

Automatically design Boolean networks from static and dynamical knowledge on a system

Stéphanie Chevalier
LRI, CNRS, U. Paris-Sud
U. Paris-Saclay, France
stephanie.chevalier@lri.fr

Christine Froidevaux
LRI, CNRS, U. Paris-Sud
U. Paris-Saclay, France
christine.froidevaux@lri.fr

Loïc Paulevé
LaBRI, CNRS, U. Bordeaux
Bordeaux INP, France
loic.pauleve@labri.fr

Andrei Zinovyev
Institut Curie, INSERM
U. PSL, Mines ParisTech, France
andrei.zinovyev@curie.fr

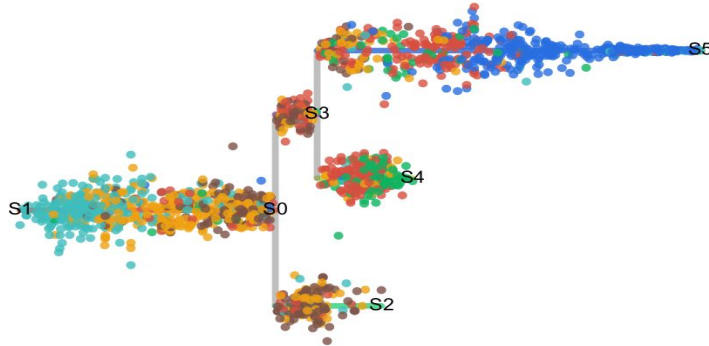
Automatically design **models** from **knowledge** on a system

(BOOLEAN NETWORKS)

(STRUCTURE & BEHAVIORS)

→ *The aim :*

Be able to model divergent processes (cell differentiation, perturbations, mutants...)



Automatically design **models** from **knowledge** on a system


(BOOLEAN NETWORKS)

(STRUCTURE & BEHAVIORS)

→ *The aim :*

Be able to model divergent processes (cell differentiation, perturbations, mutants...)

Offer an exhaustive enumeration (to study motifs / gene importance in the process)

 evolutionary optimization algorithms: *limited access to the space of solutions*

 satisfiability problems

Automatically design **models** from **knowledge** on a system

(BOOLEAN NETWORKS)

(STRUCTURE & BEHAVIORS)

→ *The aim :*

Be able to model divergent processes (cell differentiation, perturbations, mutants...)

Offer an exhaustive enumeration (to study motifs / gene importance in the process)

Be scalable for networks of more than 100 nodes, with non-deterministic dynamics

Automatically design **models** from **knowledge** on a system

(BOOLEAN NETWORKS)

(STRUCTURE & BEHAVIORS)

→ *The issue :*

Direct enumeration of the BNs compatible with the input data (static and dynamical knowledge)

→ *The methodology :*

Logical inference of a Boolean network from constraints on:



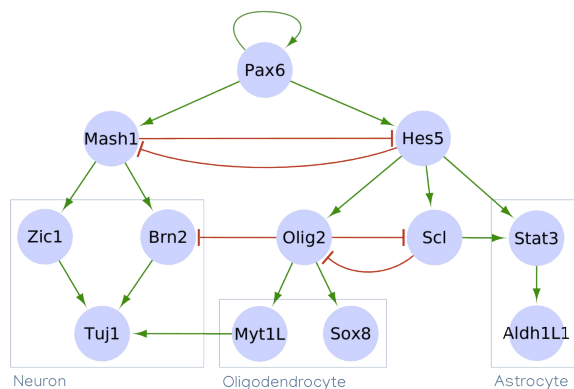
Automatically design **models** from **knowledge** on a system

(**BOOLEAN NETWORKS**)

(**STRUCTURE & BEHAVIORS**)

