# Synthesis of Boolean Networks from the Structure and Dynamics of Reaction Networks

Athénaïs Vaginay

14th November 2023

# My curriculum

- 2011–2012: medical studies
- 2012–2015: bachelor biology
- 2015–2017: master bioinformatics

  } Univ. Diderot, Paris
- 2018: engineer bioinformatics CRIStAL, Lille
  machine learning for gene expression analysis
- 2018–2023: PhD Loria / Cran / Univ. Lorraine, Nancy
  *Synthesis of Boolean networks from the structure and dynamics of reaction networks*
  Taha Boukhobza & Malika Smaïl-Tabbonne
- beginning 2024: visiting univ. Iowa, US

# Systems Biology

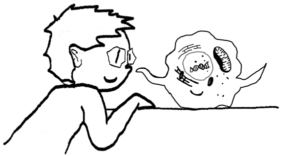
Formal modelling and reasoning about biological systems

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

**Questions**

**How does the system evolve?**

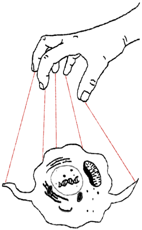
Is the population of some cell type stable over time?



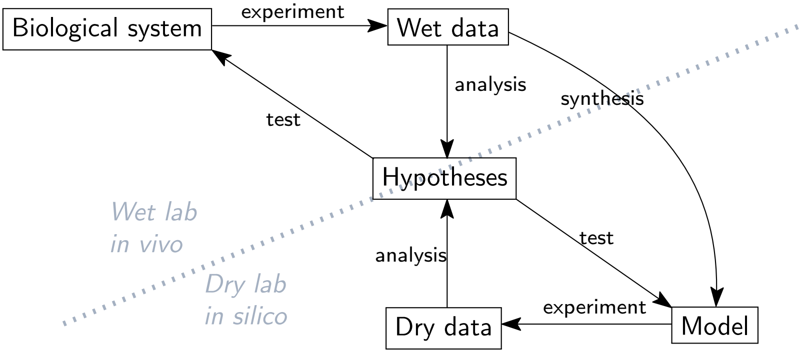
**How to control the system?**

Cure a pathological system

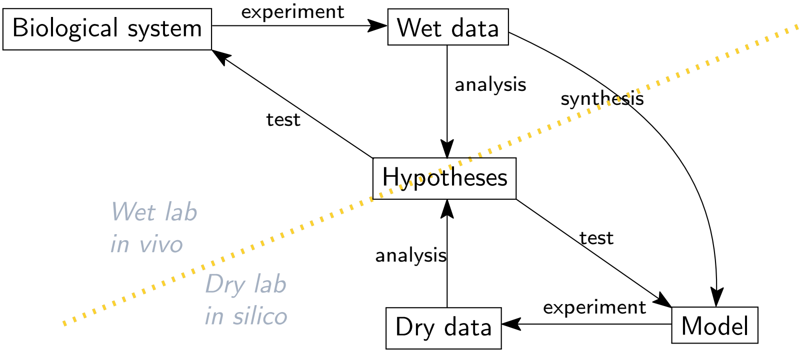
Produce more of some species of interest



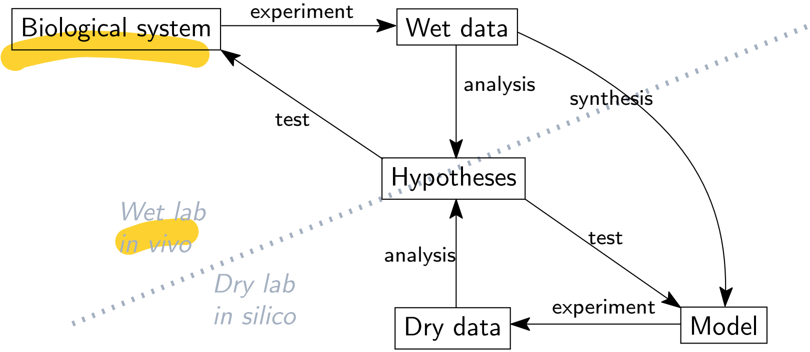
# The workflow of system biology [Kohl et al., 2010]



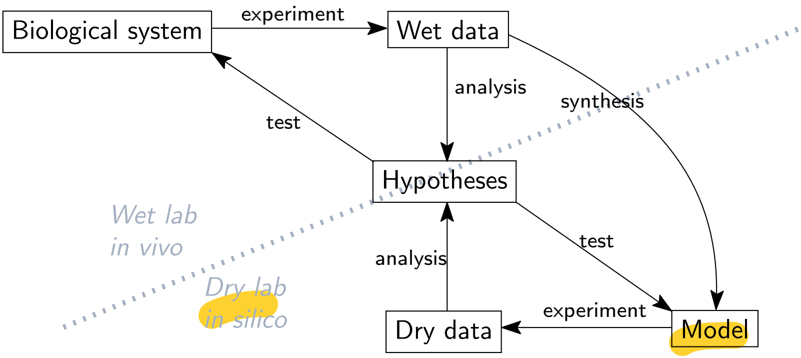
# The workflow of system biology [Kohl et al., 2010]



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



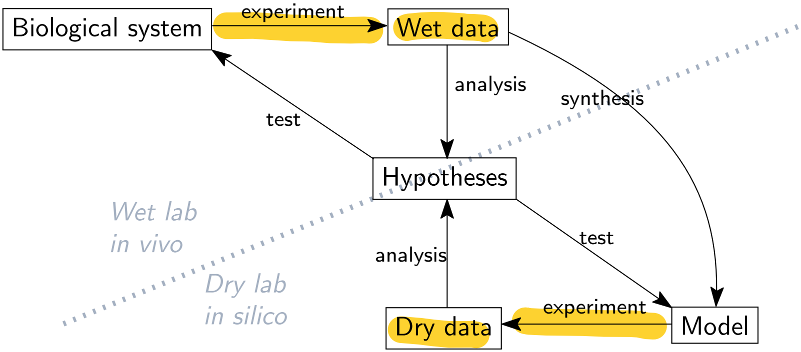
# The workflow of system biology [Kohl et al., 2010]




## Definition (Model)

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

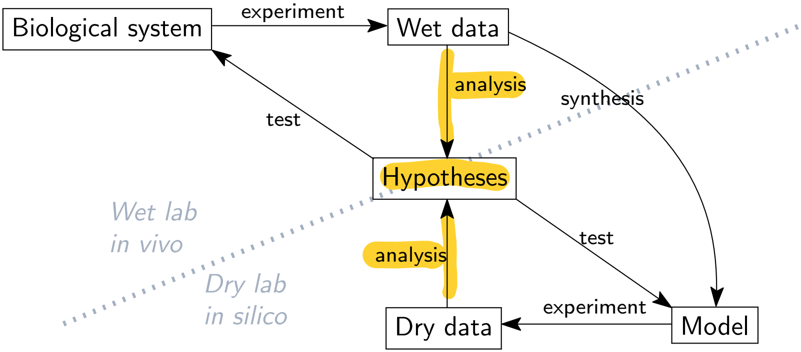
# The workflow of system biology [Kohl et al., 2010]




