# Sequential Reprogramming of Biological Network Fate

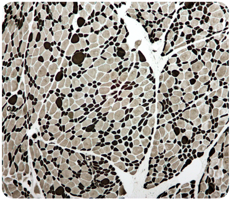
Jérémie Pardo Sergiu Ivanov Franck Delaplace

IBISC, Université Paris-Saclay, Univ Évry

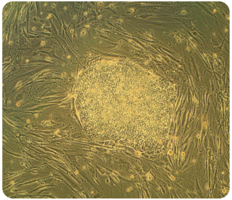
# Cell fate

...

Muscle cells

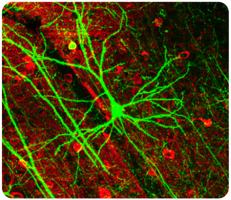


[2]



[1]

Embryonic stem cells

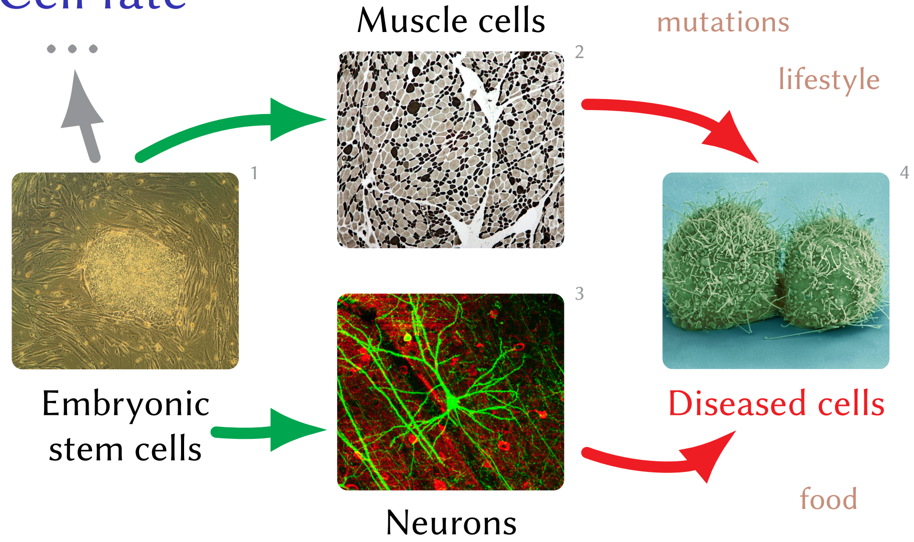


[3]

Neurons

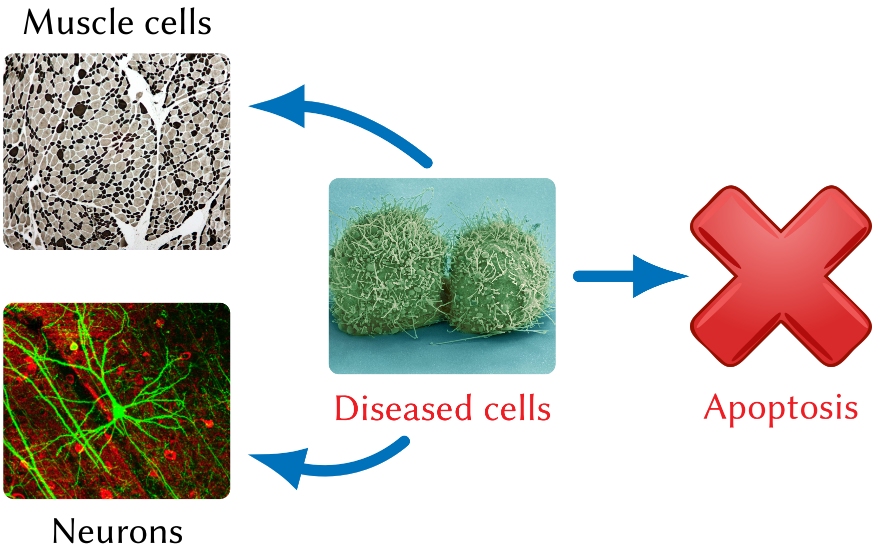
[1] https://commons.wikimedia.org/w/index.php?curid=2148036
[2] https://commons.wikimedia.org/w/index.php?curid=12772068
[3] DOI:10.1371/journal.pbio.0040029