1) The data

STRUCTURE: **known and putative interactions**
between components



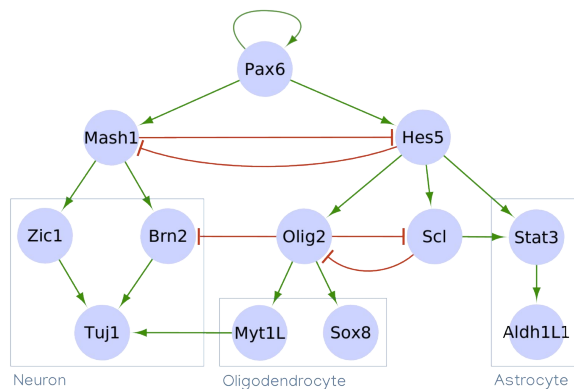
Automatically design **models** from **knowledge** on a system

(**BOOLEAN NETWORKS**)

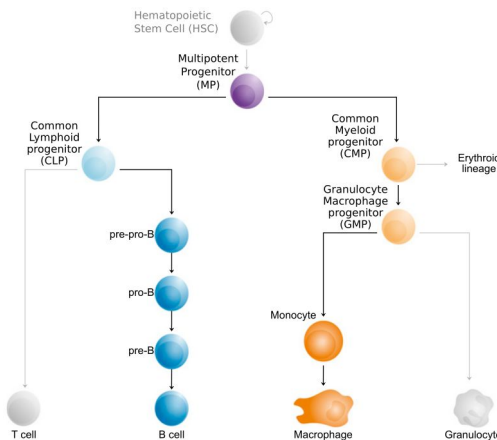
(**STRUCTURE & BEHAVIORS**)

1) The data

STRUCTURE: **known and putative interactions between components**



BEHAVIORS: **dynamics of biological observations along processes** which are (most of the time) ***partial observations*** of the system



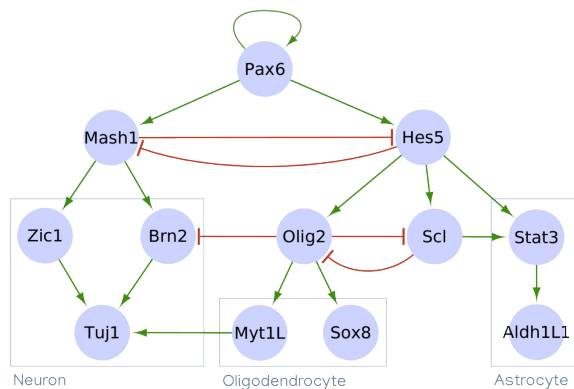
Automatically design **models** from **knowledge** on a system

(**BOOLEAN NETWORKS**)

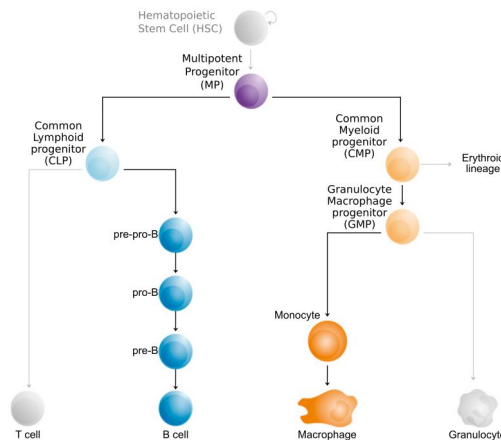
(**STRUCTURE & BEHAVIORS**)

1) The data

STRUCTURE: **known and putative interactions between components**



BEHAVIORS: **dynamics** of biological observations along processes which are (most of the time) *partial observations* of the system



example:



gene expr. in
CMP:

Flt3 = 1
Gfi1 = 0
...



gene expr. in
macrophage:

Flt3 = 1
Gfi1 = 1
...

...

