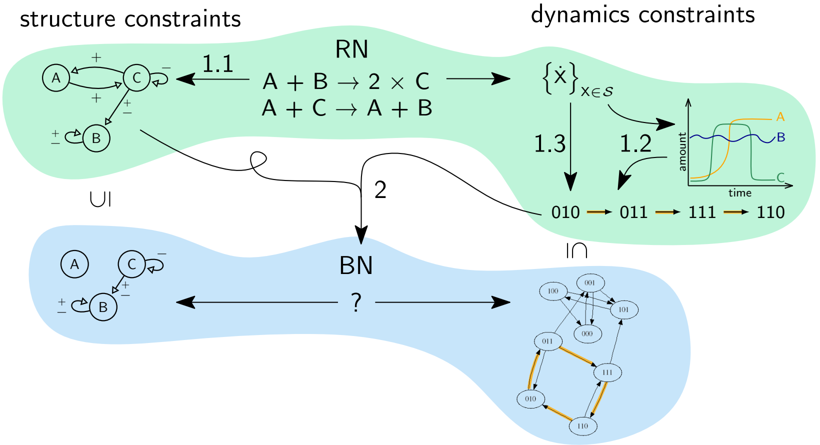
1.1: syntactic parsing of the RN

Dynamics: Boolean transitions

1.2: ODEs simulation + binarisation

1.3: abstract simulation of the ODEs

From RN to BN: which constraints



STEP 1: Retrieve constraints from the input RN

Structure: influence graph

1.1: syntactic parsing of the RN

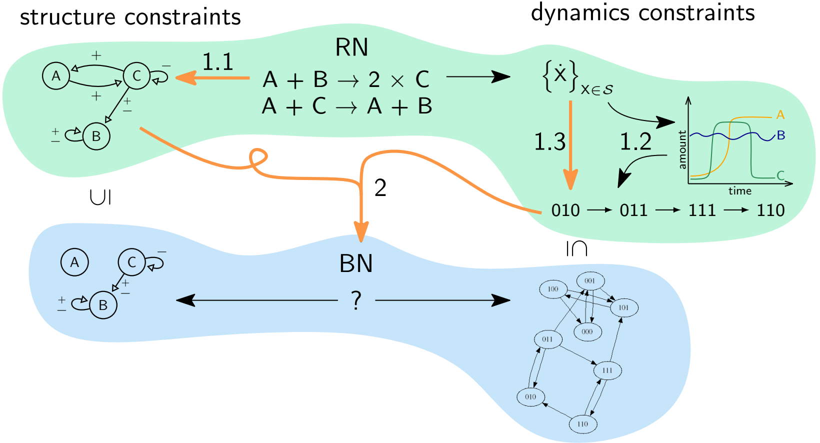
Dynamics: Boolean transitions

1.2: ODEs simulation + binarisation

1.3: abstract simulation of the ODEs

STEP 2: BN synthesis

From RN to BN: which constraints



STEP 1: Retrieve constraints from the input RN

Structure: influence graph

1.1: syntactic parsing of the RN

Dynamics: Boolean transitions

1.2: ODEs simulation + binarisation

1.3: abstract simulation of the ODEs

STEP 2: BN synthesis

Running example $\mathcal{R}_{\mathrm{enz}}$

$$\overbrace{\underbrace{\mathsf{E}+\mathsf{S}\underset{e_{\mathrm{off}}}{\overset{e_{\mathrm{on}}}{\rightleftarrows}}\mathsf{C}}_{\mathcal{R}_{\mathrm{off}}}}^{\mathcal{R}_{\mathrm{on}}}\overbrace{\xrightarrow{e_{\mathrm{cat}}}\mathsf{E}+2\times\mathsf{P}}^{\mathcal{R}_{\mathrm{cat}}}$$

Its ODEs (reconstructed)

$$\begin{cases} \dot{\mathsf{S}} = -e_{\mathrm{on}} + e_{\mathrm{off}} \\ \dot{\mathsf{E}} = -e_{\mathrm{on}} + e_{\mathrm{off}} + e_{\mathrm{cat}} \\ \dot{\mathsf{C}} = e_{\mathrm{on}} - e_{\mathrm{off}} + e_{\mathrm{cat}} \\ \dot{\mathsf{P}} = 2 \times e_{\mathrm{cat}} \end{cases}$$

Its parameters (given)

$$e_{\mathrm{on}} = 10^6 \times \mathsf{E} \times \mathsf{S}$$
$$e_{\mathrm{off}} = 0.2 \times \mathsf{C}$$
$$e_{\mathrm{cat}} = 0.1 \times \mathsf{C}$$

Outline

1. Preliminaries on reaction networks and Boolean networks
2. My method and its guarantees (where constraints pop in)

   STEP 1: Retrieve constraints from the input reaction network
   Structure: influence graph
   - 1.1: syntactic parsing of the reactions

   Dynamics: Boolean transitions
   - 1.2: ODEs simulation + binarisation
   - 1.3: abstract simulation of the ODEs [Niehren et al., 2022]

   STEP 2: BN synthesis with `ASK&D-BN` [Vaginay et al., 2021]
3. Evaluation of the approach
4. Conclusion and perspectives

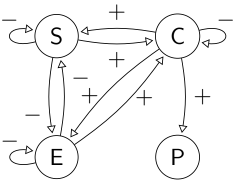
STEP 1: Retrieve an influence graph and Boolean transitions

Which contraints to build the influence graph $\mathcal{G}_{\mathcal{R}}$?

Constraints inspired from [Fages, Soliman, 2008b]

$$\mathsf{Y} \xrightarrow{-} \mathsf{X} \in \mathcal{G}_{\mathcal{R}} \text{ if } \exists \mathcal{R} = (R, e, P) : \mathsf{Y} \in R \text{ and } R(\mathsf{X}) > P(\mathsf{X})$$
$$\mathsf{Y} \xrightarrow{+} \mathsf{X} \in \mathcal{G}_{\mathcal{R}} \text{ if } \exists \mathcal{R} = (R, e, P) : \mathsf{Y} \in R \text{ and } R(\mathsf{X}) < P(\mathsf{X})$$

$\mathcal{G}_{\mathcal{R}_{\text{enz}}}$



Guaranty: Overapproximates the possible signs of $\frac{\partial \mathsf{X}}{\partial \mathsf{Y}}$
$\rightarrow$ capture all the direct influences between the species

Which constraints to retrieve Boolean transitions from $\mathcal{R}$?

Abstract simulation — Intuition

Joint work with Joachim Niehren and Cristian Versari [Niehren et al., 2022]

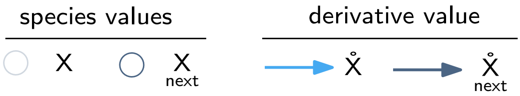
Use the rule of signs to reason on the causal relationship between the signs ($\mathbb{S} = \{-1, 0, 1\}$) of the variables values of the ODE system

Which constraints to retrieve Boolean transitions from $\mathcal{R}$?

Abstract simulation — Intuition

Joint work with Joachim Niehren and Cristian Versari [Niehren et al., 2022]

Use the rule of signs to reason on the causal relationship between the signs ($\mathbb{S} = \{-1, 0, 1\}$) of the variables values of the ODE system

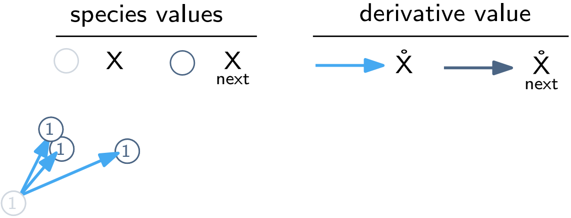


Which constraints to retrieve Boolean transitions from $\mathcal{R}$?

Abstract simulation — Intuition

Joint work with Joachim Niehren and Cristian Versari [Niehren et al., 2022]

Use the rule of signs to reason on the causal relationship between the signs ($\mathbb{S} = \{-1, 0, 1\}$) of the variables values of the ODE system



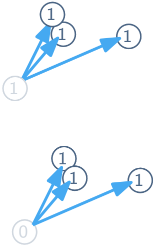
Which constraints to retrieve Boolean transitions from $\mathcal{R}$?

Abstract simulation — Intuition

Joint work with Joachim Niehren and Cristian Versari [Niehren et al., 2022]

Use the rule of signs to reason on the causal relationship between the signs ($\mathbb{S} = \{-1, 0, 1\}$) of the variables values of the ODE system

species values		derivative value	
X	X_{next}	$\mathring{X}$	$\mathring{X}_{next}$

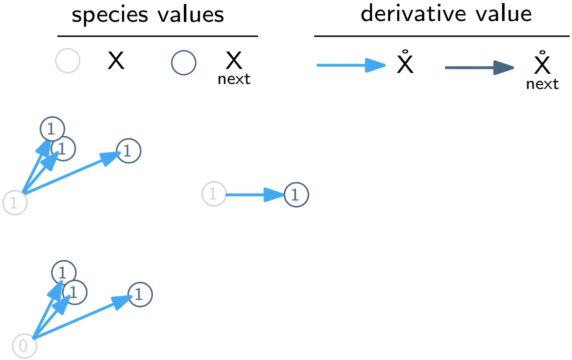


Which constraints to retrieve Boolean transitions from $\mathcal{R}$?

Abstract simulation — Intuition

Joint work with Joachim Niehren and Cristian Versari [Niehren et al., 2022]

Use the rule of signs to reason on the causal relationship between the signs ($\mathbb{S} = \{-1, 0, 1\}$) of the variables values of the ODE system

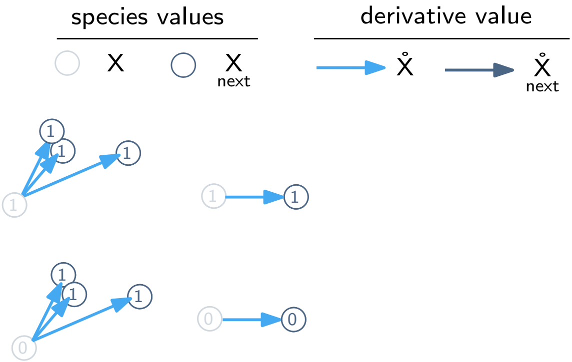


Which constraints to retrieve Boolean transitions from $\mathcal{R}$?

Abstract simulation — Intuition

Joint work with Joachim Niehren and Cristian Versari [Niehren et al., 2022]

Use the rule of signs to reason on the causal relationship between the signs ($\mathbb{S} = \{-1, 0, 1\}$) of the variables values of the ODE system

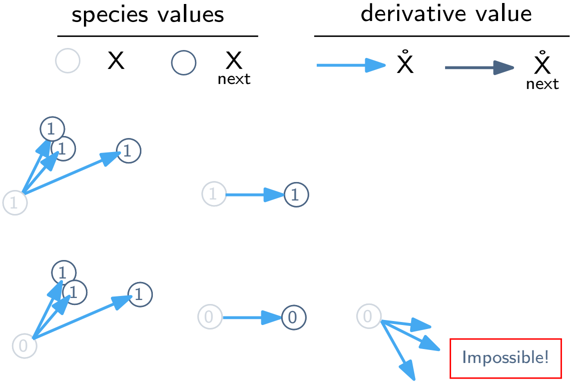


Which constraints to retrieve Boolean transitions from $\mathcal{R}$?