## Definition (Model)

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

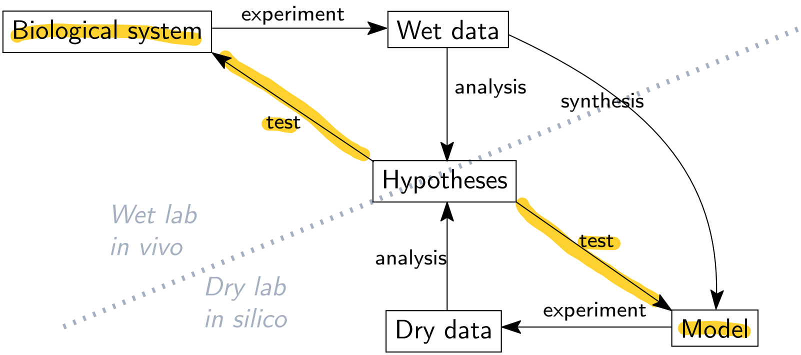
# The workflow of system biology [Kohl et al., 2010]




## Definition (Model)

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

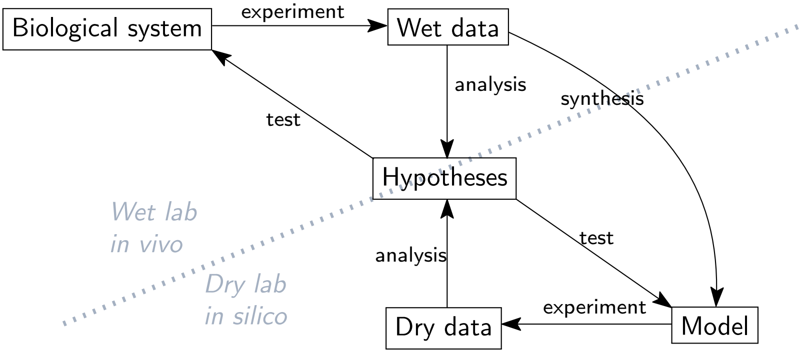
# The workflow of system biology [Kohl et al., 2010]




## Definition (Model)

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

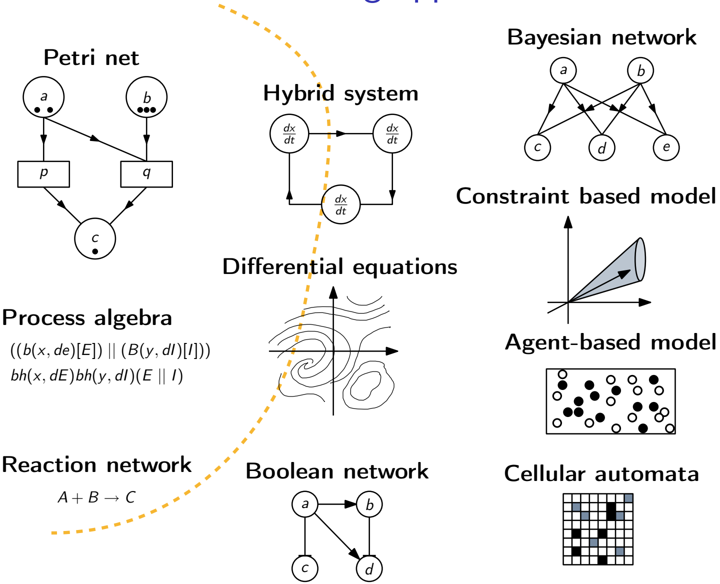
# The workflow of system biology [Kohl et al., 2010]




## Definition (Model)

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

# A dichotomic zoo of modelling approaches



# 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

# 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

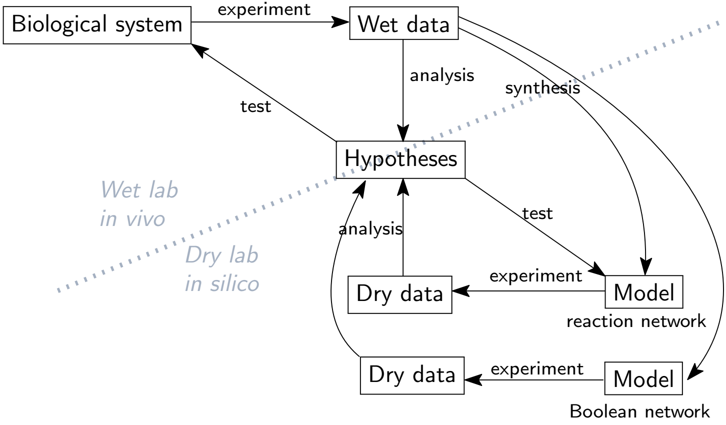
Use the simplest model that contains enough information to answer the question at hand. [Bornholdt, 2005]
Boolean networks are simpler than reaction networks.

**Problem statement**

*Automatic* transformation (abstraction) of reaction networks to Boolean networks

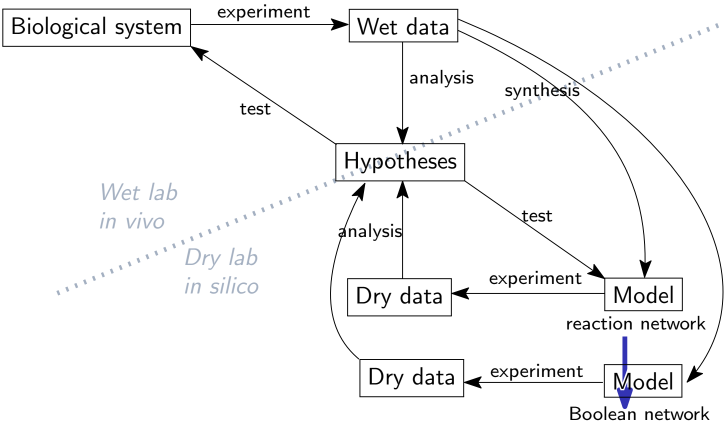
# From reactions to Boolean influences with guarantees

## Why?



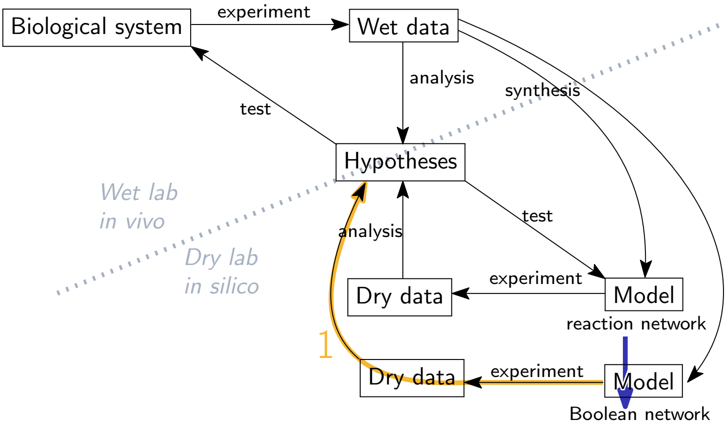
# From reactions to Boolean influences with guarantees

## Why?



# From reactions to Boolean influences with guarantees

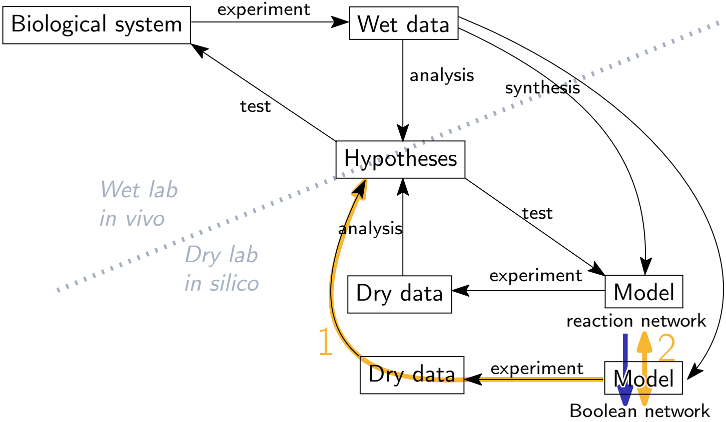
## Why?