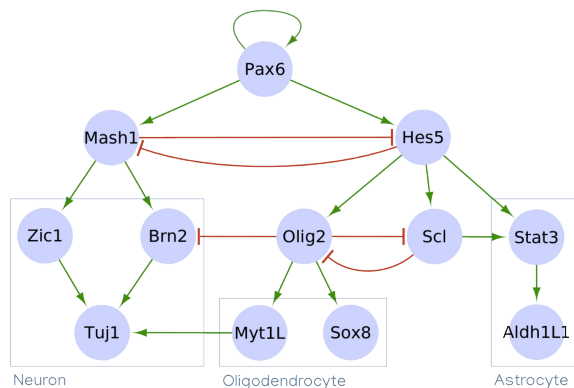
Automatically design **models** from **knowledge** on a system

(**BOOLEAN NETWORKS**)

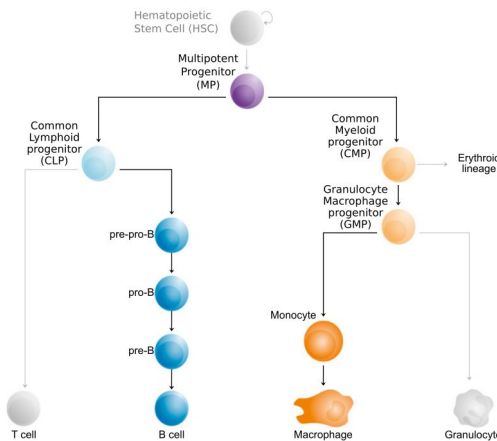
(**STRUCTURE & BEHAVIORS**)

1) The data

STRUCTURE: **known and putative interactions between components**



BEHAVIORS: **dynamics of biological observations along processes** which are (most of the time) ***partial observations*** of the system



example:



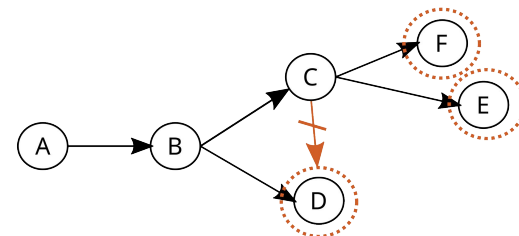
gene expr. in
CMP:

Flt3 = 1
Gfi1 = 0
...



gene expr. in
macrophage:

Flt3 = 1
Gfi1 = 1
...



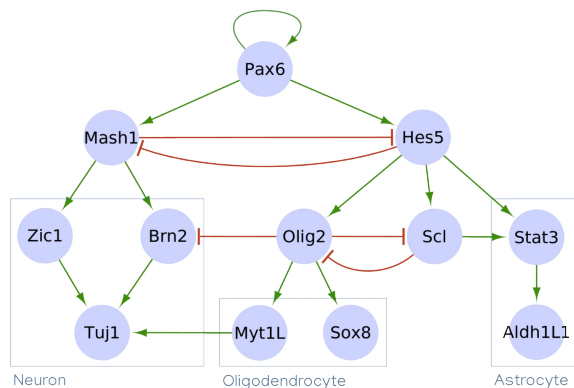
Automatically design **models** from **knowledge** on a system

(BOOLEAN NETWORKS)

(STRUCTURE & BEHAVIORS)

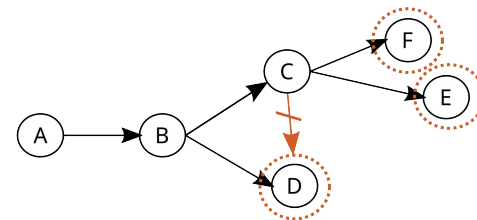
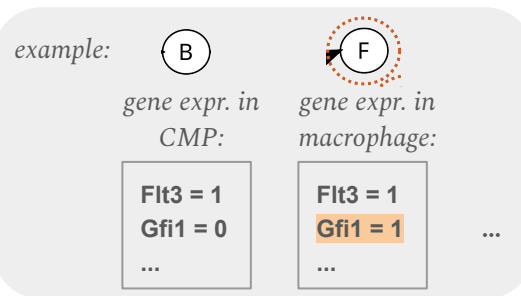
main point: **in input, the data are**

1) static knowledge (PKN)



constrains the domain of the
Boolean functions of the models

2) dynamical knowledge (observations)



constrains the dynamics of the models

Automatically design **models** from **knowledge** on a system

(**BOOLEAN NETWORKS**)

(**STRUCTURE & BEHAVIORS**)