Abstract simulation — Intuition

Joint work with Joachim Niehren and Cristian Versari [Niehren et al., 2022]

Use the rule of signs to reason on the causal relationship between the signs ($\mathbb{S} = \{-1, 0, 1\}$) of the variables values of the ODE system

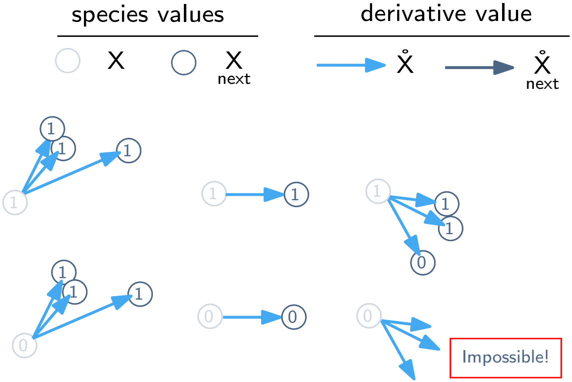


Which constraints to retrieve Boolean transitions from $\mathcal{R}$?

Abstract simulation — Intuition

Joint work with Joachim Niehren and Cristian Versari [Niehren et al., 2022]

Use the rule of signs to reason on the causal relationship between the signs ($\mathbb{S} = \{-1, 0, 1\}$) of the variables values of the ODE system



X was above 0 and its derivative was negative

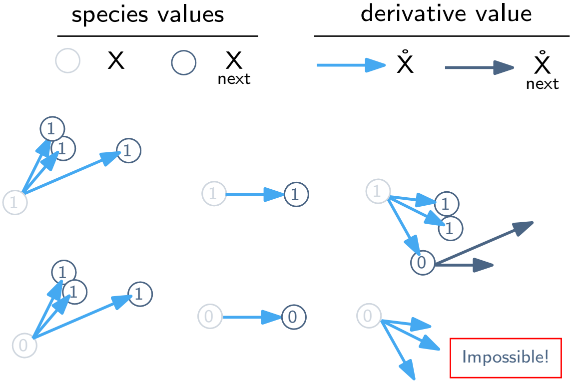
plus − *plus* = *unknown* $\rightsquigarrow$ nondeterminism

Which constraints to retrieve Boolean transitions from $\mathcal{R}$?

Abstract simulation — Intuition

Joint work with Joachim Niehren and Cristian Versari [Niehren et al., 2022]

Use the rule of signs to reason on the causal relationship between the signs ($\mathbb{S} = \{-1, 0, 1\}$) of the variables values of the ODE system



X was above 0 and its derivative was negative

plus − *plus* = *unknown* $\rightsquigarrow$ nondeterminism

Which constraints to retrieve Boolean transitions from $\mathcal{R}$?

Abstract simulation — In practice

Contribution

$$\mathcal{V} = \bigcup_{\mathsf{X} \in \mathcal{S}} \left\{ \mathsf{X}, \mathring{\mathsf{X}}, \underset{\text{next}}{\mathsf{X}}, \underset{\text{next}}{\mathring{\mathsf{X}}} \right\}$$

- Causal relationships encoded by a first-order logic formula ϕ
- Solve ϕ on $\mathbb{S} = \{-1, 0, 1\}$

$\rightsquigarrow$ relation $\mathbb{B}^{\left|\mathcal{S} \cup \mathring{\mathcal{S}}\right|} \times \mathbb{B}^{\left|\underset{\text{next}}{\mathcal{S}} \cup \underset{\text{next}}{\mathring{\mathcal{S}}}\right|}$

- Restrict the solutions on $\mathcal{S} \cup \underset{\text{next}}{\mathcal{S}}$

$\rightsquigarrow$ relation $\mathbb{B}^{|\mathcal{S}|} \times \mathbb{B}^{\left|\underset{\text{next}}{\mathcal{S}}\right|}$

Guarantee

- Keep the causalities of changes
- Proof of correctness: overapproximation of an ideal Euler simulation (perfectly adjusted time step and no computation error)

FOBNN: First-Order Boolean networks with nondeterministic updates

Which constraints to retrieve Boolean transitions from $\mathcal{R}$?

Abstract simulation — Example on $\mathcal{R}_{\text{enz}}$

$$
\begin{array}{llllll}
 & \mathring{S} = -e_{on} + e_{off} & & \wedge & \underset{\text{next}}{\mathring{S}} = -\underset{\text{next}}{e_{on}} + \underset{\text{next}}{e_{off}} & \\
\wedge & \mathring{E} = -e_{on} + e_{off} + e_{cat} & & \wedge & \underset{\text{next}}{\mathring{E}} = -\underset{\text{next}}{e_{on}} + \underset{\text{next}}{e_{off}} + \underset{\text{next}}{e_{cat}} & \\
\wedge & \mathring{C} = e_{on} - e_{off} - e_{cat} & & \wedge & \underset{\text{next}}{\mathring{C}} = \underset{\text{next}}{e_{on}} - \underset{\text{next}}{e_{off}} - \underset{\text{next}}{e_{cat}} & \\
\wedge & \mathring{P} = e_{cat} & & \wedge & \underset{\text{next}}{\mathring{P}} = \underset{\text{next}}{e_{cat}} &
\end{array}
$$

$$
\begin{array}{llll}
\wedge & \underset{\text{next}}{S} = S + \mathring{S} & \wedge & S \leq \underset{\text{next}}{S} \\
\wedge & \underset{\text{next}}{E} = E + \mathring{E} & \wedge & E \leq \underset{\text{next}}{E} \\
\wedge & \underset{\text{next}}{C} = C + \mathring{C} & \wedge & C \leq \underset{\text{next}}{C} \\
\wedge & \underset{\text{next}}{P} = P + \mathring{P} & \wedge & P \leq \underset{\text{next}}{P}
\end{array}
$$

with

$$
\begin{array}{lll}
e_{on} = 10^6 \times S \times E & e_{off} = 0.2 \times C & e_{cat} = 0.1 \times C \\
\underset{\text{next}}{e_{on}} = 10^6 \times \underset{\text{next}}{S} \times \underset{\text{next}}{E} & \underset{\text{next}}{e_{off}} = 0.2 \times \underset{\text{next}}{C} & \underset{\text{next}}{e_{cat}} = 0.1 \times \underset{\text{next}}{C}
\end{array}
$$

Which constraints to retrieve Boolean transitions from $\mathcal{R}$?

$$\overbrace{\underbrace{\mathsf{E} + \mathsf{S} \underset{e_{\mathrm{off}}}{\overset{e_{\mathrm{on}}}{\rightleftarrows}} \mathsf{C}}_{\mathcal{R}_{\mathrm{off}}}}^{\mathcal{R}_{\mathrm{on}}} \overbrace{\xrightarrow{e_{\mathrm{cat}}} \mathsf{E} + 2 \times \mathsf{P}}^{\mathcal{R}_{\mathrm{cat}}}$$

Expected transitions [SECP]:
1100 → **10 → ***1

Which constraints to retrieve Boolean transitions from $\mathcal{R}$?

$$\overbrace{\underbrace{\mathsf{E} + \mathsf{S} \underset{e_{\mathrm{off}}}{\overset{e_{\mathrm{on}}}{\rightleftarrows}} \mathsf{C}}_{\mathcal{R}_{\mathrm{off}}}}^{\mathcal{R}_{\mathrm{on}}} \overbrace{\xrightarrow{e_{\mathrm{cat}}} \mathsf{E} + 2 \times \mathsf{P}}^{\mathcal{R}_{\mathrm{cat}}}$$

Expected transitions [SECP]:
1100 → **10 → ***1

Which constraints to retrieve Boolean transitions from $\mathcal{R}$?

$$\overbrace{\underbrace{\mathsf{E} + \mathsf{S} \underset{e_{\text{off}}}{\overset{e_{\text{on}}}{\rightleftarrows}} \mathsf{C}}_{\mathcal{R}_{\text{off}}}}^{\mathcal{R}_{\text{on}}} \overbrace{\xrightarrow{e_{\text{cat}}} \mathsf{E} + 2 \times \mathsf{P}}^{\mathcal{R}_{\text{cat}}}$$

Expected transitions [SECP]:
1100 → **10 → ***1

Which constraints to retrieve Boolean transitions from $\mathcal{R}$?

$$\overbrace{\underbrace{\mathsf{E}+\mathsf{S}\overset{e_{\mathrm{on}}}{\underset{e_{\mathrm{off}}}{\rightleftarrows}}\mathsf{C}}_{\mathcal{R}_{\mathrm{off}}}}^{\mathcal{R}_{\mathrm{on}}}\overbrace{\xrightarrow{e_{\mathrm{cat}}}\mathsf{E}+2\times\mathsf{P}}^{\mathcal{R}_{\mathrm{cat}}}$$

Expected transitions [SECP]:
1100 → **10 → ***1

Which constraints to retrieve Boolean transitions from $\mathcal{R}$?

$$\overbrace{\underbrace{\mathsf{E}+\mathsf{S}\underset{e_{\mathrm{off}}}{\overset{e_{\mathrm{on}}}{\rightleftarrows}}\mathsf{C}}_{\mathcal{R}_{\mathrm{off}}}}^{\mathcal{R}_{\mathrm{on}}}\overbrace{\xrightarrow{e_{\mathrm{cat}}}\mathsf{E}+2\times\mathsf{P}}^{\mathcal{R}_{\mathrm{cat}}}$$

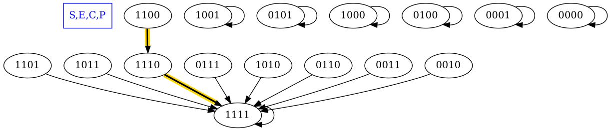
Expected transitions [SECP]:
1100 $\rightarrow$ **10 $\rightarrow$ ***1

Which constraints to retrieve Boolean transitions from $\mathcal{R}$?

$$\overbrace{\underbrace{\mathsf{E}+\mathsf{S}\underset{e_{\mathrm{off}}}{\overset{e_{\mathrm{on}}}{\rightleftarrows}}\mathsf{C}}_{\mathcal{R}_{\mathrm{off}}}}^{\mathcal{R}_{\mathrm{on}}}\overbrace{\xrightarrow{e_{\mathrm{cat}}}\mathsf{E}+2\times\mathsf{P}}^{\mathcal{R}_{\mathrm{cat}}}$$

Expected transitions [SECP]:
1100 → **10 → ***1

Abstract simulation



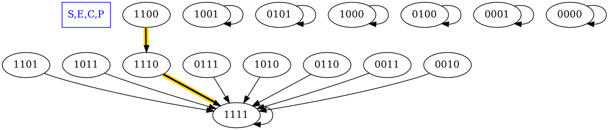
Which constraints to retrieve Boolean transitions from $\mathcal{R}$?

$$\overbrace{E + S \underset{e_{\text{off}}}{\overset{e_{\text{on}}}{\rightleftarrows}} C}^{\mathcal{R}_{\text{on}}}\overbrace{\xrightarrow{e_{\text{cat}}} E + 2 \times P}^{\mathcal{R}_{\text{cat}}}$$

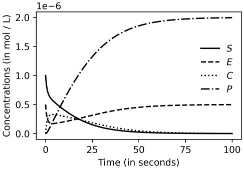
$$\underbrace{E + S \rightleftarrows C}_{\mathcal{R}_{\text{off}}}$$