1. Use BNs to facilitate some analyses

# From reactions to Boolean influences with guarantees

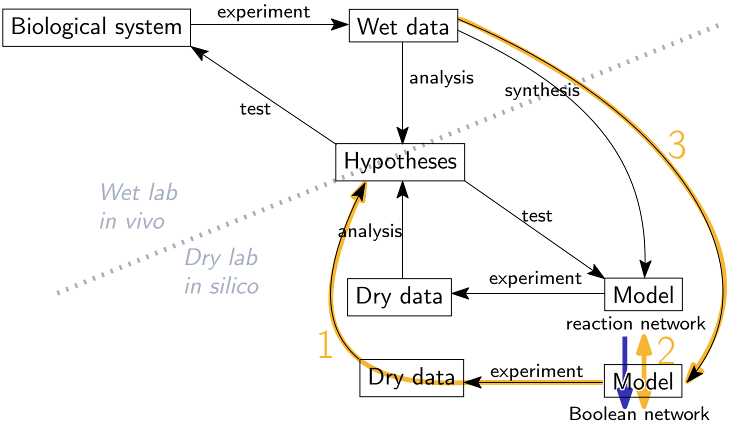
## Why?




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
3. Evaluation of the approach
4. Link to other abstractions
5. Conclusion and perspectives

# Outline

# 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

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

reaction, reactants, products, kinetics

**Example**

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

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

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

# 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

**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

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

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

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

# 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

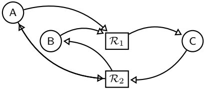
Reaction graph
$(\mathcal{S} \cup \mathcal{R}, E \subseteq (\mathcal{S} \times \mathcal{R}) \cup (\mathcal{R} \times \mathcal{S}))$

Example

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

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

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



# 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

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

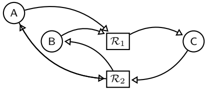
**Differential semantics**
ordinary differential equation (ODE)
$\left\{\dot{X} = \sum_{i \in 1\ldots m} e_i \times (P_i(X) - R_i(X))\right\}_{X \in \mathcal{S}}$

**Example**

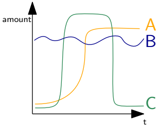
$\mathcal{S} = \{A, B, C\}$

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

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




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



# Boolean network, structure and dynamics

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

**Example**

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

$$f_A := 0$$
$$f_B := (B \wedge \neg C) \vee (\neg B \wedge C)$$
$$f_C := \neg C$$

# Boolean network, structure and dynamics

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

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

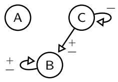
**Example**

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

$f_A := 0$

$f_B := (B \land \neg C) \lor (\neg B \land C)$

$f_C := \neg C$



# Boolean network, structure and dynamics

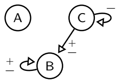
One transition function per species in $\mathcal{S}$:
$\left\{f_X : \mathbb{B}^{|\mathcal{S}|} \to \mathbb{B}\right\}_{X \in \mathcal{S}}$ $\qquad \mathbb{B} = \{0, 1\}$

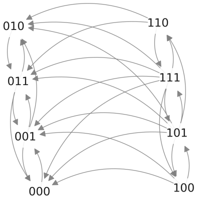
Influence graph
$IG = (\mathcal{S}, E \subseteq \mathcal{S} \times \mathcal{S}, \sigma : E \to \{+, -, \pm\})$

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

$$\mathcal{S} = \{A, B, C\}$$
$f_A := 0$
$f_B := (B \wedge \neg C) \vee (\neg B \wedge C)$
$f_C := \neg C$



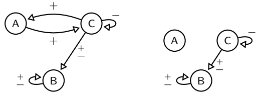



# From RN to BN with guarantees

## Which ones?

**Structure guaranty:** conserve direct influences among species

IG of input RN $\supseteq$ IG of output BN

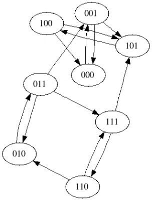



**Dynamics guaranty:** recover the Boolean transitions

Boolean transitions from input RN $\subseteq$ gen. async. TG of output BN

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

010 $\rightarrow$ 011 $\rightarrow$ 111 $\rightarrow$ 110

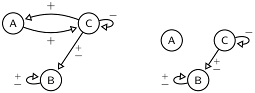


# From RN to BN with guarantees

## Which ones?

**Structure guaranty:** conserve direct influences among species

IG of input RN $\supseteq$ IG of output BN

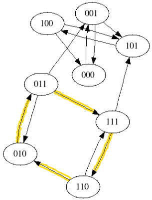



**Dynamics guaranty:** recover the Boolean transitions

Boolean transitions from input RN $\subseteq$ gen. async. TG of output BN

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

010 → 011 → 111 → 110

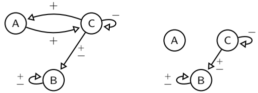


# From RN to BN with guarantees

## Which ones?

**Structure guaranty:** conserve direct influences among species

IG of input RN $\supseteq$ IG of output BN

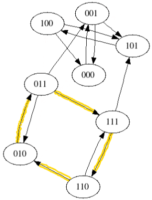



**Dynamics guaranty:** recover the Boolean transitions

Boolean transitions from input RN $\subseteq$ gen. async. TG of output BN

$$\left\{\dot{X} = \sum_{i \in 1 \ldots m} e_i \times (R_i(X) - P_i(X))\right\}_{X \in \mathcal{S}}$$

010 → 011 → 111 → 110




Coverage: proportion of recovered transitions (ideally 100%)

# Outline

1. Preliminaries on reaction networks and Boolean networks

2. My method and its guarantees

SBML2BNET

STEP 1: Retrieve from the input reaction network
**Structure**: influence graph
**Dynamics**: Boolean transitions

- binarised time series from classic simulation of the ODEs
- 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. Link to other abstractions

5. Conclusion and perspectives

# SBML2BNET – STEP 1: Retrieve an influence graph and Boolean transitions

# Running example $\mathcal{R}_{\text{enz}}$

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

Its ODEs (reconstructed)

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

Its parameters (given)

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

# Retrieve the influence graph of a reaction network

**Contribution**

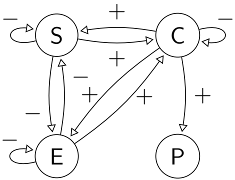
Implement the routines from [Fages, Soliman, 2008b]

"If Y is a reactant and X disapears: $Y \xrightarrow{-} X$"

**Guarantees**

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

Influence graph of $\mathcal{R}_{enz}$



# Retrieve Boolean transitions from a reaction network

## Numerical simulation and binarisation

**Contribution**

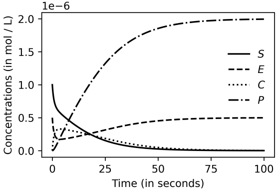
Use dedicated tools for simulation
Apply binarisation procedure