2) Boolean network

BOOLEAN NETWORK: **discrete dynamical system**

A BN of dimension n

is a function $f = (f_1, \dots, f_n)$ with

$$\forall i \in \{1, \dots, n\}, f_i : \{0, 1\}^n \rightarrow \{0, 1\}$$

A configuration is a vector $x \in \{0, 1\}^n$

example for a BN with 3 nodes:

→ *the configuration 011 means:*

- ♦ *gene 1 is silenced*
- ♦ *genes 2 & 3 are expressed*

Automatically design **models** from **knowledge** on a system

(**BOOLEAN NETWORKS**)

(**STRUCTURE & BEHAVIORS**)

2) Boolean network

BOOLEAN NETWORK: discrete dynamical system

A BN of dimension n

is a function $f = (f_1, \dots, f_n)$ with

$$\forall i \in \{1, \dots, n\}, f_i : \{0, 1\}^n \rightarrow \{0, 1\}$$

A configuration is a vector $x \in \{0, 1\}^n$

example of
a BN with
3 nodes:

$$f_1(x) := \neg x_2$$

$$f_2(x) := \neg x_1$$

$$f_3(x) := \neg x_1 \wedge x_2$$

Automatically design **models** from **knowledge** on a system

(BOOLEAN NETWORKS)

(STRUCTURE & BEHAVIORS)

2) Boolean network

BOOLEAN NETWORK: **discrete dynamical system**

A BN of dimension n

is a function $f = (f_1, \dots, f_n)$ with

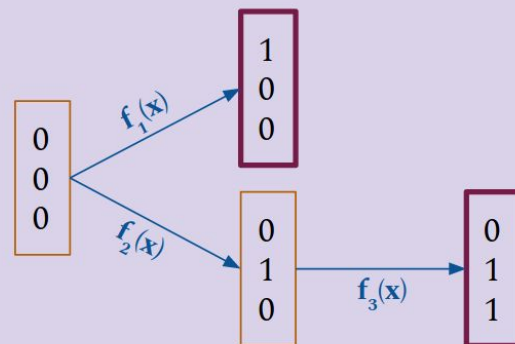
$$\forall i \in \{1, \dots, n\}, f_i : \{0, 1\}^n \rightarrow \{0, 1\}$$

A **configuration** is a vector $x \in \{0, 1\}^n$

example of
a BN with
3 nodes:

$$\begin{aligned} f_1(x) &:= \neg x_2 \\ f_2(x) &:= \neg x_1 \\ f_3(x) &:= \neg x_1 \wedge x_2 \end{aligned}$$

Dynamics of a BN:



fully asynchronous dynamics of f

→ transition stable state

Automatically design **models** from **knowledge** on a system

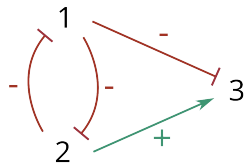
(BOOLEAN NETWORKS)

(STRUCTURE & BEHAVIORS)

Boolean network inference: a complex problem

STRUCTURE: **known and putative interactions between components**

specifies the domain
of the compatible BNs



Possible rules for node 3:

$$\begin{array}{ll} f_3(x) = 0 & ; \quad f_3(x) = x_2 \\ f_3(x) = 1 & ; \quad f_3(x) = \neg x_1 \wedge x_2 \\ f_3(x) = \neg x_1 & ; \quad f_3(x) = \neg x_1 \vee x_2 \end{array}$$

BEHAVIORS: dynamics of observations along processes
which are (most of the time) *partial observations* of the system

Automatically design **models** from **knowledge** on a system

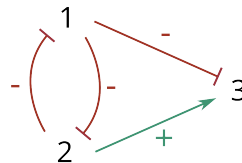
(**BOOLEAN NETWORKS**)

(**STRUCTURE & BEHAVIORS**)

Boolean network inference: a complex problem

STRUCTURE: **known and putative interactions between components**

specifies the domain of the compatible BNs



Possible rules for node 3:

$$\begin{aligned} f_3(x) = 0 & \quad ; \quad f_3(x) = x_2 \\ f_3(x) = 1 & \quad ; \quad f_3(x) = \neg x_1 \wedge x_2 \\ f_3(x) = \neg x_1 & \quad ; \quad f_3(x) = \neg x_1 \vee x_2 \end{aligned}$$