Expected transitions [SECP]:
1100 $\rightarrow$ **10 $\rightarrow$ ***1

Abstract simulation



Classic ODE simulation + Binarisation



Binarisation	Boolean configuration sequence [SECP]
Midrange	1100 → 1000 → 1010 → 0010 → 0011 → 0101
Median	1100 → 1010 → 0011 → 0101
Mean	1100 → 1010 → 1000 → 0011 → 0101
Above 0	1100 → 1111 → 1011 → 1111 → 0111

Outline

1. Preliminaries on reaction networks and Boolean networks
2. My method and its guarantees (where constraints pop in)

STEP 1: Retrieve constraints from the input reaction network

Structure: influence graph

- 1.1: syntactic parsing of the reactions

Dynamics: Boolean transitions

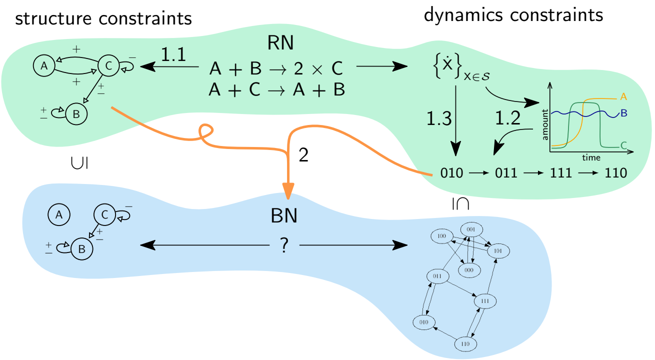
- 1.2: ODEs simulation + binarisation
- 1.3: abstract simulation of the ODEs [Niehren et al., 2022]

STEP 2: BN synthesis with `ASK&D-BN` [Vaginay et al., 2021]

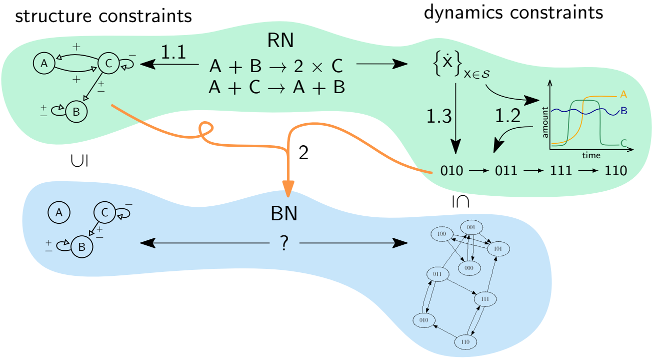
3. Evaluation of the approach
4. Conclusion and perspectives

STEP 2: Boolean network synthesis with `ASK&D-BN`

ASK&D-BN [Vaginay et al., 2021]



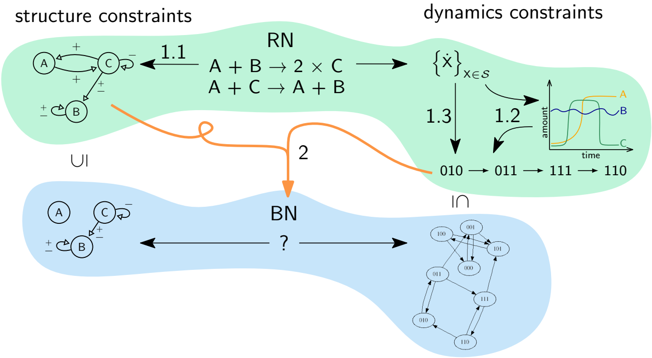
ASK&D-BN [Vaginay et al., 2021]



1. **Local search** species-wise synthesis of *all* the transition functions compatible with the given influence graph and time series

> Generate candidates → Structure constraint → Dynamic constraint → Minimality constraint

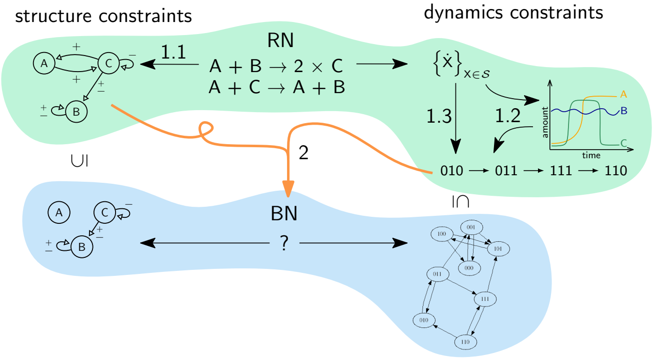
ASK&D-BN [Vaginay et al., 2021]



1. **Local search** species-wise synthesis of *all* the transition functions compatible with the given influence graph and time series

> Generate candidates → Structure constraint → Dynamic constraint → Minimality constraint
> Answer-Set Programming

ASK&D-BN [Vaginay et al., 2021]



1. **Local search** species-wise synthesis of *all* the transition functions compatible with the given influence graph and time series

   Generate candidates → Structure constraint → Dynamic constraint → Minimality constraint
   Answer-Set Programming

2. **Global assembly** produce all the possible BNs

ASK&D-BN— Local search

Generate candidates → Structure constraint → Dynamic constraint → Minimality constraint

Search space: 2^{3^k} non-redundant DNF = non-redundant disjunction of non-redundant conjunctions

ideally: the set of minimal DNF with k inputs.

ASK&D-BN— Local search

Generate candidates $\rightarrow$ Structure constraint $\rightarrow$ Dynamic constraint $\rightarrow$ Minimality constraint

Search space: 2^{3^k} non-redundant DNF $=$ non-redundant disjunction of non-redundant conjunctions

ideally: the set of minimal DNF with k inputs.

Pick a subset of non-redundant conjunctions
without subsomption and not locally-adjacent

ASK&D-BN— Local search

Generate candidates → Structure constraint → Dynamic constraint → Minimality constraint

Search space: 2^{3^k} non-redundant DNF = non-redundant disjunction of non-redundant conjunctions
ideally: the set of minimal DNF with k inputs.

> Pick a subset of non-redundant conjunctions without subsomption and not locally-adjacent

Examples

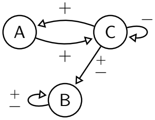
invalid candidates:
$(A \land \neg B) \lor (A \land \neg B) \lor (\neg A \land \neg C)$
$(A \land A \land \neg B) \lor (\neg A \land \neg C)$
$(A) \lor (A \land B)$
$(A \land B) \lor (A \land \neg B)$

valid candidate:
$(A \land \neg B) \lor (\neg A \land \neg C)$

ASK&D-BN— Local search

Generate candidates → Structure constraint → Dynamic constraint → Minimality constraint

influence graph of the Boolean network ⊆ influence graph of the reaction network

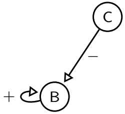


Do not select a conjunction that uses a forbidden literal

Examples

invalid conjunction: $\neg A \wedge \neg C$

valid conjunction: $\neg C \wedge B$



ASK&D-BN— Local search

Generate candidates → Structure constraint → Dynamic constraint → Minimality constraint

(1) input: Boolean transitions

Build partial truth tables for each species X: what were the values of its putative inputs **when its value changed**? ⇝ Do not assume the underlying update scheme
Compare the truth table of a candidate function to the reconstructed truth table

putative input	output

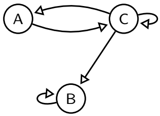
ASK&D-BN— Local search

Generate candidates → Structure constraint → Dynamic constraint → Minimality constraint

(1) input: Boolean transitions

Build partial truth tables for each species X: what were the values of its putative inputs **when its value changed**? ⇝ Do not assume the underlying update scheme
Compare the truth table of a candidate function to the reconstructed truth table

input influence graph (unsigned)



putative input	output
C	A
BC	B
AC	C

ASK&D-BN— Local search

Generate candidates → Structure constraint → Dynamic constraint → Minimality constraint

(1) input: Boolean transitions

Build partial truth tables for each species X: what were the values of its putative inputs **when its value changed**? ⇝ Do not assume the underlying update scheme
Compare the truth table of a candidate function to the reconstructed truth table

$$010 \xrightarrow{1} 011 \xrightarrow{2} 100 \xrightarrow{3} 001$$

putative input	output
C	A
BC	B
AC	C

ASK&D-BN— Local search

Generate candidates → Structure constraint → Dynamic constraint → Minimality constraint

(1) input: Boolean transitions

Build partial truth tables for each species X: what were the values of its putative inputs **when its value changed**? ⇝ Do not assume the underlying update scheme
Compare the truth table of a candidate function to the reconstructed truth table

$$010 \xrightarrow[1]{C} 011 \xrightarrow[2]{A,\ B,\ C} 100 \xrightarrow[3]{A,\ C} 001$$

putative input	output
C	A
BC	B
AC	C

ASK&D-BN— Local search

Generate candidates → Structure constraint → Dynamic constraint → Minimality constraint

(1) input: Boolean transitions

Build partial truth tables for each species X: what were the values of its putative inputs **when its value changed**? ⇝ Do not assume the underlying update scheme
Compare the truth table of a candidate function to the reconstructed truth table

$$010 \xrightarrow[1]{C} 011 \xrightarrow[2]{A,\,B,\,C} 100 \xrightarrow[3]{A,\,C} 001$$

putative input	output	
C	A	
1	1	2
BC	B	
AC	C	

ASK&D-BN— Local search

Generate candidates → Structure constraint → Dynamic constraint → Minimality constraint

(1) input: Boolean transitions

Build partial truth tables for each species X: what were the values of its putative inputs **when its value changed**? ⇝ Do not assume the underlying update scheme
Compare the truth table of a candidate function to the reconstructed truth table

$$010 \xrightarrow[1]{C} 011 \xrightarrow[2]{A,\ B,\ C} 100 \xrightarrow[3]{A,\ C} 001$$

putative input	output	
C	A	
0	0	3
1	1	2
BC	B	
AC	C	

ASK&D-BN— Local search

Generate candidates → Structure constraint → Dynamic constraint → Minimality constraint

(1) input: Boolean transitions

Build partial truth tables for each species X: what were the values of its putative inputs **when its value changed**? ⇝ Do not assume the underlying update scheme
Compare the truth table of a candidate function to the reconstructed truth table

$$010 \xrightarrow[1]{C} 011 \xrightarrow[2]{A,\ B,\ C} 100 \xrightarrow[3]{A,\ C} 001$$

putative input	output	
C	A	
0	0	3
1	1	2

BC	B	
11	0	2

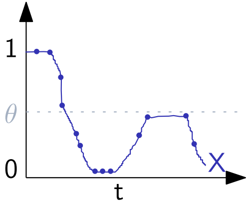
AC	C	
00	1	1
01	0	2
10	1	3

ASK&D-BN— Local search

Generate candidates → Structure constraint → Dynamic constraint → Minimality constraint

(2) input: time series

X_t: continuous value of X at time t
θ: binarisation threshold for X

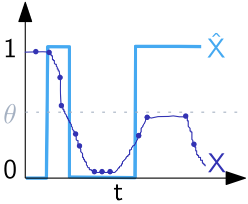


ASK&D-BN— Local search

Generate candidates → Structure constraint → Dynamic constraint → Minimality constraint

(2) input: time series

X_t: continuous value of X at time t
θ: binarisation threshold for X

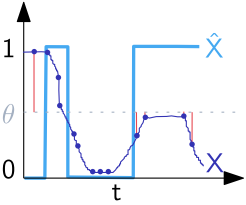


ASK&D-BN— Local search

Generate candidates → Structure constraint → Dynamic constraint → Minimality constraint