**Guarantees**

Approximate the real solution of the ODE with good accuracy [Hoops et al., 2006]
but causations are lost

For $\mathcal{R}_{\text{enz}}$:

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




Expected transitions: $1100 \rightarrow **10 \rightarrow ***1$

| Binarisation | Boolean configuration sequence SECP |
|---|---|
| Midrange | $1100 \rightarrow 1000 \rightarrow 1010 \rightarrow 0010 \rightarrow 0011 \rightarrow 0101$ |
| Median | $1100 \rightarrow 1010 \rightarrow 0011 \rightarrow 0101$ |
| Mean | $1100 \rightarrow 1010 \rightarrow 1000 \rightarrow 0011 \rightarrow 0101$ |
| Above 0 | $1100 \rightarrow 1111 \rightarrow 1011 \rightarrow 1111 \rightarrow 0111$ |

# Retrieve Boolean transitions from a reaction network

## Numerical simulation and binarisation

**Contribution**

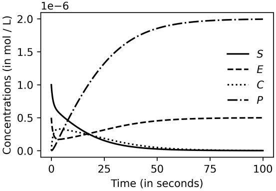
Use dedicated tools for simulation
Apply binarisation procedure

**Guarantees**

Approximate the real solution of the ODE with good accuracy [Hoops et al., 2006]
but causations are lost

For $\mathcal{R}_{\text{enz}}$:

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




Expected transitions: 1100 $\rightarrow **10 \rightarrow ***1$

| 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 |

# Retrieve Boolean transitions from a reaction network

## Numerical simulation and binarisation

**Contribution**

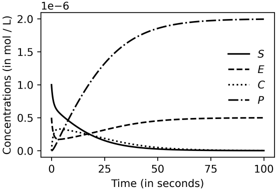
Use dedicated tools for simulation
Apply binarisation procedure

**Guarantees**

Approximate the real solution of the ODE with good accuracy [Hoops et al., 2006]
but causations are lost

For $\mathcal{R}_{\text{enz}}$:

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




Expected transitions: $1100 \rightarrow **10 \rightarrow ***1$

| Binarisation | Boolean configuration sequence SECP |
|---|---|
| Midrange | $1100 \rightarrow 1000 \rightarrow 1010 \rightarrow 0010 \rightarrow 0011 \rightarrow 0101$ |
| Median | $1100 \rightarrow 1010 \rightarrow 0011 \rightarrow 0101$ |
| Mean | $1100 \rightarrow 1010 \rightarrow 1000 \rightarrow 0011 \rightarrow 0101$ |
| Above 0 | $1100 \rightarrow 1111 \rightarrow 1011 \rightarrow 1111 \rightarrow 0111$ |

# Retrieve Boolean transitions from a reaction network

## Numerical simulation and binarisation

**Contribution**

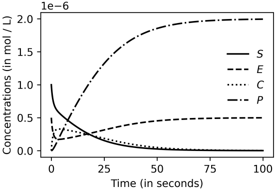
Use dedicated tools for simulation
Apply binarisation procedure

**Guarantees**

Approximate the real solution of the ODE with good accuracy [Hoops et al., 2006]
but causations are lost

For $\mathcal{R}_{\text{enz}}$:

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




Expected transitions: $1100 \rightarrow **10 \rightarrow ***1$

| 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 |

# Retrieve Boolean transitions from a reaction network

## Abstract simulation — Intuition

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

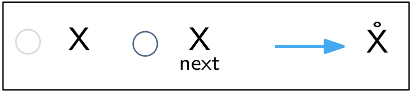
Use the rule of signs to reason on the causal relationship between the signs ($\mathbb{S} = \{-1, 0, 1\}$) of the variables values (species amount and derivatives) of the ODE system

# Retrieve Boolean transitions from a reaction network

## 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 (species amount and derivatives) of the ODE system

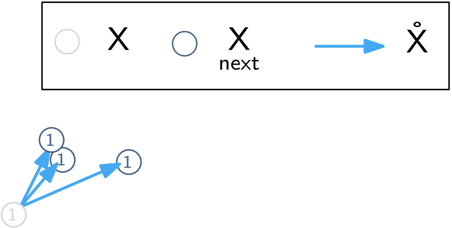


# Retrieve Boolean transitions from a reaction network

## 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 (species amount and derivatives) of the ODE system

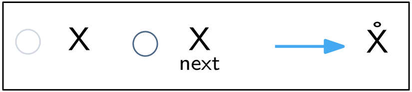


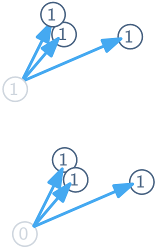
# Retrieve Boolean transitions from a reaction network

## 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 (species amount and derivatives) of the ODE system



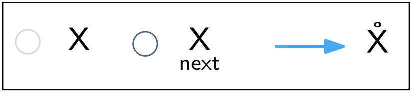



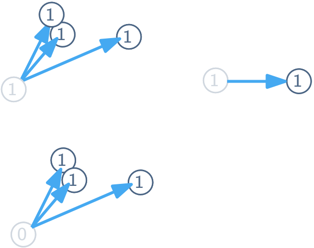
# Retrieve Boolean transitions from a reaction network

## 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 (species amount and derivatives) of the ODE system



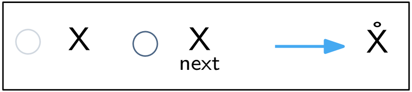



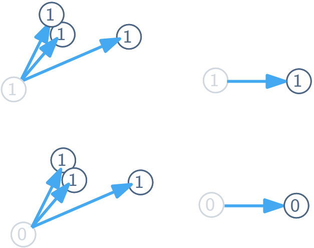
# Retrieve Boolean transitions from a reaction network

## 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 (species amount and derivatives) of the ODE system



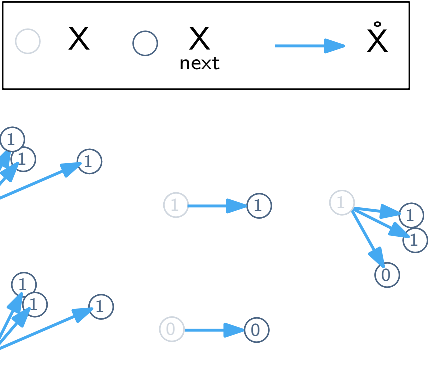



# Retrieve Boolean transitions from a reaction network

## 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 (species amount and derivatives) of the ODE system




X was above 0 and its derivative was negative

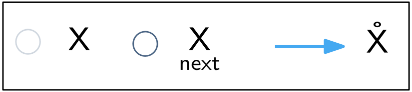
*plus* − *plus* = *unknown* ⇝ nondeterminism

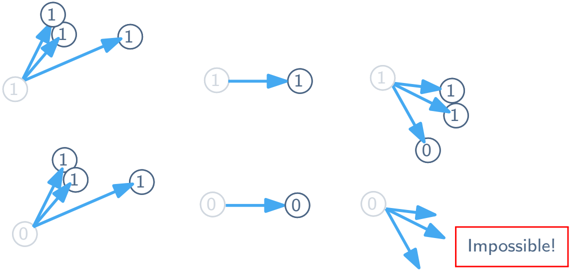
# Retrieve Boolean transitions from a reaction network