Combinatorial problem:

<i>indegree</i>	<i># monotonic Boolean functions</i>
0	2
2	6
4	168
6	7,828,354
8	56,130,437,228,687,557,907,788

BEHAVIORS: dynamics of observations along processes which are (most of the time) *partial observations* of the system

Automatically design **models** from **knowledge** on a system

(**BOOLEAN NETWORKS**)

(**STRUCTURE & BEHAVIORS**)

Boolean network inference: a complex problem

STRUCTURE: **known and putative interactions between components**

<i>indegree</i>	<i># monotonic Boolean functions</i>
0	2
2	6
4	168
6	7,828,354
8	56,130,437,228,687,557,907,788

combinatorial explosion

BEHAVIORS: dynamics of observations along processes which are (most of the time) *partial observations* of the system

Automatically design **models** from **knowledge** on a system

(BOOLEAN NETWORKS)

(STRUCTURE & BEHAVIORS)

Boolean network inference: a complex problem

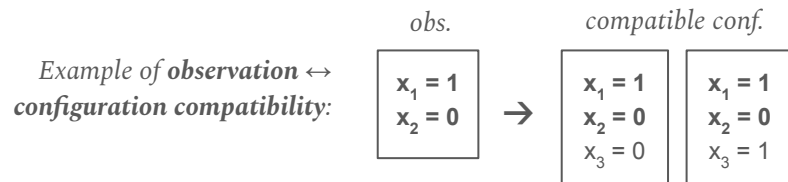
STRUCTURE: **known and putative interactions between components**

indegree	# monotonic Boolean functions
0	2
2	6
4	168
6	7,828,354
8	56,130,437,228,687,557,907,788

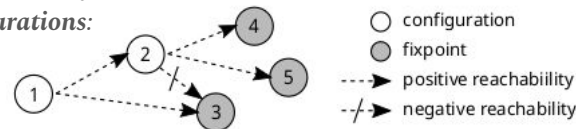
combinatorial explosion

BEHAVIORS: **dynamics** of observations along processes which are (most of the time) **partial observations** of the system

a BN is compatible if, in its dynamics, configurations compatible with the partial observations respect the behaviors (**reachability, stable properties**)



Example of dynamics of compatible configurations:



Automatically design **models** from **knowledge** on a system

(**BOOLEAN NETWORKS**)

(**STRUCTURE & BEHAVIORS**)

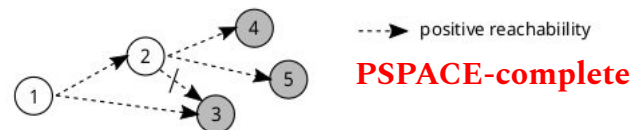
Boolean network inference: a complex problem

STRUCTURE: **known and putative interactions**
between components

<i>indegree</i>	<i># monotonic Boolean functions</i>
0	2
2	6
4	168
6	7,828,354
8	56,130,437,228,687,557,907,788

combinatorial explosion

BEHAVIORS: **dynamics** of observations along processes
which are (most of the time) *partial observations* of the system



hard complexity

Automatically design **models** from **knowledge** on a system

(BOOLEAN NETWORKS)

(STRUCTURE & BEHAVIORS)

Boolean network inference: a complex problem

combinatorial explosion & hard complexity

⇨ strategy:

Formulate the inference as a Boolean satisfiability problem

Answer-Set Programming: designed for solving combinatorial satisfaction problem

Domain & observations **taken into account** during the enumeration: ~~model checking~~

Automatically design **models** from **knowledge** on a system

(BOOLEAN NETWORKS)

(STRUCTURE & BEHAVIORS)

1) Answer-Set Programming (ASP) syntax

brief overview
of ASP syntax

A Logic Program in ASP is a set of logical rules of the form:

$$a_0 \leftarrow a_1, \dots, a_n, \text{ not } a_{n+1}, \dots, \text{ not } a_{n+k}.$$

with integrity constraints as:

$$\leftarrow a_1, \dots, a_n, \text{ not } a_{n+1}, \dots, \text{ not } a_{n+k}.$$