# Cell fate



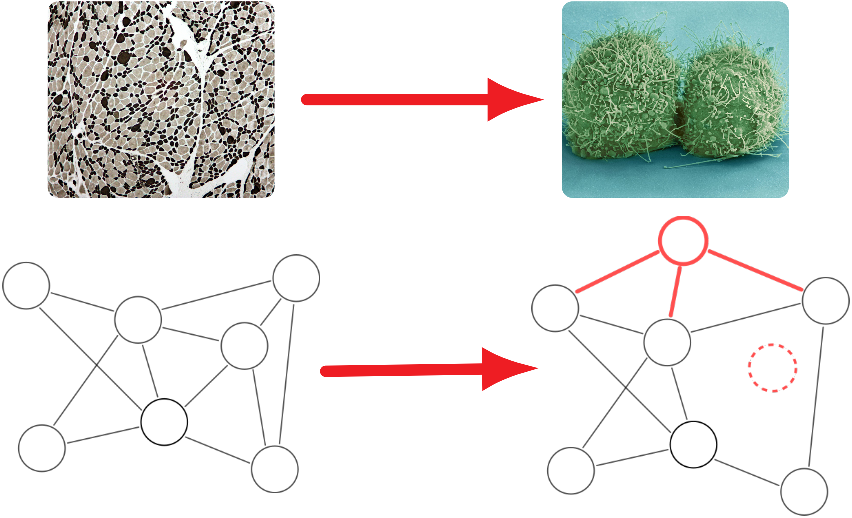

[1] https://commons.wikimedia.org/w/index.php?curid=2148036
[2] https://commons.wikimedia.org/w/index.php?curid=12772068
[3] DOI:10.1371/journal.pbio.0040029
[4] https://en.wikipedia.org/wiki/File:HeLa-V.jpg

# Cell fate reprogramming



# Network medicine

## Disease = perturbations of biological networks



# Therapy = network reprogramming

Reprogram a diseased network to make it healthy again

# Network controllability

How to formally capture biological networks?

How to reprogram a formal network?

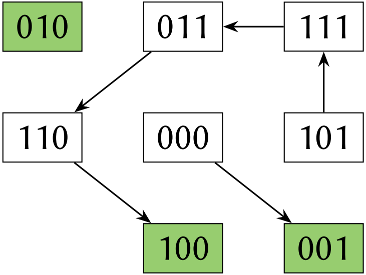
# Boolean networks

Boolean variables + boolean update functions

$$
\begin{aligned}
f_{x_1} &= (x_1 \wedge \overline{x}_2) \vee (x_1 \wedge \overline{x}_3) \vee (\overline{x}_1 \wedge x_2 \wedge x_3) \\
f_{x_2} &= (x_1 \wedge x_3) \vee (\overline{x}_1 \wedge x_2) \\
f_{x_3} &= (x_1 \wedge x_3) \vee (\overline{x}_1 \wedge \overline{x}_2)
\end{aligned}
$$

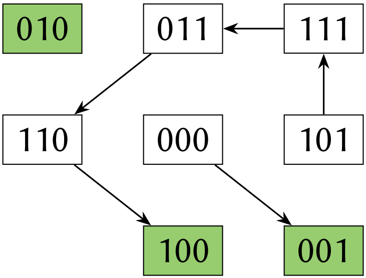
---

Synchronous dynamics: all variables are always updated




Stable states: 010, 100, 001.

# Stable states ~ phenotypes



# Reprogramming of Boolean networks

Diseased
stable states

→

Healthy
stable states

# Boolean control networks



$$
\begin{aligned}
f_{x_1} &= (x_1 \wedge \overline{x}_2) \vee (x_1 \wedge \overline{x}_3) \vee (\overline{x}_1 \wedge x_2 \wedge x_3) \\
f_{x_2} &= (x_1 \wedge x_3) \vee (\overline{x}_1 \wedge x_2) \\
f_{x_3} &= \big((x_1 \wedge x_3) \vee (\overline{x}_1 \wedge \overline{x}_2)\big) \wedge u^0 \vee \overline{u^1}
\end{aligned}
$$