(2) input: time series

X_t: continuous value of X at time t
θ: binarisation threshold for X
$\mathcal{U}$: set of unexplained time steps

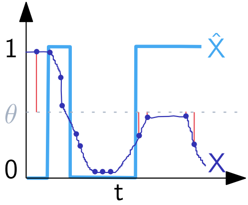


ASK&D-BN— Local search

Generate candidates → Structure constraint → Dynamic constraint → Minimality constraint

(2) input: time series

X_t: continuous value of X at time t
θ: binarisation threshold for X
$\mathcal{U}$: set of unexplained time steps
$E = \sum_{t \in \mathcal{U}} |\theta - X_t|$ To minimise (ideally 0)



ASK&D-BN— Local search

Generate candidates → Structure constraint → Dynamic constraint → Minimality constraint

Select candidates with the smallest expressions (subset and/or cardinal minimal) ⇝ most general conditions

putative input	observed output
AB	X
00	
01	0
10	1
11	

ASK&D-BN— Local search

Generate candidates → Structure constraint → Dynamic constraint → Minimality constraint

Select candidates with the smallest expressions (subset and/or cardinal minimal) ⇝ most general conditions

putative input	observed output	possible completions			
AB	X				
00		0	1	0	1
01	0	0	0	0	0
10	1	1	1	1	1
11		0	0	1	1

ASK&D-BN— Local search

Generate candidates → Structure constraint → Dynamic constraint → Minimality constraint

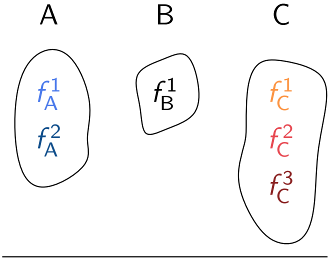
Select candidates with the smallest expressions (subset and/or cardinal minimal) ⇝ most general conditions

putative input AB	observed output X	possible completions			
00		0	1	0	1
01	0	0	0	0	0
10	1	1	1	1	1
11		0	0	1	1
	subset minimal candidates	$A \wedge \neg B$	$\neg B$	A	$A \vee \neg B$
	size	2	1	1	2

card. min. candidates

ASK&D-BN— Global assembly

Cartesian product of the set of transition functions synthesised for each species



$$\mathcal{B}_1 = \{ f_A^1, f_B^1, f_C^1 \}$$
$$\mathcal{B}_2 = \{ f_A^1, f_B^1, f_C^2 \}$$
$$\mathcal{B}_3 = \{ f_A^1, f_B^1, f_C^3 \}$$
$$\mathcal{B}_4 = \{ f_A^2, f_B^1, f_C^1 \}$$
$$\mathcal{B}_5 = \{ f_A^2, f_B^1, f_C^2 \}$$
$$\mathcal{B}_6 = \{ f_A^2, f_B^1, f_C^3 \}$$

Outline

Evaluation of the approach

Evaluation of the approach

1. The BN synthesis itself [Vaginay et al., 2021]
   `ASK&D-BN` versus `REVEAL`[1], `Best-Fit`[2] and `Caspo-TS`[3]

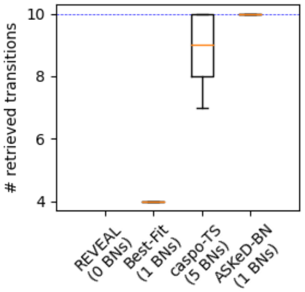
2. One specific variant of the complete approach on real-world reaction networks [Vaginay et al., 2021, Vaginay et al., 2022]
   influence graph + time series and midrange binarisation

3. Several variants of the complete approach on $\mathcal{R}_{\text{enz}}$
   compare concrete and abstract simulation

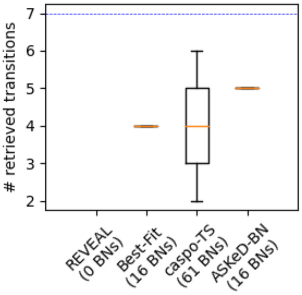
[1] [Liang et al., 1998] [2] [Lähdesmäki et al., 2003] [3] [Ostrowski et al., 2016]

Evaluation of the BN synthesis step

A. thaliana
5 species, 10 transitions

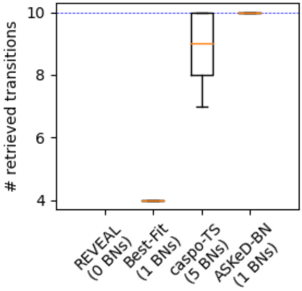


yeast
4 species, 7 transitions

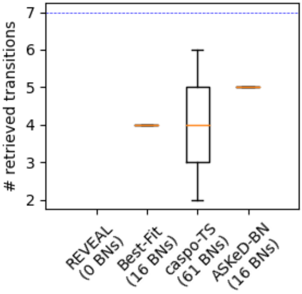


Evaluation of the BN synthesis step

A. thaliana
5 species, 10 transitions



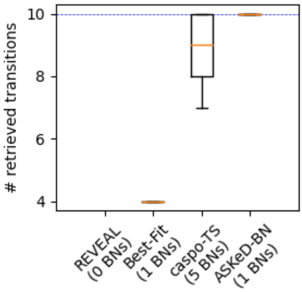
yeast
4 species, 7 transitions



- `REVEAL` fails

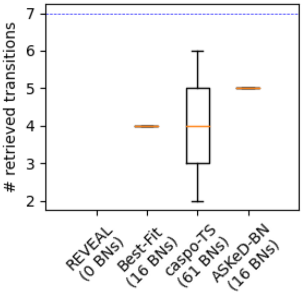
Evaluation of the BN synthesis step

A. thaliana
5 species, 10 transitions



- `REVEAL` fails

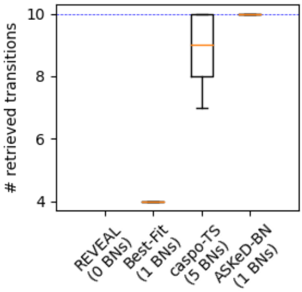
yeast
4 species, 7 transitions



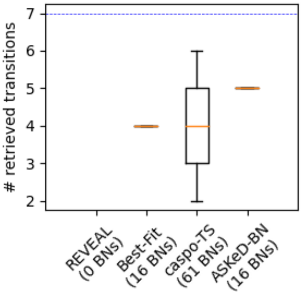
- `Best-Fit` lacks consistency

Evaluation of the BN synthesis step

A. thaliana
5 species, 10 transitions



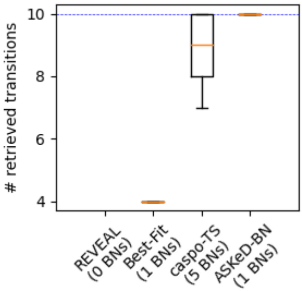
yeast
4 species, 7 transitions



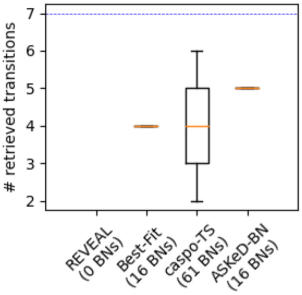
- `REVEAL` fails
- `Caspo-TS` returns more BNs, some of them with poor coverage because of reachability constraint
- `Best-Fit` lacks consistency

Evaluation of the BN synthesis step

A. thaliana
5 species, 10 transitions



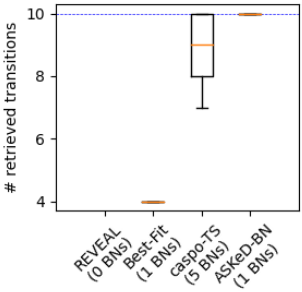
yeast
4 species, 7 transitions



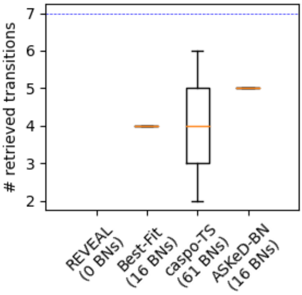
- `REVEAL` fails
- `Caspo-TS` returns more BNs, some of them with poor coverage because of reachability constraint
- `Best-Fit` lacks consistency
- `ASK&D-BN` returns a small number of BN, with good coverage and low variance ✓

Evaluation of the BN synthesis step

A. thaliana
5 species, 10 transitions



yeast
4 species, 7 transitions



- `REVEAL` fails
- `Caspo-TS` returns more BNs, some of them with poor coverage because of reachability constraint
- `Best-Fit` lacks consistency
- `ASK&D-BN` returns a small number of BN, with good coverage and low variance ✓

⇝ Confirmed on $>$ 300 datasets generated from existing BNs from the repository of PyBoolNet

Outline

1. Preliminaries on reaction networks and Boolean networks
2. My method and its guarantees (where constraints pop in)
3. Evaluation of the approach
4. Conclusion and perspectives

Conclusion and perspectives

Conclusion

Automatic synthesis of Boolean networks from a given reaction network, with guarantees. ✓

Conclusion

Automatic synthesis of Boolean networks from a given reaction network, with guarantees. ✓

- Methodology: Boolean networks synthesis from constraints

  Structure: Influence graph from syntactic parsing of the reactions

  - captures all the direct influences among species

  Dynamics: Boolean transitions

  from numerical simulation of the ODEs + binarisation

  - good approximation or the analytical solution
  - but we lose causality

  from abstract simulation of the ODEs

  - correct overapproximation of perfect Euler that captures causality

Conclusion

Automatic synthesis of Boolean networks from a given reaction network, with guarantees. ✓

- Methodology: Boolean networks synthesis from constraints

  Structure: Influence graph from syntactic parsing of the reactions

  - captures all the direct influences among species

  Dynamics: Boolean transitions

  from numerical simulation of the ODEs + binarisation

  - good approximation or the analytical solution
  - but we lose causality

  from abstract simulation of the ODEs

  - correct overapproximation of perfect Euler that captures causality

- Implementation: the SBML2BNET pipeline (+ `ASK&D-BN`)

Conclusion

Automatic synthesis of Boolean networks from a given reaction network, with guarantees. ✓

- Methodology: Boolean networks synthesis from constraints

  Structure: Influence graph from syntactic parsing of the reactions

  - captures all the direct influences among species

  Dynamics: Boolean transitions

  from numerical simulation of the ODEs + binarisation

  - good approximation or the analytical solution
  - but we lose causality

  from abstract simulation of the ODEs

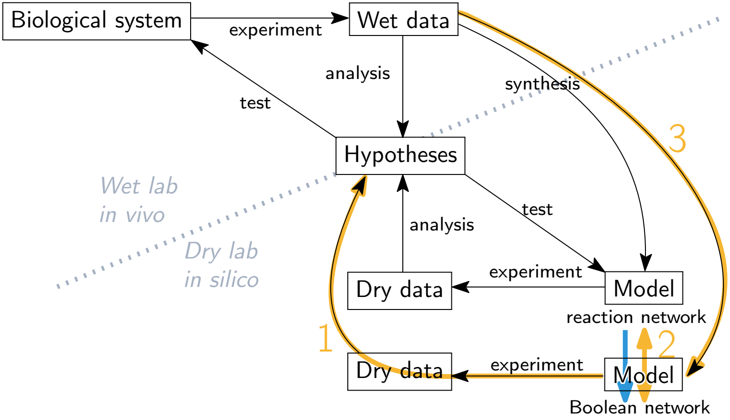
  - correct overapproximation of perfect Euler that captures causality

- Implementation: the SBML2BNET pipeline (+ `ASK&D-BN`)

- Evaluation

From reactions to Boolean influences with guarantees

Why?