## 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 (species amount and derivatives) of the ODE system




X was above 0 and its derivative was negative

*plus* − *plus* = *unknown* ⇝ nondeterminism

# Retrieve Boolean transitions from a reaction network

## Abstract simulation — In practice

**Contribution**

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

- Causal relationships encoded by a first-order logic formula $\phi$
- Solve $\phi$ on the structure of signs $\mathbb{S} = \{-1, 0, 1\}$
- Restrict the solutions on $\mathcal{S} \cup \underset{\text{next}}{\mathcal{S}}$
  $\rightsquigarrow$ relation $\mathbb{B}^{|\mathcal{S}|} \times \mathbb{B}^{|\underset{\text{next}}{\mathcal{S}}|}$

**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

# Retrieve Boolean transitions from a reaction network

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

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

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

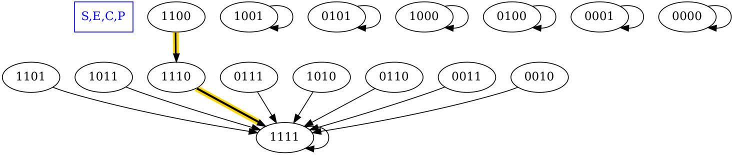
with

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

# Retrieve Boolean transitions from a reaction network

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

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



# Outline

1. Preliminaries on reaction networks and Boolean networks
2. My method and its guarantees

**SBML2BNET**

STEP 1: Retrieve from the input reaction network

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

- binarised time series from classic simulation of the ODEs
- 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. Link to other abstractions
5. Conclusion and perspectives

# SBML2BNET – STEP 2: Boolean network synthesis with ASK&D-BN

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

**Input**

**Structure**

Influence graph

**Dynamics**

Time series / Boolean time series

List of Boolean transitions

**Output**

Set of *compatible* Boolean networks

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

| **Input** | **Output** |
| --- | --- |
| **Structure**<br>Influence graph<br>**Dynamics**<br>Time series / Boolean time series<br>List of Boolean transitions | Set of *compatible*<br>Boolean networks |

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]

**Input**

**Structure**
Influence graph
**Dynamics**
Time series / Boolean time series
List of Boolean transitions

**Output**

Set of *compatible* Boolean networks

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

# ASK&D-BN— Local search

Generate candidates → Structure constraint → Dynamic constraint → Minimality constraint

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

Pick a subset of non-redundant conjunctions

# ASK&D-BN— Local search

Generate candidates → Structure constraint → Dynamic constraint → Minimality constraint

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

Pick a subset of non-redundant conjunctions

## Examples

invalid candidates:

$(A \wedge \neg B) \vee (A \wedge \neg B) \vee (\neg A \wedge \neg C)$

$(A \wedge A \wedge \neg B) \vee (\neg A \wedge \neg C)$

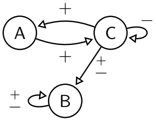
valid candidate:

$(A \wedge \neg B) \vee (\neg A \wedge \neg C)$

# ASK&D-BN— Local search

Generate candidates → Structure constraint → Dynamic constraint → Minimality constraint

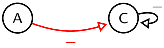
influence graph of the Boolean network ⊆ influence graph of the reaction network



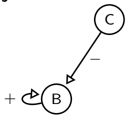

## Do not select a conjunction that uses a forbidden literal

### Example

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 |
| --- | --- |

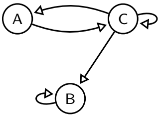
# ASK&D-BN— Local search

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

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

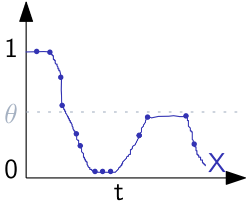
| putative input | output | |
|---|---|---|
| C | A | |
| 0 | 0 | (3) |
| 1 | 1 | (2) |
| BC | B | |
| 11 | 0 | (2) |
| AC | C | |
| 00 | 1 | (1) |
| 01 | 0 | (2) |
| 10 | 1 | (3) |

# 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 $t$
$\theta$: binarisation threshold for X

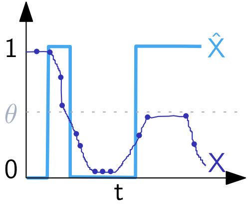


# ASK&D-BN— Local search

Generate candidates → Structure constraint → Dynamic constraint → Minimality constraint

**(2) input: time series**

$X_t$: continuous value of X at time $t$
$\theta$: binarisation threshold for X

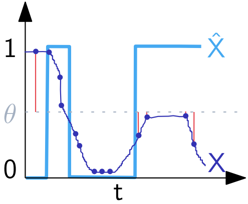


# ASK&D-BN— Local search

Generate candidates → Structure constraint → Dynamic constraint → Minimality constraint

(2) input: time series

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

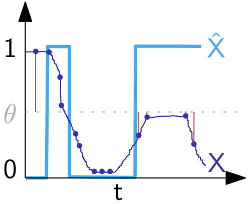


# ASK&D-BN— Local search

Generate candidates → Structure constraint → Dynamic constraint → Minimality constraint

**(2) input: time series**

$X_t$: continuous value of X at time $t$
$\theta$: 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 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 |

# ASK&D-BN— Local search

Generate candidates → Structure constraint → Dynamic constraint → Minimality constraint

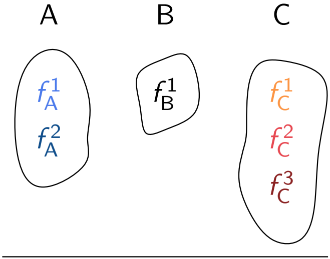
Select candidates with the smallest expressions (subset and/or cardinal minimal) ⇝ most general conditions

| 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 SBML2BNET

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

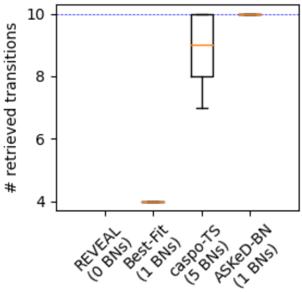
2. One specific variant of the complete approach on real-world reaction networks [Vaginay et al., 2021, Vaginay et al., 2022]:
   influence graph + time series and midrange binarisation

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

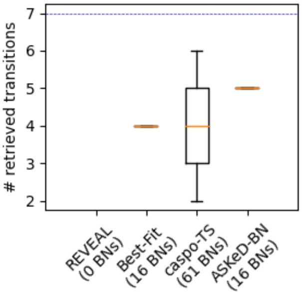
[1][Liang et al., 1998] [2][Lähdesmäki et al., 2003] [3][Ostrowski et al., 2016]