## Control inputs:

$u^0 \leftarrow 0$ — freezes $x_3$ to 0

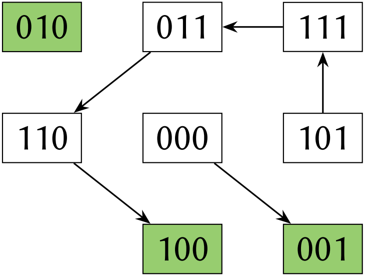
$u^1 \leftarrow 0$ — freezes $x_3$ to 1

Célia Biane, Franck Delaplace. Causal reasoning on Boolean control networks based on abduction: theory and application to Cancer drug discovery. IEEE/ACM Trans. Comput. Biol. Bioinform. (2018).

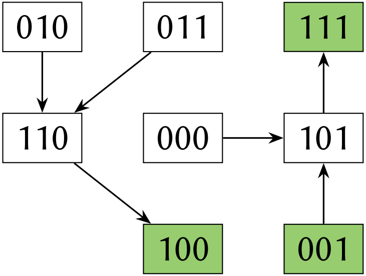
# BCN dynamics

$$
\begin{aligned}
f_{x_1} &= (x_1 \wedge \overline{x}_2) \vee (x_1 \wedge \overline{x}_3) \vee (\overline{x}_1 \wedge x_2 \wedge x_3) \\
f_{x_2} &= (x_1 \wedge x_3) \vee (\overline{x}_1 \wedge x_2) \\
f_{x_3} &= (x_1 \wedge x_3) \vee (\overline{x}_1 \wedge \overline{x}_2)
\end{aligned}
$$

Uncontrolled




$x_1$ frozen to 1



# One-shot reprogramming

Drive a network to a given set of states.

---

Célia Biane, Franck Delaplace. Causal reasoning on Boolean control networks based on abduction: theory and application to Cancer drug discovery. IEEE/ACM Trans. Comput. Biol. Bioinform. (2018).

- Showed that inference is NP-hard.
- Gave a control inference algorithm.
  - prime implicants
  - integer linear programming (ILP)
  - parsimonious controls

# Sequential control

Sequential control yields better therapies.

Michael Lee, Albert S. Ye, Alexandra K. Gardino, Anne Heijink, Peter Sorger, Gavin Macbeath, and Michael Yaffe. Sequential application of anti-cancer drugs enhances cell death by re-wiring apoptotic signaling networks. Cell, 149:780–794, 05 2012.

Sequential control better models sequential processes.

Eric R. Fearon and Bert Vogelstein. A genetic model for colorectal tumorigenesis. Cell, 61(5):759–767, 1990.

# Control sequences in BCN

$$
\begin{aligned}
f_{x_1} &= (x_1 \wedge \overline{x}_2) \vee (x_1 \wedge \overline{x}_3) \vee (\overline{x}_1 \wedge x_2 \wedge x_3) \\
f_{x_2} &= (x_1 \wedge x_3) \vee (\overline{x}_1 \wedge x_2) \\
f_{x_3} &= \big((x_1 \wedge x_3) \vee (\overline{x}_1 \wedge \overline{x}_2)\big) \wedge u^0 \vee \overline{u^1}
\end{aligned}
$$

Control inputs $U = \{u^i\}$

Control $\mu : U \to \{0, 1\}$

Control sequence $\mu_{[k]} = (\mu_1, \ldots, \mu_k)$

# Sequentially controlled dynamics

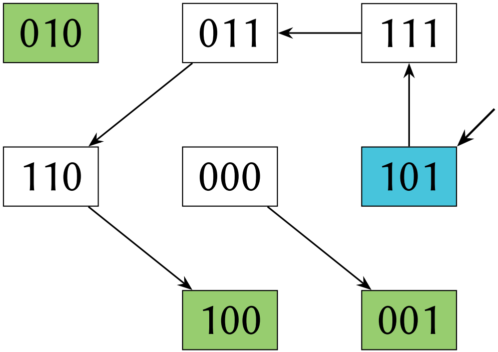
Control sequence $\mu_{[k]} = (\mu_1, \ldots, \mu_k)$

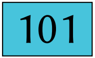
Master system —drives→ Boolean control network