1. Use BNs to facilitate some analyses
2. Explore the formal relationship between RN and BN
3. Improve the BN synthesis methods

Perspectives

1. To facilitate some analyses
   Make SBML2BNET easy to use, use more evaluation criteria, include more knowledge in the synthesis, analyse FO-BNN themselves (process more RN, compute attractors(*))

2. Explore the formal relationship between RN and BN
   Two conjectures to investigate, reverse process(*)

3. Improve the BN synthesis methods
   Investigate, in a controled environnement
   - when we can't fullfill the constraints(*)
   - overfitting to *the* sequence of configuration?
   - impact of the choice of the binarisation procedure and error measure

Perspectives

1. To facilitate some analyses
   Make SBML2BNET easy to use, use more evaluation criteria, include more knowledge in the synthesis, **analyse FO-BNN themselves (process more RN, compute attractors(*))**
2. Explore the formal relationship between RN and BN
   Two conjectures to investigate, reverse process(*)
3. Improve the BN synthesis methods
   Investigate, in a controled environnement
   - **when we can't fullfill the constraints(*)**
   - overfitting to *the* sequence of configuration?
   - impact of the choice of the binarisation procedure and error measure

Publications

J. Niehren, C. Lhoussaine and **AV**. *Core SBML and its Formal Semantics* CMSB: International Conference on Computational Methods in Systems Biology 2023

Abstract simu. J. Niehren, **AV**, and C. Versari. *Abstract Simulation of Reaction Networks via Boolean Networks* CMSB: International Conference on Computational Methods in Systems Biology 2022

SBML2BNET **AV**, T. Boukhobza, and M. Smaïl-Tabbone. *From Quantitative SBML Models to Boolean Networks* CNA: Complex Networks & Their Applications X 2022

SBML2BNET **AV**, T. Boukhobza, and M. Smaïl-Tabbone. *From Quantitative SBML Models to Boolean Networks* Applied Network Science 2022

ASK&D-BN **AV**, T. Boukhobza, and M. Smaïl-Tabbone. *Automatic Synthesis of Boolean Networks from Biological Knowledge and Data* OLA: Optimization and Learning 2021

A. Hirtz, N. Lebourdais, F. Rech, Y. Bailly, **AV**, M. Smaïl-Tabbone, H. Dubois-Pot-Schneider, and H. Dumond. *GPER Agonist G-1 Disrupts Tubulin Dynamics and Potentiates Temozolomide to Impair Glioblastoma Cell Proliferation* Cells 2021

Thank you for your attention.

References I

- [Bornholdt, 2005]
  S. Bornholdt
  Less Is More in Modeling Large Genetic Networks,
  2005
- [Fages, Soliman, 2008a]
  F. Fages, S. Soliman,
  Abstract Interpretation and Types for Systems Biology,
  Theoretical Computer Science, vol. 403, pp. 52–70, 2008
- [Fages, Soliman, 2008b]
  F. Fages, S. Soliman,
  From Reaction Models to Influence Graphs and Back: A Theorem,
  Lecture Notes in Computer Science, pp. 90–102 2008
- [Hoops et al., 2006]
  S. Hoops et al.
  COPASI—a COmplex PAthway SImulator,
  Bioinformatics, vol. 22, pp. 3067–3074 2006

References II

- [Kohl et al., 2010]
  P. Kohl et al.
  Systems Biology: An Approach,
  Clinical Pharmacology & Therapeutics vol. 88-1 pp. 25–33 2010,
- [Lähdesmäki et al., 2003]
  H. Lähdesmäki et al.
  On Learning Gene Regulatory Networks under the Boolean Network Model,
  Machine Learning, vol. 52-1 pp. 147–167 2003,
- [Liang et al., 1998]
  S. Liang et al.
  REVEAL, a General Reverse Engineering Algorithm for Inference of Genetic Network Architectures
  Pacific Symposium on Biocomputing. pp. 18–29, 1998,
- [Malik-Sheriff et al., 2020]
  R. Malik-Sheriff et al.
  BioModels—15 Years of Sharing Computational Models in Life Science
  Nucleic Acids Research vol. 48-D1, pp. D407-D415, 2020

References III

- [Niehren et al., 2022]
  J. Niehren et al.
  Abstract Simulation of Reaction Networks via Boolean Networks
  CMSB: International Conference on Computational Methods in Systems Biology 2022,
- [Ostrowski et al., 2016]
  M. Ostrowski et al.
  Boolean Network Identification from Perturbation Time Series Data Combining Dynamics Abstraction and Logic Programming
  Biosystems vol. = 149, pp. 139–153, 2016
- [Vaginay et al., 2021]
  A. Vaginay, et al.
  Automatic Synthesis of Boolean Networks from Biological Knowledge and Data
  Communications in Computer and Information Science pp. 156–170, 2021

References IV

- [Vaginay et al., 2021]
  A. Vaginay, et al.
  From Quantitative SBML Models to Boolean Networks
  Complex Networks & Their Applications X 2021
- [Vaginay et al., 2022]
  A. Vaginay, et al.
  From Quantitative SBML Models to Boolean Networks
  Applied Network Science vol. 7-1 pp. 1–23, 2022

Our abstraction versus other abstractions

Reaction-thinking

Reaction network

Boolean network

Influence thinking

Our abstraction versus other abstractions

Reaction-thinking

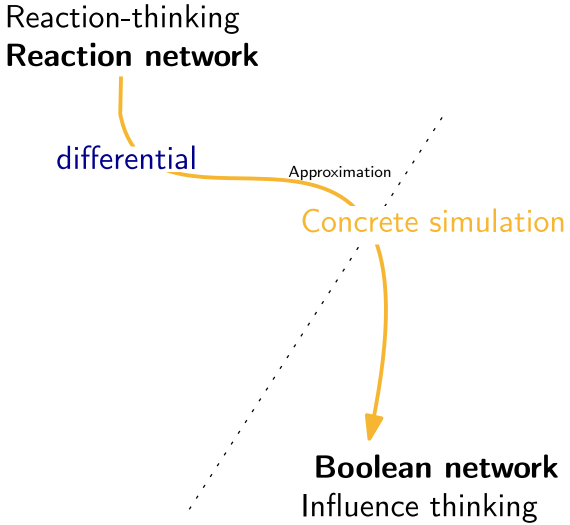
Reaction network

differential

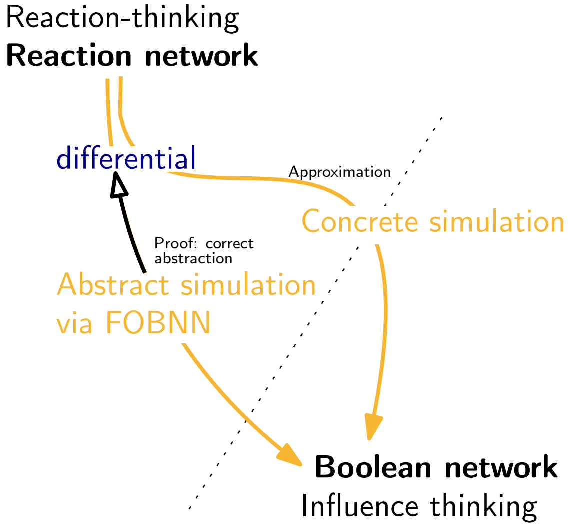
Boolean network

Influence thinking

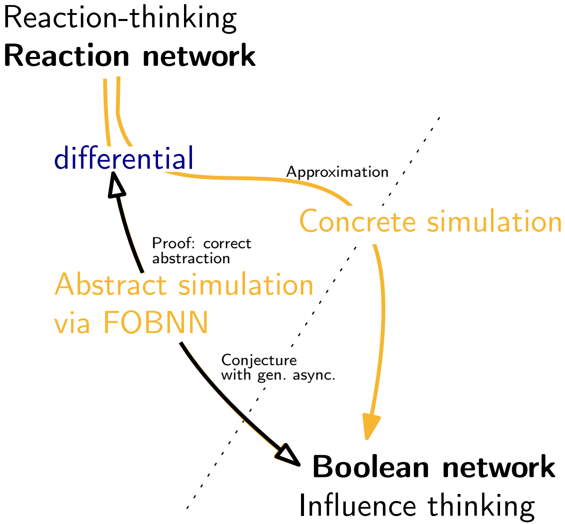
Our abstraction versus other abstractions



Our abstraction versus other abstractions



Our abstraction versus other abstractions



Our abstraction versus other abstractions

Reaction-thinking

Reaction network

stochastic

correct abstraction

discrete

correct abstraction

Boolean

differential

Proof: correct abstraction

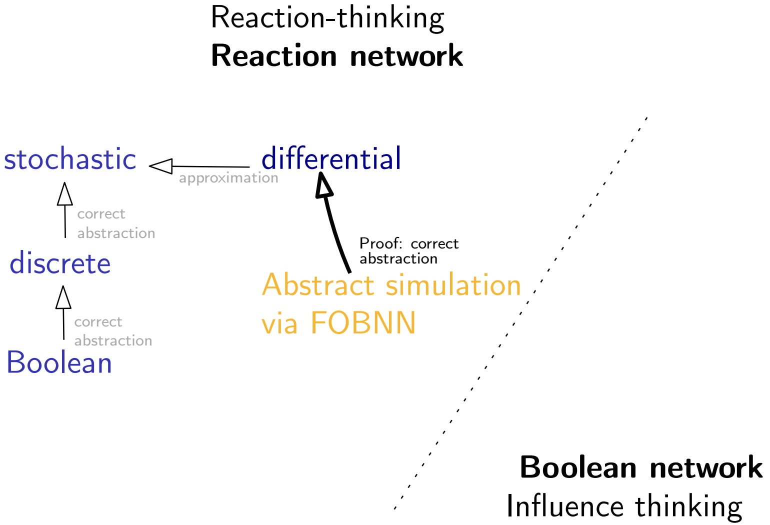
Abstract simulation via FOBNN

Boolean network

Influence thinking

[Fages, Soliman, 2008a]

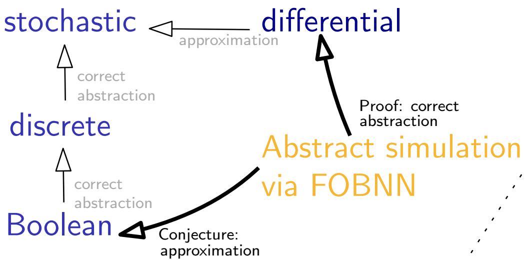
Our abstraction versus other abstractions



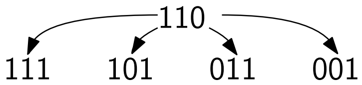
[Fages, Soliman, 2008a]

Our abstraction versus other abstractions

Reaction-thinking
Reaction network



$A + B \rightarrow C$



[Fages, Soliman, 2008a]

Boolean network
Influence thinking

ASK&D-BN— Local search

Candidate transition function

Search space: $2^{3^{|\mathcal{S}|}}$ non-redundant DNF = non-redundant disjunction of non-redundant conjunctions

ASK&D-BN— Local search

Candidate transition function

Search space: $2^{3^{|\mathcal{S}|}}$ non-redundant DNF = non-redundant disjunction of non-redundant conjunctions