# ASK&D-BN versus REVEAL, Best-Fit, and Caspo-TS

*A. thaliana*
5 species, 10 transitions

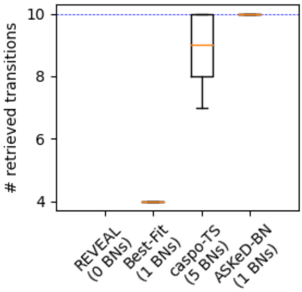



*yeast*
4 species, 7 transitions

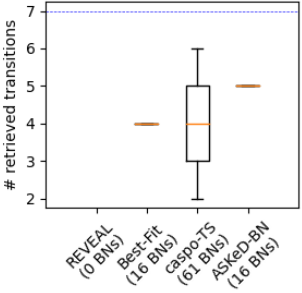


# ASK&D-BN versus REVEAL, Best-Fit, and Caspo-TS

*A. thaliana*
5 species, 10 transitions



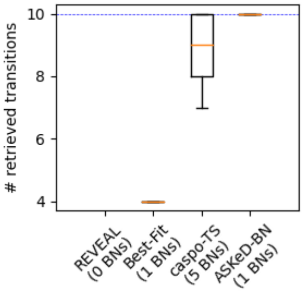

*yeast*
4 species, 7 transitions




- REVEAL fails

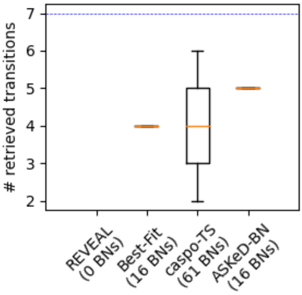
# ASK&D-BN versus REVEAL, Best-Fit, and Caspo-TS

*A. thaliana*
5 species, 10 transitions




- REVEAL fails

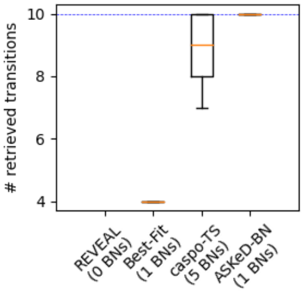
*yeast*
4 species, 7 transitions



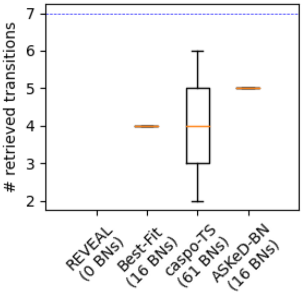

- Best-Fit lacks consistency

# ASK&D-BN versus REVEAL, Best-Fit, and Caspo-TS

*A. thaliana*
5 species, 10 transitions



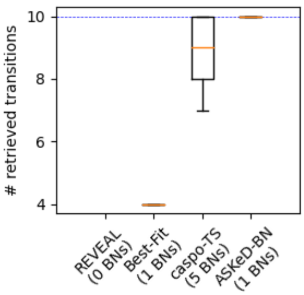

*yeast*
4 species, 7 transitions



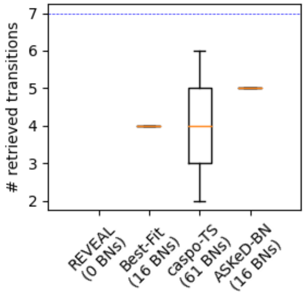

- REVEAL fails
- Caspo-TS returns more BNs, some of them with poor coverage because of reachability constraint
- Best-Fit lacks consistency

# ASK&D-BN versus REVEAL, Best-Fit, and Caspo-TS

*A. thaliana*
5 species, 10 transitions




*yeast*
4 species, 7 transitions

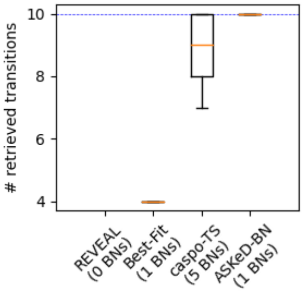



- REVEAL fails
- Caspo-TS returns more BNs, some of them with poor coverage because of reachability constraint

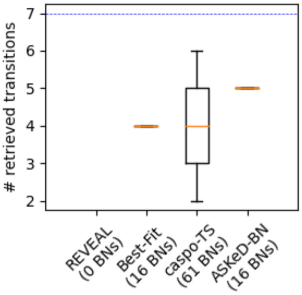
- Best-Fit lacks consistency
- ASK&D-BN returns a small number of BN, with good coverage and low variance ✓

# ASK&D-BN versus REVEAL, Best-Fit, and Caspo-TS

*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

# Results to real-world reaction networks (from BioModels[4])

**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

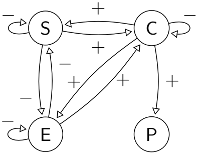
**Result:**

- 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

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

# A closer look: $\mathcal{R}_{\text{enz}}$

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




## Setting n°1

- influence graph
- time series
- binarised time series

midrange (0.8) and median (0.6):

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

→ Coverage depends on the binarisation procedure, BNs miss some influences

## Setting n°2

- full graph from abstract simulation

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

→ Perfect coverage but does not comply with the influence graph

$\Longrightarrow$ They do not capture the same thing

# Outline

# Relation to other abstractions

# Our abstraction versus other abstractions

Reaction-thinking

**Reaction network**

**Boolean network**

Influence thinking

# Our abstraction versus other abstractions

Reaction-thinking

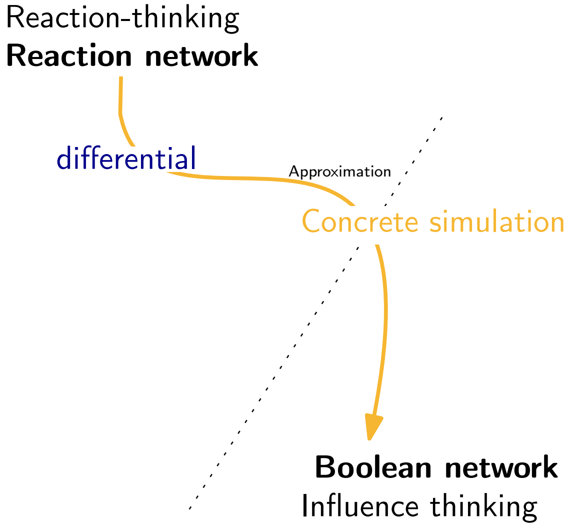
**Reaction network**

differential

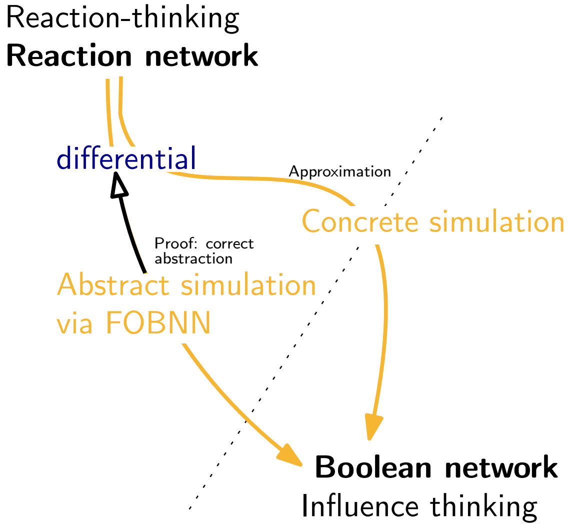
**Boolean network**

Influence thinking

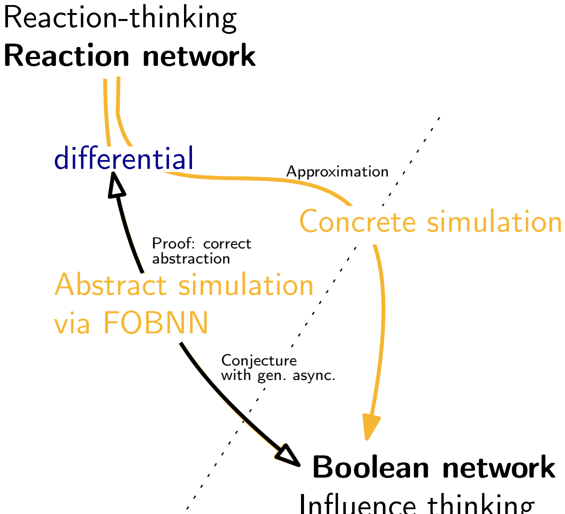
# Our abstraction versus other abstractions