1. Apply $\mu_i$ for $k$ steps.
   - $k$ chosen non-deterministically
2. Switch to $\mu_{i+1}$.

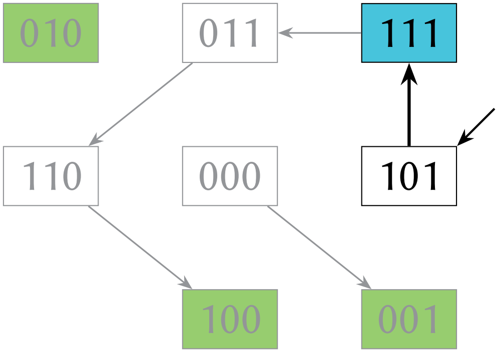
Independent semantics

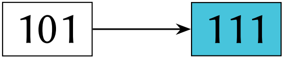
# Example of independent semantics



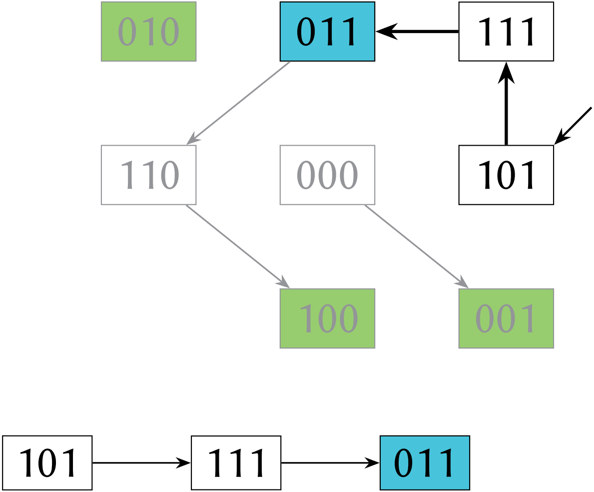



# Example of independent semantics

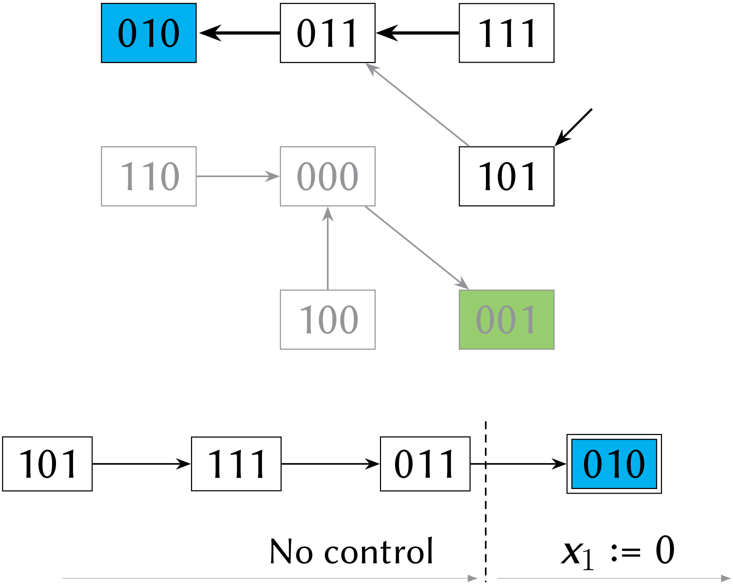


# Example of independent semantics

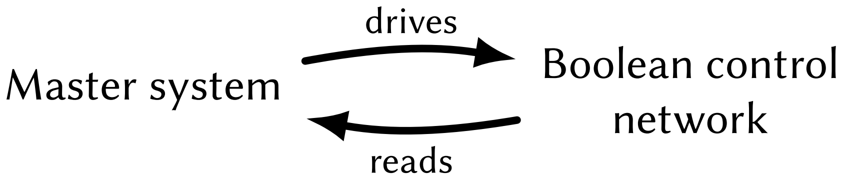


# Example of independent semantics



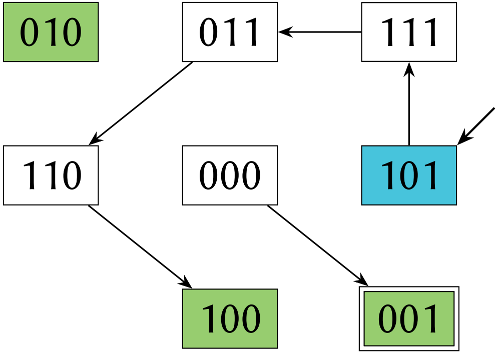
# Biological time and ConEvs semantics

Control sequence $\mu_{[k]} = (\mu_1, \dots, \mu_k)$

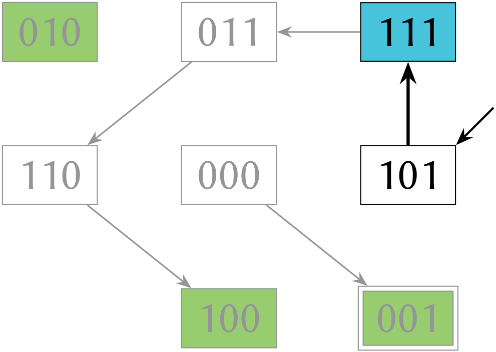