Suitable for solving combinatorial satisfaction problem

Computes **stable models**: minimal sets of a_i satisfying the rules



[Gelfond and Lifschitz, 1988]

Automatically design **models** from **knowledge** on a system

(BOOLEAN NETWORKS)

(STRUCTURE & BEHAVIORS)

2) Our encoding

Main lines of the logic program:

- the description of a BN
- the domain of its functions
= *PKN*
- the way to compute its dynamic
= *semantics*
- the properties of its dynamics
= *observations*

The solver enumerates the solutions
(solutions = BNs compatible with data = models)

The implementation of BNs:

Boolean function:

expressed in propositional logic
under Disjunctive Normal Form

encoded by `clause(N,C,L,S)`
predicates such that:

- atom *L*
- with sign *S* (-1, 1)
- is included in the *C*th clause
- of *f_N*

example: $f_a(x) = x_c \vee (\neg x_a \wedge x_b)$



is encoded: `clause(a,1,c,1).`
`clause(a,2,a,-1).`
`clause(a,2,b,1).`

Encoding of the canonicity for exhaustive enumeration:

2 solutions = 2 non-equivalent BNs

⇔ enforced by a total ordering between the clauses

Automatically design **models** from **knowledge** on a system

(BOOLEAN NETWORKS)

(STRUCTURE & BEHAVIORS)

encoding: 2 families of dynamical constraints:

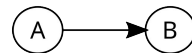
existence of a property vs **universality of a property**

Existential dynamical constraints:

$\exists \dots$  $\forall \dots \exists \dots$ 

time series: positive reachability

\exists path between configurations compatible with successive observations.



Automatically design **models** from **knowledge** on a system

(BOOLEAN NETWORKS)

(STRUCTURE & BEHAVIORS)

encoding: 2 families of dynamical constraints:

existence of a property vs **universality of a property**

Existential dynamical constraints:

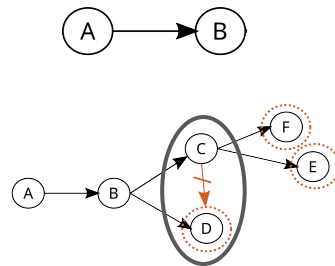
$\exists \dots$  $\forall \dots \exists \dots$ 

time series: positive reachability

\exists path between configurations compatible with successive observations.

bifurcating trajectories: negative reachability

\nexists path between configurations compatible with bifurcating observations.



Automatically design **models** from **knowledge** on a system

(BOOLEAN NETWORKS)

(STRUCTURE & BEHAVIORS)

encoding: 2 families of dynamical constraints:

existence of a property vs **universality of a property**

Existential dynamical constraints:



time series: positive reachability

\exists path between configurations compatible with successive observations.

bifurcating trajectories: negative reachability

\nexists path between configurations compatible with bifurcating observations.

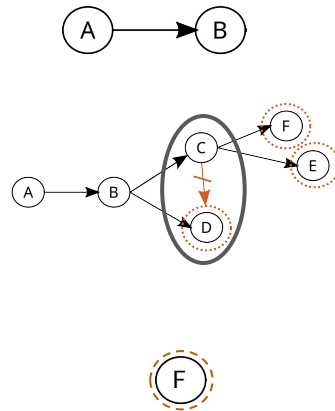
stable behaviors:

- **fixpoint**

A config. compatible with a stable observation is a fixpoint.

- **trapspace:**

Given an obs. with stability hypotheses on some nodes, these nodes are fixed from a compatible configuration.



