Inference of Boolean networks from gene interaction graphs using a SAT solver

David A. Rosenblueth   Stalin Muñoz   Miguel Carrillo
Eugenio Azpeitia

Universidad Nacional Autónoma de México

Outline

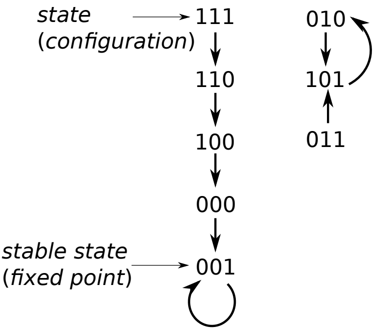
synchronous *Boolean network* (next state)

$$\overline{x} = (x_1, x_2, x_3) \qquad f(\overline{x}) = (f_1(\overline{x}), f_2(\overline{x}), f_3(\overline{x}))$$

synchronous *Boolean network* (next state)

$$\overline{x} = (x_1, x_2, x_3) \qquad f(\overline{x}) = (f_1(\overline{x}), f_2(\overline{x}), f_3(\overline{x}))$$

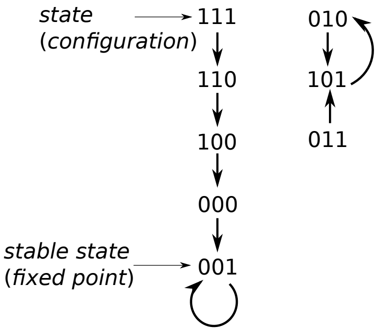
1. *state graph (dynamics)*



synchronous *Boolean network* (next state)

$$\overline{x} = (x_1, x_2, x_3) \qquad f(\overline{x}) = (f_1(\overline{x}), f_2(\overline{x}), f_3(\overline{x}))$$

1. *state graph* (*dynamics*)
2. *truth table* for next value of each bit



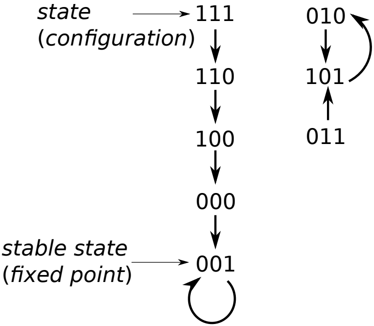
1. Griffin (prototype): Input and Output

Boolean network (synchronous Boolean network)

synchronous *Boolean network* (next state)

$$\overline{x} = (x_1, x_2, x_3) \qquad f(\overline{x}) = (f_1(\overline{x}), f_2(\overline{x}), f_3(\overline{x}))$$

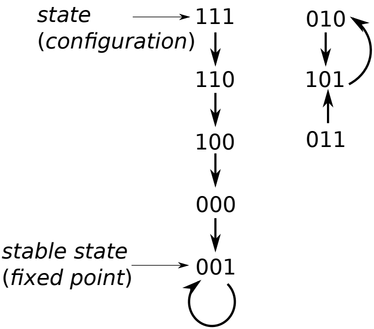
1. *state graph* (*dynamics*)
2. *truth table* for next value of each bit
3. *Boolean formula* for next value of each bit



synchronous *Boolean network* (next state)

$$\overline{x} = (x_1, x_2, x_3) \qquad f(\overline{x}) = (f_1(\overline{x}), f_2(\overline{x}), f_3(\overline{x}))$$

1. *state graph* (*dynamics*)
2. *truth table* for next value of each bit
3. *Boolean formula* for next value of each bit

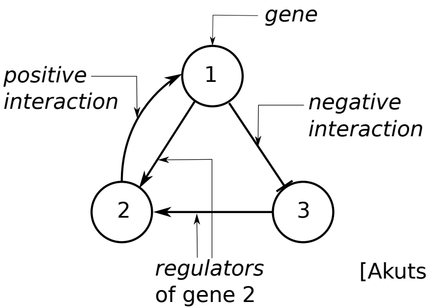


- gene regulatory networks
- pathway signaling networks
- cell cycles

1. Griffin: Input and Output

Interaction graph (topology)

Interaction graph



[Akutsu et al. 1999]