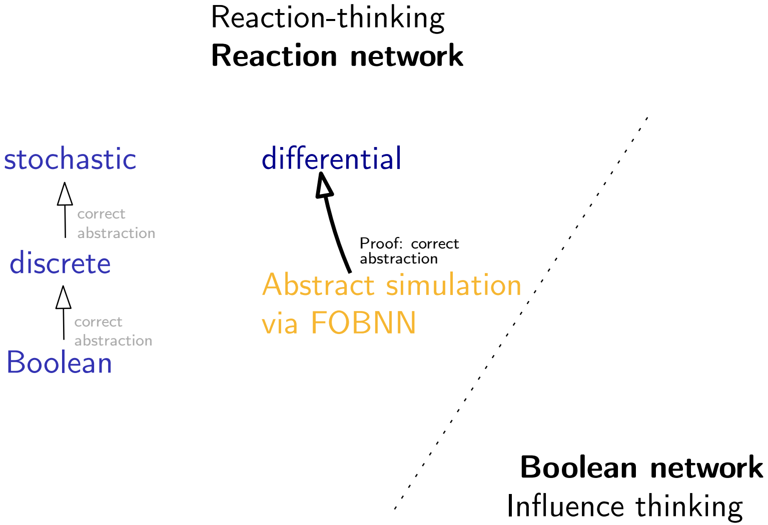
# Our abstraction versus other abstractions



# Our abstraction versus other abstractions

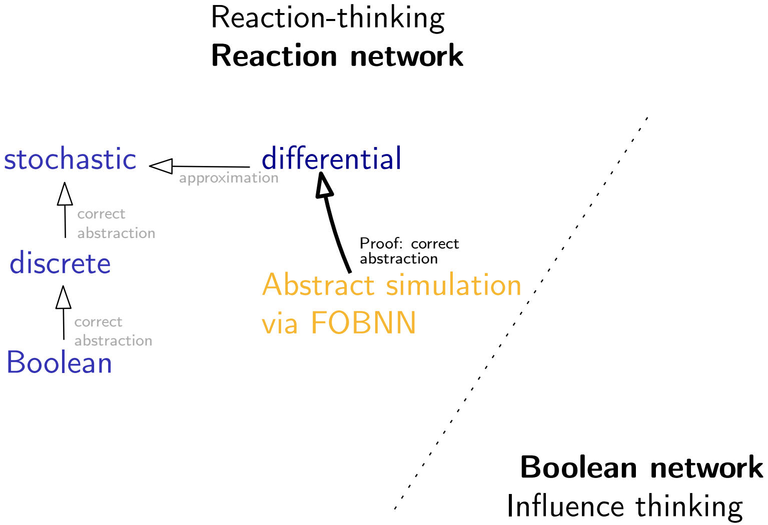


# Our abstraction versus other abstractions



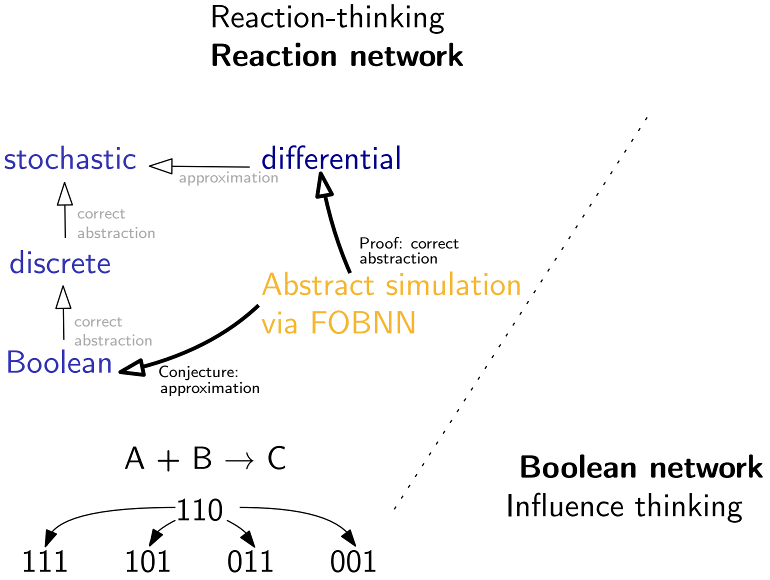

[Fages, Soliman, 2008a]

# Our abstraction versus other abstractions




[Fages, Soliman, 2008a]

# Our abstraction versus other abstractions




[Fages, Soliman, 2008a]

# Outline

# Conclusion and perspectives

# Conclusion

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

# Conclusion

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

- 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 guarentees. ✓

- 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 guarentees. ✓

- 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

# Perspectives

1. *Ad hoc* solution 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. Better understanding of the formal relationship between reaction networks and Boolean network
   Two conjectures to investigate, reverse process(*)

3. Improve the Boolean networks synthesis methods when applied to wet data
   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. *Ad hoc* solution 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. Better understanding of the formal relationship between reaction networks and Boolean network
   **Two conjectures to investigate, reverse process(*)**

3. Improve the Boolean networks synthesis methods when applied to wet data
   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

# 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
```

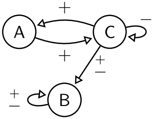
**Example**

`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 ⊆ influence graph of the reaction network



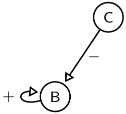

Do not select a conjunction that uses a forbidden literal

```
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 |
|---|---|

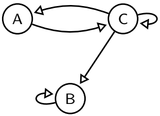
# ASK&D-BN— Local search

## 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**? ⇝ 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**? ⇝ Do not assume the underlying update scheme
Compare the truth table of a candidate function to the reconstructed truth table

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

| 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**? $\leadsto$ 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

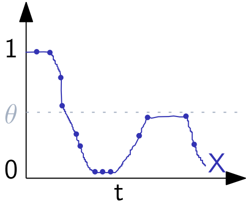
## Dynamics constraints

(2) input: time series

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

$X_t$: continuous value of X at time $t$

$\theta$: binarisation threshold for X



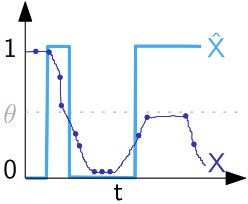
# ASK&D-BN— Local search

## Dynamics constraints

**(2) input: time series**

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

$X_t$: continuous value of X at time $t$
$\theta$: binarisation threshold for X



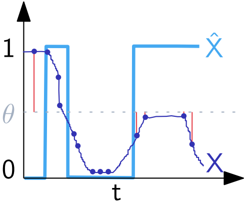
# ASK&D-BN— Local search

## Dynamics constraints

(2) input: time series

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

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



# ASK&D-BN— Local search

## Dynamics constraints

### (2) input: time series

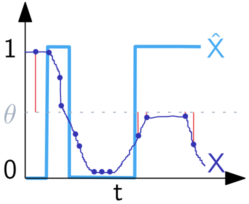
```
#minimize{E@2 : error(E)}. %
```

$X_t$: continuous value of X at time $t$

$\theta$: 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 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 |