Automatically design **models** from **knowledge** on a system

(BOOLEAN NETWORKS)

(STRUCTURE & BEHAVIORS)

encoding: 2 families of dynamical constraints:

existence of a property vs **universality of a property**

Universal dynamical constraints:

$\exists \dots$ 

$\forall \dots \exists \dots$ 

⇒ 2QBF constraints (*saturation technique [Eiter & Gottlob - 1995]*)

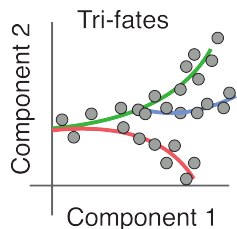
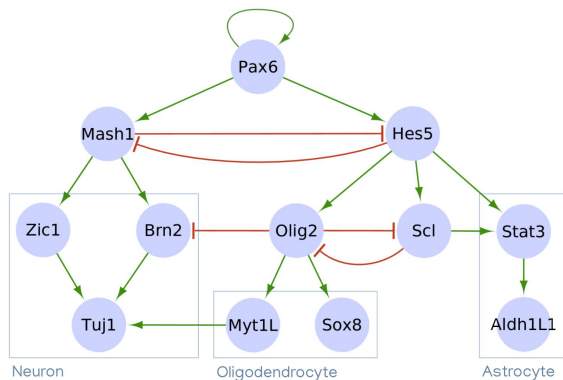
Work in
Progress

Automatically design **models** from **knowledge** on a system

(BOOLEAN NETWORKS)

(STRUCTURE & BEHAVIORS)

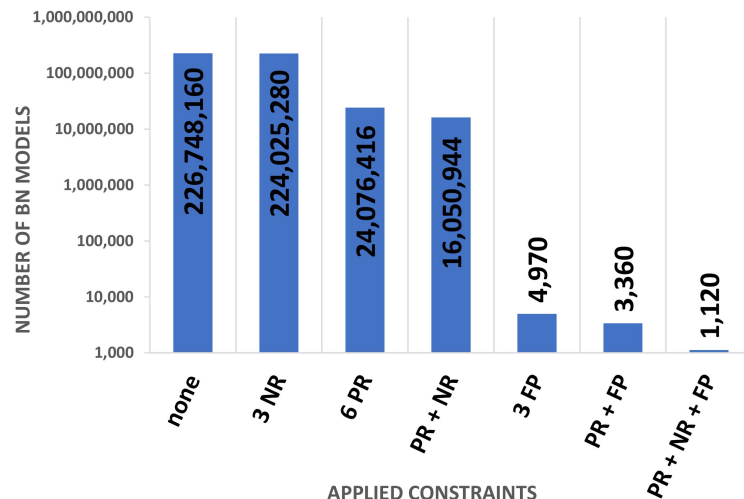
1) Test of constraint impact: on a biological application

central nervous system development

6 pos. reach (PR)
 3 neg. reach (NR)
 3 fixpoint (FP)

Impact of the constraints:

NUMBER OF BNs COMPATIBLE WITH CNS DATA
 W.R.T. VARIOUS PROPERTIES



Automatically design **models** from **knowledge** on a system

(BOOLEAN NETWORKS)

(STRUCTURE & BEHAVIORS)

2) Larger applications

In the case of larger domains and dynamical data not constraining enough:

~~exhaustive~~ enumeration (too many solutions to be exploited)

- ⇒ **enforce diversity during enumeration:**
to run partial inference while keeping
the benefit of having sets of models

Work in
Progress

Automatically design **models** from **knowledge** on a system

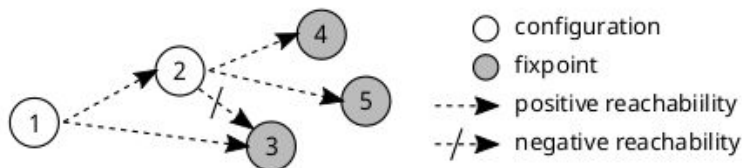
(BOOLEAN NETWORKS)

(STRUCTURE & BEHAVIORS)