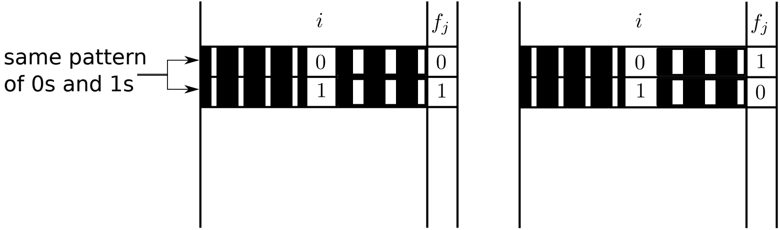
meaning of interaction?

interaction graph, underspecification

1. Griffin: Input and Output

Meaning of positive and negative interactions

positive $i \rightarrow j$ | *negative* $i \dashv j$



truth table for gene j | truth table for gene j

possible to have both positive and negative regulation (*ambiguous*)

1. Griffin: Input and Output

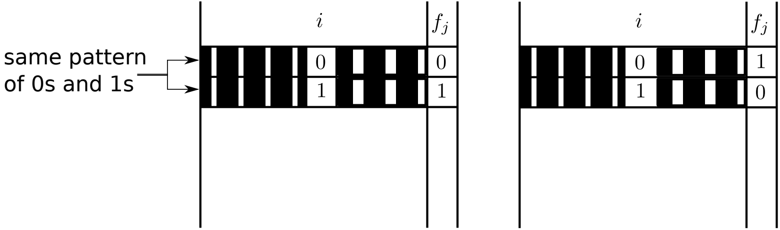
Meaning of positive and negative interactions

positive

$i \to j$

negative

$i \dashv j$

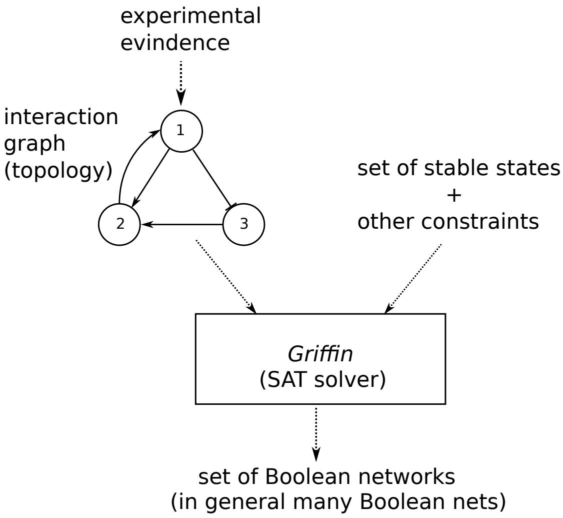


truth table for gene j truth table for gene j

possible to have both positive and negative regulation (*ambiguous*)

$$i \to j \text{ iff } \exists \overline{x}. f_j(\overline{x}[1/x_i]) > f_j(\overline{x}[0/x_i])$$

$$i \dashv j \text{ iff } \exists \overline{x}. f_j(\overline{x}[1/x_i]) < f_j(\overline{x}[0/x_i])$$


experimental
evindence
interaction
graph
(topology)
1
2
3
set of stable states
+
other constraints
Griffin
(SAT solver)
set of Boolean networks
(in general many Boolean nets)

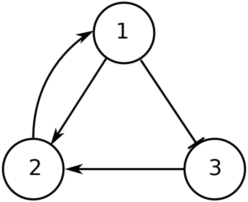
truth table of Boolean networks f represented with "entry" variables

x	$f(x)$		
0 0 0	$f_1 000$	$f_2 000$	$f_3 000$
0 0 1	$f_1 001$	$f_2 001$	$f_3 001$
0 1 0	$f_1 010$	$f_2 010$	$f_3 010$
0 1 1	$f_1 011$	$f_2 011$	$f_3 011$
1 0 0	$f_1 100$	$f_2 100$	$f_3 100$
1 0 1	$f_1 101$	$f_2 101$	$f_3 101$
1 1 0	$f_1 110$	$f_2 110$	$f_3 110$
1 1 1	$f_1 111$	$f_2 111$	$f_3 111$

2. Boolean-network inference
Step 2: Incorporation of regulators

x	$f(x)$		
0 0 0	$f_1 000$	$f_2 000$	$f_3 000$
0 0 1	$f_1 001$	$f_2 001$	$f_3 001$
0 1 0	$f_1 010$	$f_2 010$	$f_3 010$
0 1 1	$f_1 011$	$f_2 011$	$f_3 011$
1 0 0	$f_1 100$	$f_2 100$	$f_3 100$
1 0 1	$f_1 101$	$f_2 101$	$f_3 101$
1 1 0	$f_1 110$	$f_2 110$	$f_3 110$
1 1 1	$f_1 111$	$f_2 111$	$f_3 111$

interaction graph



representative:

for each column: same color = same variable

$$\mathrm{repr}(f_3 000) = \mathrm{repr}(f_3 001) = \mathrm{repr}(f_3 010) = \mathrm{repr}(f_3 011)$$