Pick a subset of non-redundant conjunctions

```
% GIVEN : conj(ID, Component, Sign}
% conj(ID, Species, Sign}
conj(1, a,  1). conj(1, b,-1). conj(1, c,  0). %  A ∧ ¬B
conj(2, a, -1). conj(2, b, 0). conj(2, c, -1). % ¬A ∧ ¬C
conj(3, a, -1). conj(3, b,-1). conj(3, c, -1). % ¬A ∧ ¬B ∧ ¬C
...
1{conjTakenID(0..maxNbPossibleConj)}. % choice rule
```

ASK&D-BN— Local search

Candidate transition function

Search space: $2^{3^{|\mathcal{S}|}}$ non-redundant DNF = non-redundant disjunction of non-redundant conjunctions

Pick a subset of non-redundant conjunctions

```
% GIVEN : conj(ID, Component, Sign}
% conj(ID, Species, Sign}
conj(1, a,  1). conj(1, b,-1). conj(1, c,  0). %  A ∧ ¬B
conj(2, a, -1). conj(2, b, 0). conj(2, c, -1). % ¬A ∧ ¬C
conj(3, a, -1). conj(3, b,-1). conj(3, c, -1). % ¬A ∧ ¬B ∧ ¬C
...
1{conjTakenID(0..maxNbPossibleConj)}. % choice rule
```

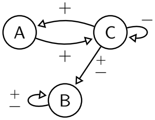
Example

`conjTakenID(1). conjTakenID(2).` ⇒ candidate = $(A \land \neg B) \lor (\neg A \land \neg C)$

ASK&D-BN— Local search

Structure constraints

influence graph of the Boolean network $\subseteq$ influence graph of the reaction network



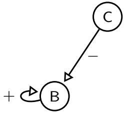
Do not select a conjunction that uses a forbidden literal

```
ig(ParentID, x, V) :- conjTaken(ConjID, ParentID, V); V!=0.
:- ig(ParentID, x, V) ; not pig(ParentID, x, V).
```

Example

invalid conjunction: $\neg A \wedge \neg C$

valid conjunction: $\neg C \wedge B$



ASK&D-BN— Local search

Dynamics constraints

(1) input: Boolean transitions

Build partial truth tables for each species X: what were the values of its putative inputs **when its value changed**? $\rightsquigarrow$ Do not assume the underlying update scheme
Compare the truth table of a candidate function to the reconstructed truth table

putative input	output

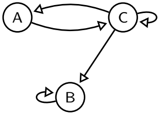
ASK&D-BN— Local search

Dynamics constraints

(1) input: Boolean transitions

Build partial truth tables for each species X: what were the values of its putative inputs **when its value changed**? $\rightsquigarrow$ Do not assume the underlying update scheme
Compare the truth table of a candidate function to the reconstructed truth table

input influence graph (unsigned)



putative input	output
C	A
BC	B
AC	C

ASK&D-BN— Local search

Dynamics constraints

(1) input: Boolean transitions

Build partial truth tables for each species X: what were the values of its putative inputs **when its value changed**? ⇝ Do not assume the underlying update scheme
Compare the truth table of a candidate function to the reconstructed truth table

$$010 \xrightarrow[1]{} 011 \xrightarrow[2]{} 100 \xrightarrow[3]{} 001$$

putative input	output
C	A
BC	B
AC	C

ASK&D-BN— Local search

Dynamics constraints

(1) input: Boolean transitions

Build partial truth tables for each species X: what were the values of its putative inputs **when its value changed**? $\rightsquigarrow$ Do not assume the underlying update scheme

Compare the truth table of a candidate function to the reconstructed truth table

$$010 \xrightarrow[1]{C} 011 \xrightarrow[2]{A,B,C} 100 \xrightarrow[3]{A,C} 001$$

putative input	output
C	A
BC	B
AC	C

ASK&D-BN— Local search

Dynamics constraints

(1) input: Boolean transitions

Build partial truth tables for each species X: what were the values of its putative inputs **when its value changed**? $\rightsquigarrow$ Do not assume the underlying update scheme
Compare the truth table of a candidate function to the reconstructed truth table

$$010 \xrightarrow[1]{C} 011 \xrightarrow[2]{A,B,C} 100 \xrightarrow[3]{A,C} 001$$

putative input	output	
C	A	
1	1	2
BC	B	
AC	C	

ASK&D-BN— Local search

Dynamics constraints

(1) input: Boolean transitions

Build partial truth tables for each species X: what were the values of its putative inputs **when its value changed**? $\rightsquigarrow$ Do not assume the underlying update scheme
Compare the truth table of a candidate function to the reconstructed truth table

$$010 \xrightarrow[1]{C} 011 \xrightarrow[2]{A,B,C} 100 \xrightarrow[3]{A,C} 001$$

putative input	output	
C	A	
0	0	3
1	1	2
BC	B	
AC	C	

ASK&D-BN— Local search

Dynamics constraints

(1) input: Boolean transitions

Build partial truth tables for each species X: what were the values of its putative inputs **when its value changed**? $\rightsquigarrow$ Do not assume the underlying update scheme
Compare the truth table of a candidate function to the reconstructed truth table

$$010 \xrightarrow[1]{C} 011 \xrightarrow[2]{A,B,C} 100 \xrightarrow[3]{A,C} 001$$

putative input	output	
C	A	
0	0	3
1	1	2
BC	B	
11	0	2
AC	C	
00	1	1
01	0	2
10	1	3

ASK&D-BN— Local search

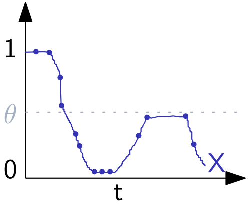
Dynamics constraints

(2) input: time series

```
#minimize{E@2 : error(E)}. %
```

X_t: continuous value of X at time t

θ: binarisation threshold for X



ASK&D-BN— Local search

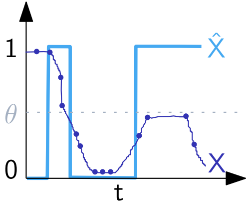
Dynamics constraints

(2) input: time series

```
#minimize{E@2 : error(E)}. %
```

X_t: continuous value of X at time t

θ: binarisation threshold for X



ASK&D-BN— Local search

Dynamics constraints

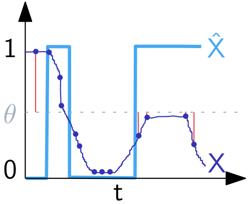
(2) input: time series

```
#minimize{E@2 : error(E)}. %
```

X_t: continuous value of X at time t

θ: binarisation threshold for X

$\mathcal{U}$: set of unexplained time steps



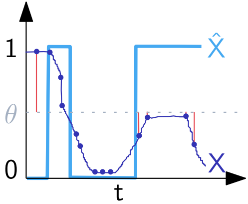
ASK&D-BN— Local search

Dynamics constraints

(2) input: time series

```
#minimize{E@2 : error(E)}. %
```

X_t: continuous value of X at time t
θ: binarisation threshold for X
$\mathcal{U}$: set of unexplained time steps
$E = \sum_{t \in \mathcal{U}} |\theta - X_t|$ To minimise (ideally 0)



ASK&D-BN— Local search

Minimality constraint

Select candidates with the smallest expressions (subset and/or cardinal minimal)
⇝ most general conditions

```
sizeconj(C, S):-conjTakenID(C);S=#sum{|V|,N:conj(C, N, V)} .
sizeDNF(S):- S=#sum{N,C: sizeconj(C, N), conjTakenID(C)} .
#minimize{S@1 : sizeDNF(S)}. % Find mincard expressions
% + generate all combinations to find all the subset min expressions
```

putative input	observed output
AB	X
00	
01	0
10	1
11	

ASK&D-BN— Local search

Minimality constraint

Select candidates with the smallest expressions (subset and/or cardinal minimal)
⇝ most general conditions

```
sizeconj(C, S):-conjTakenID(C);S=#sum{|V|,N:conj(C, N, V)} .
sizeDNF(S):- S=#sum{N,C: sizeconj(C, N), conjTakenID(C)} .
#minimize{S@1 : sizeDNF(S)}. % Find mincard expressions
% + generate all combinations to find all the subset min expressions
```

putative input	observed output	possible completions			
AB	X				
00		0	1	0	1
01	0	0	0	0	0
10	1	1	1	1	1
11		0	0	1	1

ASK&D-BN— Local search

Minimality constraint

Select candidates with the smallest expressions (subset and/or cardinal minimal)
⇝ most general conditions

```
sizeconj(C, S):-conjTakenID(C);S=#sum{|V|,N:conj(C, N, V)} .
sizeDNF(S):- S=#sum{N,C: sizeconj(C, N), conjTakenID(C)} .
#minimize{S@1 : sizeDNF(S)}. % Find mincard expressions
% + generate all combinations to find all the subset min expressions
```

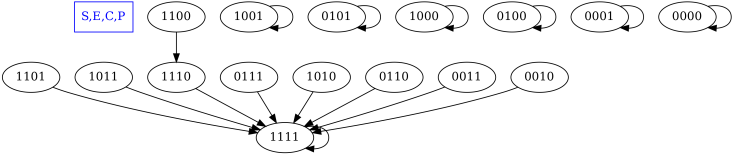
putative input AB	observed output X	possible completions			
00		0	1	0	1
01	0	0	0	0	0
10	1	1	1	1	1
11		0	0	1	1
	subset minimal candidates	$A \wedge \neg B$	$\neg B$	A	$A \vee \neg B$
	size	2	1	1	2

card. min. candidates

FOBNN fix-points with SAT

Given an FOBNN ϕ with variables $\mathcal{V} = \bigcup_{\mathsf{X} \in \mathcal{S}} \{\mathsf{X}, \mathring{\mathsf{X}}, \underset{\text{next}}{\mathsf{X}}, \underset{\text{next}}{\mathring{\mathsf{X}}}\}$, find the signed assignments α of ϕ such that:

$$\forall \mathsf{X} \in \mathcal{S} : \alpha(\mathsf{X}) = \alpha(\underset{\text{next}}{\mathsf{X}}) \text{ (and no others!)}$$



Hans-Jörg Schurr (Univ. of Iowa).

Functional dependency for detecting dynamics conflicts

Set of attributes $\mathcal{V}$ (relation scheme)
A set r of tuples that maps each attributes to a value of its domain ($t[X] \in dom(X)$)

A functional dependency (FD) F is an expression of the form $X \to Y$, where $X, Y \subseteq \mathcal{V}$
F holds in a relation r ($r \models f$) if:

$$\forall t_1, t_2 \in r, t_1[X] = t_2[X] \implies t_1[Y] = t_2[Y]$$

Find counterexamples when it does not hold (work on the conflict-graph).
Find the maximum (biggest) independent sets.

g3-error: minimal proportion of tuples to remove from r to satisfy $F \rightsquigarrow$ coverage measure
Simon Vilmin (AMU) and Pierre Faure–Giovagnoli (LIRIS): relax the equality by using a predicate p instead, study how the complexity of the problems depends on the properties of p (reflexivity, symetry, transitivity, antisymetry)

Functional dependency for detecting dynamics conflicts

Set of attributes $\mathcal{V}$ (relation scheme)
A set r of tuples that maps each attributes to a value of its domain ($t[X] \in dom(X)$)

A functional dependency (FD) is an expression of the form $X \rightarrow Y$, where $X, Y \subseteq \mathcal{V}$
f holds in a relation r ($r \models f$) if:

$$\forall t_1, t_2 \in r, t_1[X] = t_2[X] \implies t_1[Y] = t_2[Y]$$

Find counterexamples when it does not hold (work on the conflict-graph).
Find the maximum (biggest) independent sets.

g3-error: minimal proportion of tuples to remove from r to satisfy $f \rightsquigarrow$ coverage measure

Simon Vilmin (AMU) and Pierre Faure–Giovagnoli (LIRIS): relax the equality by using a predicate p instead, study how the complexity of the problems depends on the properties of p (reflexivity, symmetry, transitivity, antisymmetry)

Functional dependency for detecting dynamics conflicts

Set of variables $\mathcal{V} = \mathcal{S} \cup \underset{\text{next}}{\mathcal{S}}$ (relation scheme)

A set r of tuples that maps each attributes to a value of its domain ($t[X] \in dom(X)$)

A functional dependency (FD) is an expression of the form $X \rightarrow Y$, where $X, Y \subseteq \mathcal{V}$

f holds in a relation r ($r \models f$) if:

$$\forall t_1, t_2 \in r, t_1[X] = t_2[X] \implies t_1[Y] = t_2[Y]$$

Find counterexamples when it does not hold (work on the conflict-graph).