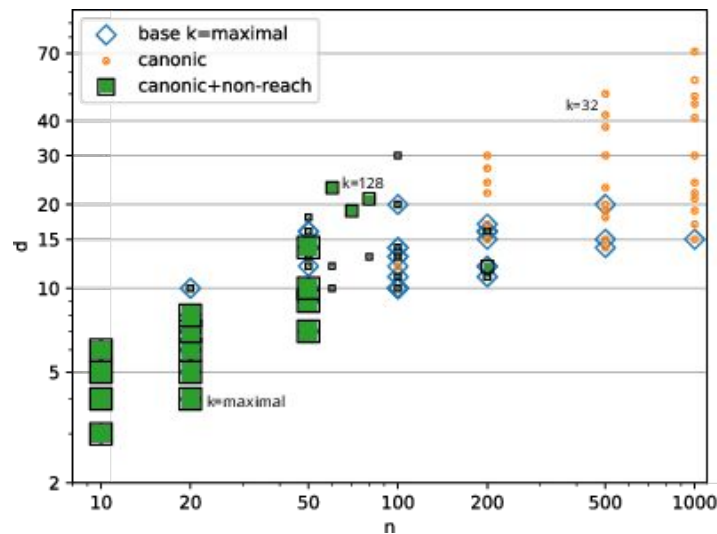
2) Test of scalability: inference on random network

Domains: random scale-free directed graphs,
with different biases on the in-degree of nodes

Dynamics: a generic dynamical property
of a two stages differentiation process



⇒ Partial enumeration **up to 1000 nodes**



Successfully solved instances
within 2h of CPU time (2.5Ghz)

Automatically design **models** from **knowledge** on a system

(BOOLEAN NETWORKS)

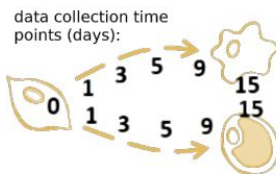
(STRUCTURE & BEHAVIORS)

3) Examples of applications

1) APPLICATION WITH RNA-SEQ DATA TO STUDY MESENCHYMAL DIFFERENTIATION

Differentiation data

binarization (tool *RefBool*^[4]) of RNA-Seq bulk data collected at multiple time points



PKN

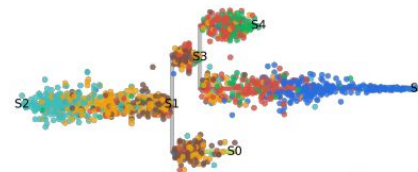
the network of transcription factors extracted from MetaCore (~1000 nodes)

- **RESULTS** in 10-30 min with ~16-32 Go partial enumeration of compatible BNs

2) APPLICATION WITH scRNA-SEQ DATA TO STUDY HEMATOPOIESIS

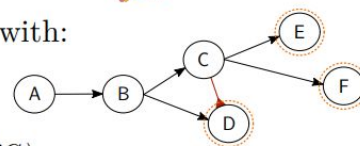
Differentiation data

from a pseudo-time trajectory, (built from single-cell data on differentiating cells^[3] and the tool *STREAM*^[5])



construction of a dynamical model with:

- observations at extremities of branches, containing the binarized expression of 19 transition genes (*TG*)
- 5 PR + 1 NR + 3 TP or 3 FP



PKN

the whole SIGNOR database (5656 nodes)

Automatically design **models** from **knowledge** on a system

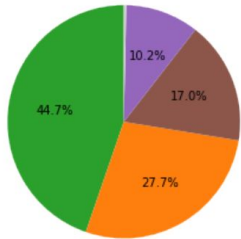
(BOOLEAN NETWORKS)

(STRUCTURE & BEHAVIORS)

4) After the inference,
how perform simulations with a set of models?

Ensemble modelling with MaBoSS (usable via *pymaboss*):
i.e. simulations from sets of Boolean networks

For questions:  vincent.noel@curie.fr



Automatically design **models** from **knowledge** on a system

(BOOLEAN NETWORKS)

(STRUCTURE & BEHAVIORS)

Contribution:

Boolean network inference method in ASP

Features w.r.t. the state of the art:

- new constraints (negative reachability, trapspace)
- mix reachability and stable properties
- scalability

Work in progress:

- Encoding of 2QBF constraints to check universal properties
- Application on single-cell differentiation data, using cells as time points

Thank you for your attention !
Do you have questions?



*Synthesis of Boolean Networks from
Biological Dynamical Constraints
using Answer-Set Programming*

Stéphanie Chevalier, Christine Froidevaux, Andrei Zinovyev, Loïc Paulevé