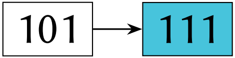

1. Apply $\mu_i$.
2. Switch to $\mu_{i+1}$ at a stable state.

# Example of ConEvs sematics

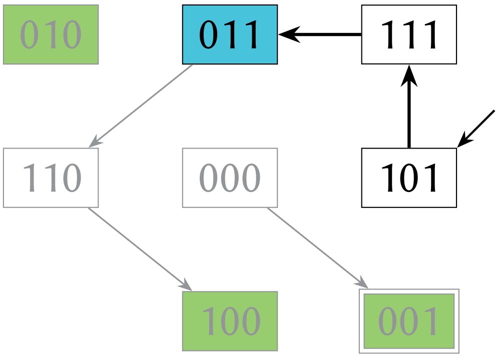



# Example of ConEvs sematics

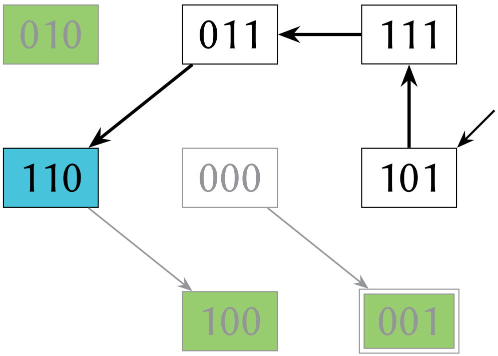


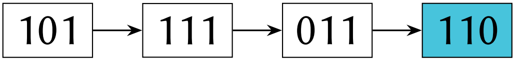



# Example of ConEvs sematics

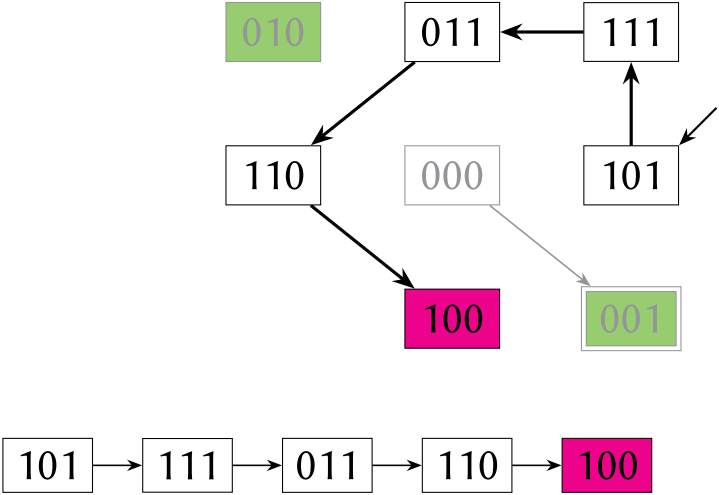



# Example of ConEvs sematics

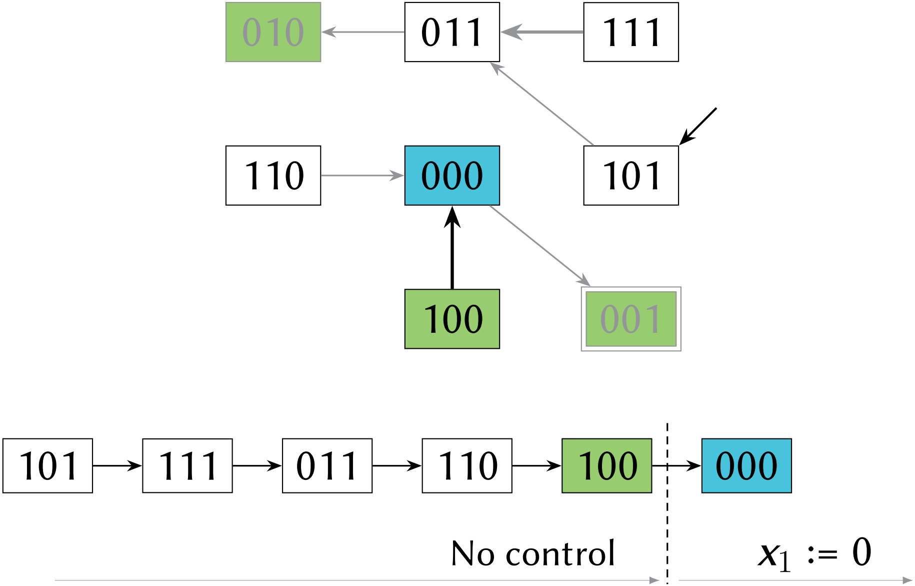


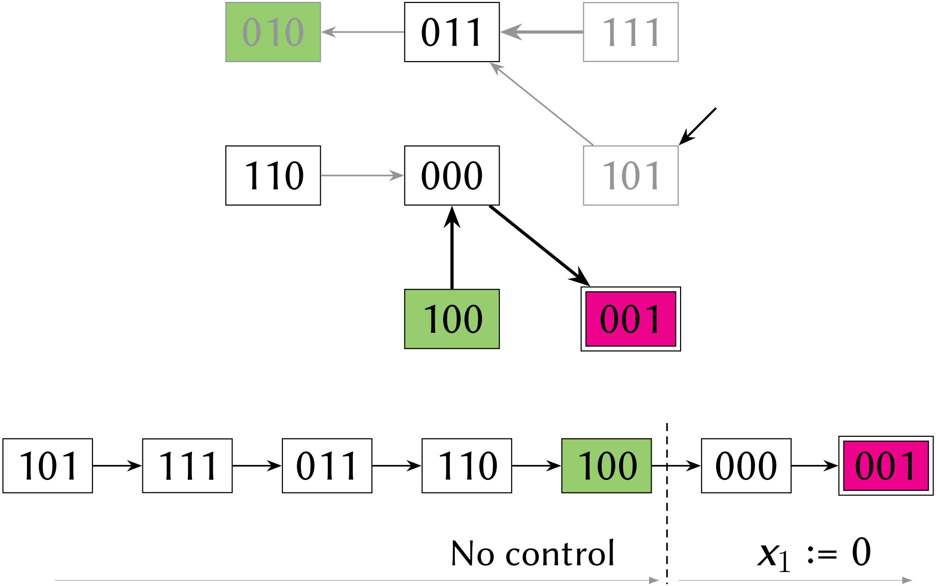
# Example of ConEvs sematics



# Example of ConEvs sematics



# Example of ConEvs sematics



# The CoFaSe inference problem

Starting states —control sequence?→ Target stable states

# CoFaSe is PSPACE-hard.

CoFaSe generalizes reachability (PSPACE-complete).

---

- very hard, “much harder” than NP-hard
- at least as hard as PSPACE
  - problems solvable in polynomial space

# Uncontrollable variables

$$
\begin{aligned}
f_{x_1} &= (x_1 \wedge \overline{x}_2) \vee (x_1 \wedge \overline{x}_3) \vee (\overline{x}_1 \wedge x_2 \wedge x_3) \\
f_{x_2} &= (x_1 \wedge x_3) \vee (\overline{x}_1 \wedge x_2) \\
f_{x_3} &= \big((x_1 \wedge x_3) \vee (\overline{x}_1 \wedge \overline{x}_2)\big) \wedge u^0 \vee \overline{u^1}
\end{aligned}
$$

Controllable variables $CV = \{x_3\}$

Uncontrollable variables $UCV = \{x_1, x_2\}$

---

Factorize the state space by UCV profiles.