Find the maximum (biggest) independent sets.

g3-error: minimal proportion of tuples to remove from r to satisfy $f \rightsquigarrow$ coverage measure

Simon Vilmin (AMU) and Pierre Faure–Giovagnoli (LIRIS): relax the equality by using a predicate p instead, study how the complexity of the problems depends on the properties of p (reflexivity, symmetry, transitivity, antisymmetry)

Functional dependency for detecting dynamics conflicts

Set of variables $\mathcal{V} = \mathcal{S} \cup \underset{\text{next}}{\mathcal{S}}$ (relation scheme)

A set r of transitions that maps each attributes to a value of its domain ($t[X] \in dom(X) = \mathbb{B}^k$)

A functional dependency (FD) is an expression of the form $X \rightarrow Y$, where $X, Y \subseteq \mathcal{V}$
f holds in a relation r ($r \models f$) if:

$$\forall t_1, t_2 \in r, t_1[X] = t_2[X] \implies t_1[Y] = t_2[Y]$$

Find counterexamples when it does not hold (work on the conflict-graph).
Find the maximum (biggest) independent sets.

g3-error: minimal proportion of tuples to remove from r to satisfy $f \rightsquigarrow$ coverage measure

Simon Vilmin (AMU) and Pierre Faure–Giovagnoli (LIRIS): relax the equality by using a predicate p instead, study how the complexity of the problems depends on the properties of p (reflexivity, symmetry, transitivity, antisymmetry)

Functional dependency for detecting dynamics conflicts

Set of variables $\mathcal{V} = \mathcal{S} \cup \underset{\text{next}}{\mathcal{S}}$ (relation scheme)

A set r of transitions that maps each attributes to a value of its domain ($t[X] \in dom(X) = \mathbb{B}^k$)

r	A	B	C	$\underset{\text{next}}{\text{A}}$	$\underset{\text{next}}{\text{B}}$	$\underset{\text{next}}{\text{C}}$	$X \subseteq \mathcal{S}$
t_1	0	0	0	0	0	0	●
t_2	0	1	1	1	0	0	●
t_3	0	0	0	0	0	1	●

...

A functional dependency (FD) is an expression of the form $X \to Y$, where $X, Y \subseteq \mathcal{V}$
f holds in a relation r ($r \models f$) if:

$$\forall t_1, t_2 \in r, t_1[X] = t_2[X] \implies t_1[Y] = t_2[Y]$$

Find counterexamples when it does not hold (work on the conflict-graph).
Find the maximum (biggest) independent sets.

g3-error: minimal proportion of tuples to remove from r to satisfy $f \leadsto$ coverage measure

Simon Vilmin (AMU) and Pierre Faure–Giovagnoli (LIRIS): relax the equality by using a predicate p instead, study how the complexity of the problems depends on the properties of p (reflexivity, symmetry, transitivity, antisymmetry)

Functional dependency for detecting dynamics conflicts

Set of variables $\mathcal{V} = \mathcal{S} \cup \underset{\text{next}}{\mathcal{S}}$ (relation scheme)

A set r of transitions that maps each attributes to a value of its domain ($t[X] \in dom(X) = \mathbb{B}^k$)

r	A	B	C	$\underset{\text{next}}{\text{A}}$	$\underset{\text{next}}{\text{B}}$	$\underset{\text{next}}{\text{C}}$	$X \subseteq \mathcal{S}$
t_1	0	0	0	0	0	0	●
t_2	0	1	1	1	0	0	●
t_3	0	0	0	0	0	1	●

...

A functional dependency (FD) is an expression of the form $X \to Y$, where $X, Y \subseteq \mathcal{V} \rightsquigarrow$ a transition function f holds in a relation r ($r \models f$) if:

$$\forall t_1, t_2 \in r, t_1[X] = t_2[X] \implies t_1[Y] = t_2[Y]$$

Find counterexamples when it does not hold (work on the conflict-graph).
Find the maximum (biggest) independent sets.

g3-error: minimal proportion of tuples to remove from r to satisfy $f \rightsquigarrow$ coverage measure

Simon Vilmin (AMU) and Pierre Faure–Giovagnoli (LIRIS): relax the equality by using a predicate p instead, study how the complexity of the problems depends on the properties of p (reflexivity, symmetry, transitivity, antisymmetry)

Functional dependency for detecting dynamics conflicts

Set of variables $\mathcal{V} = \mathcal{S} \cup \underset{\text{next}}{\mathcal{S}}$ (relation scheme)

A set r of transitions that maps each attributes to a value of its domain ($t[X] \in dom(X) = \mathbb{B}^k$)

r	A	B	C	$\underset{\text{next}}{\text{A}}$	$\underset{\text{next}}{\text{B}}$	$\underset{\text{next}}{\text{C}}$	$X \subseteq \mathcal{S}$
t_1	0	0	0	0	0	0	●
t_2	0	1	1	1	0	0	●
t_3	0	0	0	0	0	1	●

...

A functional dependency (FD) is an expression of the form $X \rightarrow Y$, where $X, Y \subseteq \mathcal{V} \rightsquigarrow$ a transition function f holds in a relation r ($r \models f$) if:

$$\forall t_1, t_2 \in r, t_1[X] = t_2[X] \implies t_1[Y] = t_2[Y]$$

Find counterexamples when it does not hold (work on the conflict-graph).
Find the maximum (biggest) independent sets.

g3-error: minimal proportion of tuples to remove from r to satisfy $f \rightsquigarrow$ coverage measure

Simon Vilmin (AMU) and Pierre Faure–Giovagnoli (LIRIS): relax the equality by using a predicate p instead, study how the complexity of the problems depends on the properties of p (reflexivity, symmetry, transitivity, antisymmetry)

Functional dependency for detecting dynamics conflicts

Set of variables $\mathcal{V} = \mathcal{S} \cup \underset{\text{next}}{\mathcal{S}}$ (relation scheme)

A set r of transitions that maps each attributes to a value of its domain ($t[X] \in dom(X) = \mathbb{B}^k$)

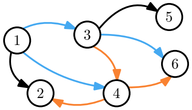
r	A	B	C	$\underset{\text{next}}{\text{A}}$	$\underset{\text{next}}{\text{B}}$	$\underset{\text{next}}{\text{C}}$	$X \subseteq \mathcal{S}$
t_1	0	0	0	0	0	0	●
t_2	0	1	1	1	0	0	●
t_3	0	0	0	0	0	1	●

. . .

A functional dependency (FD) is an expression of the form $X \to Y$, where $X, Y \subseteq \mathcal{V} \rightsquigarrow$ a transition function f holds in a relation r ($r \models f$) if:

$$\forall t_1, t_2 \in r, t_1[X] = t_2[X] \implies t_1[Y] = t_2[Y]$$

Find counterexamples when it does not hold (work on the conflict-graph).
Find the maximum (biggest) independent sets.



g3-error: minimal proportion of tuples to remove from r to satisfy $f \rightsquigarrow$ coverage measure

Simon Vilmin (AMU) and Pierre Faure–Giovagnoli (LIRIS): relax the equality by using a predicate p instead, study how the complexity of the problems depends on the properties of p (reflexivity, symmetry, transitivity, antisymmetry)

Functional dependency for detecting dynamics conflicts

Set of variables $\mathcal{V} = \mathcal{S} \cup \underset{\text{next}}{\mathcal{S}}$ (relation scheme)

A set r of transitions that maps each attributes to a value of its domain ($t[X] \in dom(X) = \mathbb{B}^k$)

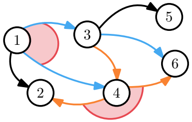
r	A	B	C	$\underset{\text{next}}{\text{A}}$	$\underset{\text{next}}{\text{B}}$	$\underset{\text{next}}{\text{C}}$	$X \subseteq \mathcal{S}$
t_1	0	0	0	0	0	0	●
t_2	0	1	1	1	0	0	●
t_3	0	0	0	0	0	1	●

...

A functional dependency (FD) is an expression of the form $X \to Y$, where $X, Y \subseteq \mathcal{V} \rightsquigarrow$ a transition function f holds in a relation r ($r \models f$) if:

$$\forall t_1, t_2 \in r, t_1[X] = t_2[X] \implies t_1[Y] = t_2[Y]$$

Find counterexamples when it does not hold (work on the conflict-graph).
Find the maximum (biggest) independent sets.



g3-error: minimal proportion of tuples to remove from r to satisfy $f \rightsquigarrow$ coverage measure

Simon Vilmin (AMU) and Pierre Faure–Giovagnoli (LIRIS): relax the equality by using a predicate p instead, study how the complexity of the problems depends on the properties of p (reflexivity, symmetry, transitivity, antisymmetry)

Functional dependency for detecting dynamics conflicts

Set of variables $\mathcal{V} = \mathcal{S} \cup \underset{\text{next}}{\mathcal{S}}$ (relation scheme)

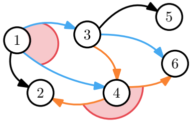
A set r of transitions that maps each attributes to a value of its domain ($t[X] \in dom(X) = \mathbb{B}^k$)

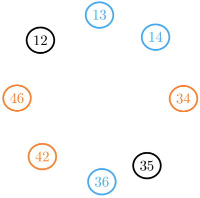
r	A	B	C	$\underset{\text{next}}{\text{A}}$	$\underset{\text{next}}{\text{B}}$	$\underset{\text{next}}{\text{C}}$	$X \subseteq \mathcal{S}$
t_1	0	0	0	0	0	0	●
t_2	0	1	1	1	0	0	●
t_3	0	0	0	0	0	1	●
				...			