# 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
```

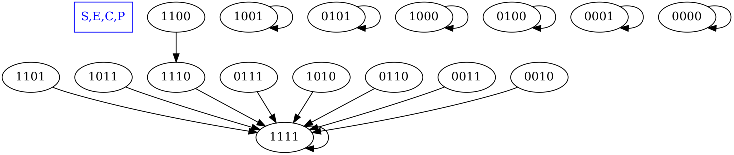
| putative input AB | observed output X | possible completions | | | |
|---|---|---|---|---|---|
| 00 | | 0 | 1 | 0 | 1 |
| 01 | 0 | 0 | 0 | 0 | 0 |
| 10 | 1 | 1 | 1 | 1 | 1 |
| 11 | | 0 | 0 | 1 | 1 |
| | subset minimal candidates | $A \wedge \neg B$ | $\neg B$ | $A$ | $A \vee \neg B$ |
| | size | 2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

card. min. candidates

# FOBNN fix-points with SAT

Given an FOBNN $\phi$ 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 $\alpha$ of $\phi$ 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. 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 $X \rightarrow Y$, 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, 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 $X \to Y$, 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, 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 $X \rightarrow Y$, 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 $X \rightarrow Y$, 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 $X \rightarrow Y$, 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 ($t[X] \in dom(X) = \mathbb{B}^k$)

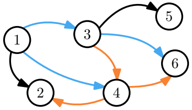
| $r$ | A | B | C | $\underset{\text{next}}{A}$ | $\underset{\text{next}}{B}$ | $\underset{\text{next}}{C}$ | $X \subseteq \mathcal{S}$ |
|---|---|---|---|---|---|---|---|
| $t_1$ | 0 | 0 | 0 | 0 | 0 | 0 | ● |
| $t_2$ | 0 | 1 | 1 | 1 | 0 | 0 | ● |
| $t_3$ | 0 | 0 | 0 | 0 | 0 | 1 | ● |

...

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 ($t[X] \in dom(X) = \mathbb{B}^k$)

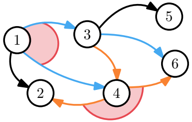
| $r$ | A | B | C | $\underset{\text{next}}{\text{A}}$ | $\underset{\text{next}}{\text{B}}$ | $\underset{\text{next}}{\text{C}}$ | $X \subseteq \mathcal{S}$ |
|---|---|---|---|---|---|---|---|
| $t_1$ | 0 | 0 | 0 | 0 | 0 | 0 | ● |
| $t_2$ | 0 | 1 | 1 | 1 | 0 | 0 | ● |
| $t_3$ | 0 | 0 | 0 | 0 | 0 | 1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

...

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 ($t[X] \in dom(X) = \mathbb{B}^k$)

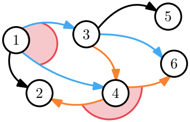
| $r$ | A | B | C | $\underset{\text{next}}{\text{A}}$ | $\underset{\text{next}}{\text{B}}$ | $\underset{\text{next}}{\text{C}}$ | $X \subseteq \mathcal{S}$ |
|---|---|---|---|---|---|---|---|
| $t_1$ | 0 | 0 | 0 | 0 | 0 | 0 | ● |
| $t_2$ | 0 | 1 | 1 | 1 | 0 | 0 | ● |
| $t_3$ | 0 | 0 | 0 | 0 | 0 | 1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

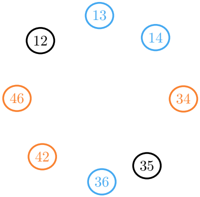
. . .

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)

# 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 ($t[X] \in dom(X) = \mathbb{B}^k$)

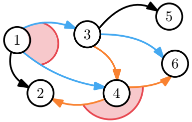
| $r$ | A | B | C | A next | B next | C next | $X \subseteq \mathcal{S}$ |
|---|---|---|---|---|---|---|---|
| $t_1$ | 0 | 0 | 0 | 0 | 0 | 0 | ● |
| $t_2$ | 0 | 1 | 1 | 1 | 0 | 0 | ● |
| $t_3$ | 0 | 0 | 0 | 0 | 0 | 1 | ● |

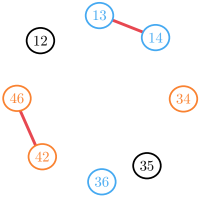
...

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 ($t[X] \in dom(X) = \mathbb{B}^k$)

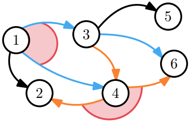
| $r$ | A | B | C | A next | B next | C next | $X \subseteq \mathcal{S}$ |
|---|---|---|---|---|---|---|---|
| $t_1$ | 0 | 0 | 0 | 0 | 0 | 0 | ● |
| $t_2$ | 0 | 1 | 1 | 1 | 0 | 0 | ● |
| $t_3$ | 0 | 0 | 0 | 0 | 0 | 1 | ● |

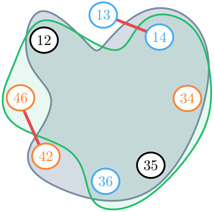
...

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.



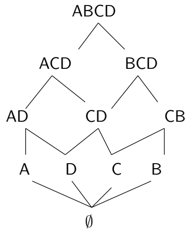

g3-error: minimal proportion of tuples to remove from $r$ to satisfy $f \rightsquigarrow$ coverage measure

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 $\mathcal{S}$: $\boldsymbol{\mathcal{R}} = \{R_i \rightarrow P_i\}_{i=1\dots m}$
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

$$\boldsymbol{\mathcal{R}} = \{$$
$$\mathcal{R}_1 : A + B \rightarrow C + D$$
$$\mathcal{R}_2 : A + C \rightarrow D$$
$$\mathcal{R}_3 : B + D \rightarrow C$$
$$\}$$

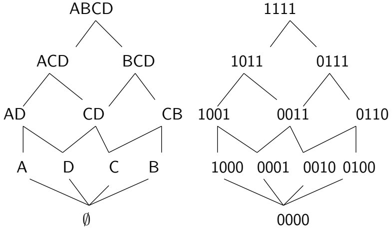



Simon Vilmin (AMU), Loïc Paulevé? (LABRI)

# Learn reaction networks from Boolean transitions

Reaction network with species in $\mathcal{S}$: $\boldsymbol{\mathcal{R}} = \{R_i \to P_i\}_{i=1\dots m}$
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

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




given a closure system, find the implication base(s)

$\stackrel{?}{=}$

given Boolean transitions, find the reaction network(s)

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

# Not well-formed reaction networks

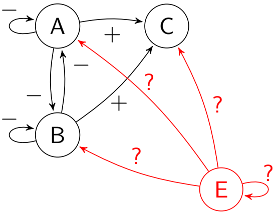
$$X \xrightarrow{k \times Y} \_$$

$\frac{\partial X}{\partial Y} \neq 0$ 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)

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

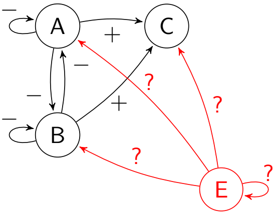



[5] [Fages et al. 2012]

# 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"[5], but some can be fixed → add a step in the pipeline

[5] [Fages et al. 2012]