# Practical bound on complexity

**Theorem**

The length $k$ of a minimal control sequence $\mu_{[k]}$ is

$$k \leq \alpha\, 2^{|UCV|} \quad \alpha \text{ constant}$$

$^{\star}$ $\mu_{[k]}$ is minimal if $\nexists \nu_{[m]}$ with $m < k$.

---

UCV $\sim$ biomarkers, few in practice

# Exact bounds on complexity

## Independent semantics

$$k \leq 2^{|UCV|}$$

Don't visit any configuration of UCV more than once.

## ConEvs semantics

$$k \leq 2^{|UCV|+1}$$

Don't visit any configuration of UCV more than twice.

# Algorithm 1: Exhaustive search

$S_\alpha$ the set of starting states $S_\omega$ the set of target states

## Phase 1

Try inferring a one-shot control$^\star$ leading to $S_\omega$.

## Phase 2

Try inferring a one-shot control$^\star$ leading to a yet unvisited configuration of UCV.

Restart from the reached configurations.

Célia Biane and Franck Delaplace. Causal reasoning on Boolean control networks based on abduction: theory and application to cancer drug discovery. IEEE/ACM transactions on computational biology and bioinformatics, 2018.

# Algorithm 2: Total control

The dynamics of CV almost never matters.

$$\Downarrow$$

Infer sequences of total controls.

- a total control freezes all CV

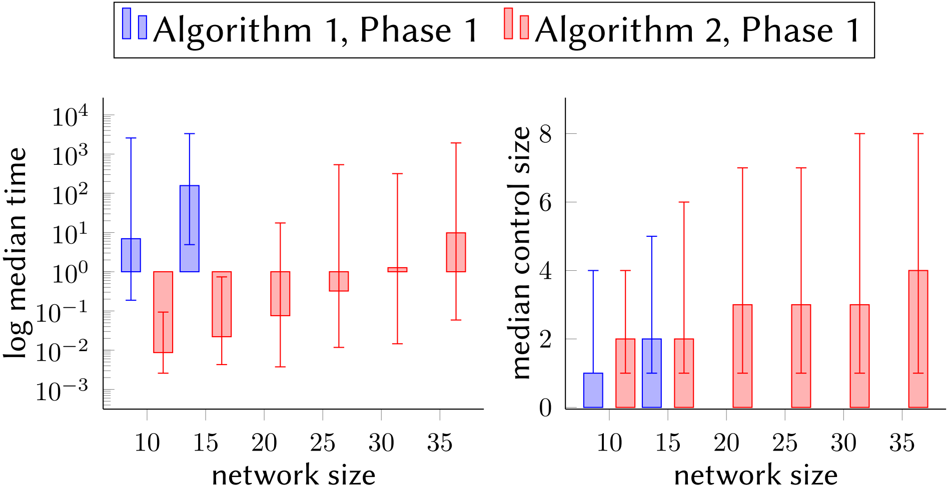
$$\Downarrow$$

Essentially consider only the subnetwork of UCV.

# Algorithm 2: Some details

- Same general scheme as in Algorithm 1.
- Use a SAT solver to infer total controls.
  - simpler and faster than one-shot inference in Algorithm 1
- Infer smaller controls from total controls.
  - usually much smaller

# Benchmarks




Batches of 100 random scale-free networks, 10–35 nodes.

Inferring control sequences is hard.

# Contributions

1. Set up a formal framework for inference of control sequences.
2. Establish practical upper bounds on the size of minimal sequences.
3. Design two algorithms for exact and approximate inference of control sequences.

# Future work

1. Further explore sequence properties.
2. Applications to gene regulatory networks.
3. Partition the variables into observable, controllable, and internal.
4. Consider the asynchronous mode.

# Sequences ∈ PSPACE-hard

$k \in \alpha 2^{|UCV|} \implies$ exhaustive exploration

Jérémie Pardo

Sergiu Ivanov

Franck Delaplace

université evry val-d'essonne

université PARIS-SACLAY

RFSI
réseau francilien
en sciences informatiques

îledeFrance