A functional dependency (FD) is an expression of the form $X \to Y$, where $X, Y \subseteq \mathcal{V} \rightsquigarrow$ a transition function f holds in a relation r ($r \models f$) if:

$$\forall t_1, t_2 \in r, t_1[X] = t_2[X] \implies t_1[Y] = t_2[Y]$$

Find counterexamples when it does not hold (work on the conflict-graph).
Find the maximum (biggest) independent sets.





g3-error: minimal proportion of tuples to remove from r to satisfy $f \rightsquigarrow$ coverage measure

Simon Vilmin (AMU) and Pierre Faure–Giovagnoli (LIRIS): relax the equality by using a predicate p instead, study how the complexity of the problems depends on the properties of p (reflexivity, symmetry, transitivity, antisymmetry)

Functional dependency for detecting dynamics conflicts

Set of variables $\mathcal{V} = \mathcal{S} \cup \underset{\text{next}}{\mathcal{S}}$ (relation scheme)

A set r of transitions that maps each attributes to a value of its domain ($t[X] \in dom(X) = \mathbb{B}^k$)

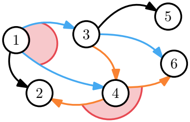
r	A	B	C	$\underset{\text{next}}{\text{A}}$	$\underset{\text{next}}{\text{B}}$	$\underset{\text{next}}{\text{C}}$	$X \subseteq \mathcal{S}$
t_1	0	0	0	0	0	0	●
t_2	0	1	1	1	0	0	●
t_3	0	0	0	0	0	1	●

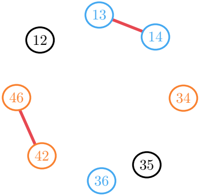
...

A functional dependency (FD) is an expression of the form $X \to Y$, where $X, Y \subseteq \mathcal{V} \rightsquigarrow$ a transition function f holds in a relation r ($r \models f$) if:

$$\forall t_1, t_2 \in r, t_1[X] = t_2[X] \implies t_1[Y] = t_2[Y]$$

Find counterexamples when it does not hold (work on the conflict-graph).
Find the maximum (biggest) independent sets.





g3-error: minimal proportion of tuples to remove from r to satisfy $f \rightsquigarrow$ coverage measure

Simon Vilmin (AMU) and Pierre Faure–Giovagnoli (LIRIS): relax the equality by using a predicate p instead, study how the complexity of the problems depends on the properties of p (reflexivity, symmetry, transitivity, antisymmetry)

Functional dependency for detecting dynamics conflicts

Set of variables $\mathcal{V} = \mathcal{S} \cup \underset{\text{next}}{\mathcal{S}}$ (relation scheme)

A set r of transitions that maps each attributes to a value of its domain ($t[X] \in dom(X) = \mathbb{B}^k$)

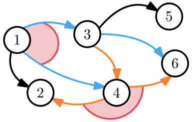
r	A	B	C	$\underset{\text{next}}{\text{A}}$	$\underset{\text{next}}{\text{B}}$	$\underset{\text{next}}{\text{C}}$	$X \subseteq \mathcal{S}$
t_1	0	0	0	0	0	0	●
t_2	0	1	1	1	0	0	●
t_3	0	0	0	0	0	1	●

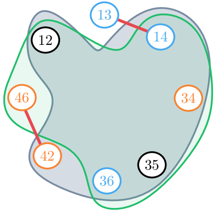
...

A functional dependency (FD) is an expression of the form $X \rightarrow Y$, where $X, Y \subseteq \mathcal{V} \rightsquigarrow$ a transition function f holds in a relation r ($r \models f$) if:

$$\forall t_1, t_2 \in r, t_1[X] = t_2[X] \implies t_1[Y] = t_2[Y]$$

Find counterexamples when it does not hold (work on the conflict-graph).
Find the maximum (biggest) independent sets.





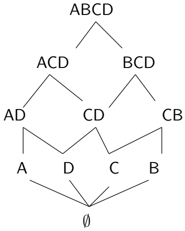
g3-error: minimal proportion of tuples to remove from r to satisfy $f \rightsquigarrow$ coverage measure

Simon Vilmin (AMU) and Pierre Faure–Giovagnoli (LIRIS): relax the equality by using a predicate p instead, study how the complexity of the problems depends on the properties of p (reflexivity, symmetry, transitivity, antisymmetry)

Learn reaction networks from Boolean transitions

Implication base with variables in $\mathcal{S}$: $\boldsymbol{\mathcal{R}} = \{R_i \to P_i\}_{i=1\dots m}$
Closed-set: "element of $\mathcal{P}(\mathcal{S})$ such that we cannot derive anything new using $\boldsymbol{\mathcal{R}}$"
Closure system = the set $\mathcal{C}$ of closed-sets of $\boldsymbol{\mathcal{R}}$
$\mathcal{C}$ ordered by $\subseteq \rightsquigarrow$ a lattice

$$\begin{aligned}
&\boldsymbol{\mathcal{R}} = \{\\
&\mathcal{R}_1 : \mathrm{A} + \mathrm{B} \to \mathrm{C} + \mathrm{D}\\
&\mathcal{R}_2 : \mathrm{A} + \mathrm{C} \to \mathrm{D}\\
&\mathcal{R}_3 : \mathrm{B} + \mathrm{D} \to \mathrm{C}\\
&\}
\end{aligned}$$

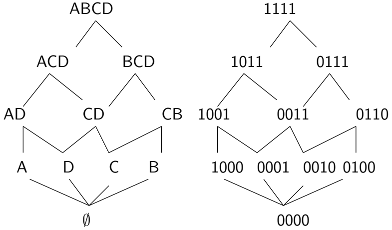


Simon Vilmin (AMU), Loïc Paulevé? (LABRI)

Learn reaction networks from Boolean transitions

Reaction network with species in $\mathcal{S}$: $\boldsymbol{\mathcal{R}} = \{R_i \to P_i\}_{i=1\ldots m}$
Closed-set: "element of $\mathcal{P}(\mathcal{S})$ such that we cannot derive anything new using $\boldsymbol{\mathcal{R}}$"
Closure system $=$ the set $\mathcal{C}$ of closed-sets of $\boldsymbol{\mathcal{R}}$
$\mathcal{C}$ ordered by $\subseteq \rightsquigarrow$ a lattice

$$\begin{aligned} \boldsymbol{\mathcal{R}} = \{ & \\ \mathcal{R}_1 :& \mathrm{A} + \mathrm{B} \to \mathrm{C} + \mathrm{D} \\ \mathcal{R}_2 :& \mathrm{A} + \mathrm{C} \to \mathrm{D} \\ \mathcal{R}_3 :& \mathrm{B} + \mathrm{D} \to \mathrm{C} \\ \} & \end{aligned}$$

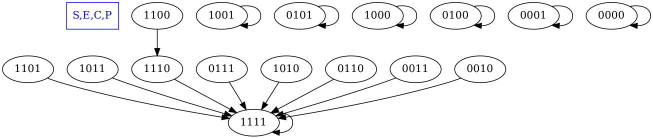


given a closure system, find the implication base(s)
$\stackrel{?}{=}$
given Boolean fixed-points, find the reaction network(s)

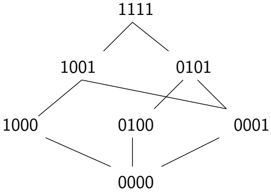
Simon Vilmin (AMU), Loïc Paulevé? (LABRI)

Learn reaction networks from Boolean transitions

input state graph:



derived closure system:



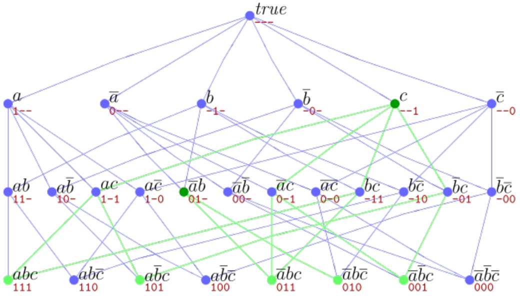
derived implication:

E + S → P ✓

Minimal DNF

Given a set S of inputs for which a function f eval. to 1, each minimal-by-inclusion set of nodes that covers exactly S forms a (subset-)minimal DNF of f.
f might have several (subset-)minimal DNFs.

Example: $S = \{abc, a\bar{b}c, \bar{a}bc, \bar{a}b\bar{c}, \bar{a}\bar{b}c\}$ (light green) $\rightsquigarrow$ $\{\bar{a}b, c\}$ (dark green)



Not well-formed reaction networks

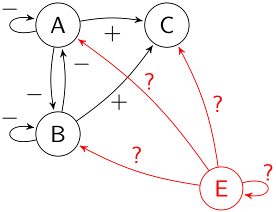
$$\mathsf{X} \xrightarrow{k \times \mathsf{Y}} \_$$

$\frac{\partial \mathsf{X}}{\partial \mathsf{Y}} \neq 0$ NOT captured by the syntactic influence graph.

Impact of SBML inconsistencies on structure extraction

Ex. BIOMD n°44: 1 BN generated; coverage=0.55
some kinetics use components not listed in the reactants nor modifiers → incomplete SIG (missing parents)

$$A + B \xrightarrow{f(A,B,\textcolor{red}{E})} C$$

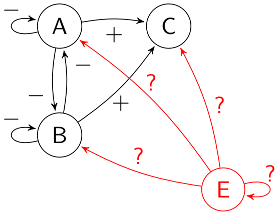


[4] [Fages et al. 2012]

Impact of SBML inconsistencies on structure extraction

Ex. BIOMD n°44: 1 BN generated; coverage=0.55
some kinetics use components not listed in the reactants nor modifiers → incomplete SIG (missing parents)

$$A + B \xrightarrow{f(A,B,E)} C$$



> 60% of SBML models from Biomodels are not "well-formed"[4], but some can be fixed → add a step in the pipeline

[4] [Fages et al. 2012]

Results to real-world reaction networks (from BioModels[5])

Input: an **extended** reaction network rules and events
Output: a set of compatible Boolean networks, according to `ASK&D-BN`

Setting:

- hard structure constraint (**extended** influence graph)
- soft dynamics constraints (time series and midrange binarisation)
- mincard DNF

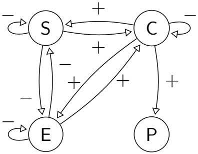
Result:

- on 155 reaction networks processed in less than 30 hours
- we synthesise perfect Boolean networks for ∼90% of them ✓
  139/155 sets of BNs have a coverage proportion median = 1

[5] [Malik-Sheriff et al., 2020]

Comparison of two settings on $\mathcal{R}_{\mathrm{enz}}$

$$\underbrace{\overbrace{\mathsf{E} + \mathsf{S} \underset{e_{\mathrm{off}}}{\overset{e_{\mathrm{on}}}{\rightleftarrows}} \mathsf{C}}^{\mathcal{R}_{\mathrm{on}}}}_{\mathcal{R}_{\mathrm{off}}} \overbrace{\xrightarrow{e_{\mathrm{cat}}} \mathsf{E} + 2 \times \mathsf{P}}^{\mathcal{R}_{\mathrm{cat}}}$$



- influence graph
- time series
- binarised time series

midrange (0.8) and median (0.6):

$$f_{\mathsf{S}} := \neg \mathsf{E}$$
$$f_{\mathsf{E}} := \neg \mathsf{S}$$
$$f_{\mathsf{C}} := \mathsf{S}$$
$$f_{\mathsf{P}} := \mathsf{C}$$

$\rightarrow$ Coverage depends on the binarisation procedure, BNs miss some influences

- full graph from abstract simulation

$$f_{\mathsf{S}} := \mathsf{C} \vee \mathsf{S}$$
$$f_{\mathsf{E}} := \mathsf{E} \vee \mathsf{C}$$
$$f_{\mathsf{C}} := (\mathsf{E} \wedge \mathsf{S}) \vee \mathsf{C}$$
$$f_{\mathsf{P}} := \mathsf{C} \vee \mathsf{P}$$

$\rightarrow$ Perfect coverage, but does not comply with the influence graph

$\Longrightarrow$ They do not capture the same thing