Boolean formula in *conjunctive normal form*:

$$(\ell_0 \textbf{ or } \ldots \textbf{ or } \ell_1) \textbf{ and } \ldots \textbf{ and } (\ell_{r-1} \textbf{ or } \ldots \textbf{ or } \ell_r)$$

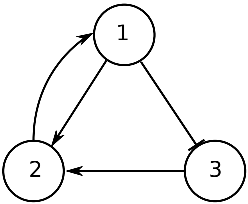
where:

$$\ell_i = \begin{cases} x_i \text{ or} \\ \textbf{not } x_i \end{cases}$$

- Find values (0, 1) for each x_i such that value of formula is 1.
- Phenomenal progress in last 15 years: millions of variables for first solution.

x	$f(x)$		
0 0 0	$f_1 000$	$f_2 000$	$f_3 000$
0 0 1	$f_1 000$	$f_2 001$	$f_3 000$
0 1 0	$f_1 010$	$f_2 000$	$f_3 000$
0 1 1	$f_1 010$	$f_2 001$	$f_3 000$
1 0 0	$f_1 000$	$f_2 100$	$f_3 100$
1 0 1	$f_1 000$	$f_2 101$	$f_3 100$
1 1 0	$f_1 010$	$f_2 100$	$f_3 100$
1 1 1	$f_1 010$	$f_2 101$	$f_3 100$

interaction graph



$2 \to 1$:

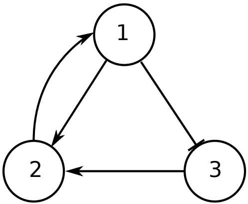
$$\varphi^{+}_{RG}(2,1) = (\textbf{not } f_1 000) \textbf{ and } f_1 010$$

2. Boolean-network inference

Interaction formulas

x	$f(x)$		
0 0 0	$f_1 000$	$f_2 000$	$f_3 000$
0 0 1	$f_1 000$	$f_2 001$	$f_3 000$
0 1 0	$f_1 010$	$f_2 000$	$f_3 000$
0 1 1	$f_1 010$	$f_2 001$	$f_3 000$
1 0 0	$f_1 000$	$f_2 100$	$f_3 100$
1 0 1	$f_1 000$	$f_2 101$	$f_3 100$
1 1 0	$f_1 010$	$f_2 100$	$f_3 100$
1 1 1	$f_1 010$	$f_2 101$	$f_3 100$

interaction graph

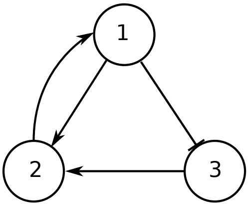


$2 \to 1$:

$$\varphi^{+}_{RG}(2,1) = (\textbf{not } f_1 000) \textbf{ and } f_1 010$$

x	$f(x)$		
0 0 0	$f_1 000$	$f_2 000$	$f_3 000$
0 0 1	$f_1 000$	$f_2 001$	$f_3 000$
0 1 0	$f_1 010$	$f_2 000$	$f_3 000$
0 1 1	$f_1 010$	$f_2 001$	$f_3 000$
1 0 0	$f_1 000$	$f_2 100$	$f_3 100$
1 0 1	$f_1 000$	$f_2 101$	$f_3 100$
1 1 0	$f_1 010$	$f_2 100$	$f_3 100$
1 1 1	$f_1 010$	$f_2 101$	$f_3 100$

interaction graph



$2 \rightarrow 1$:

$$\varphi^{+}_{RG}(2,1) = (\textbf{not}\ f_1 000)\ \textbf{and}\ f_1 010$$

$$\varphi^{+}_{R_G}(2,1) = (\textbf{not } f_1000) \textbf{ and } f_1010$$

$$\varphi^{+}_{R_G}(1,2) = ((\textbf{not } f_2000) \textbf{ and } f_2100) \\ \textbf{or } ((\textbf{not } f_2001) \textbf{ and } f_2101)$$

$$\varphi^{+}_{R_G}(3,2) = (\textbf{not } f_2000) \textbf{ and } f_2001 \\ \textbf{or } ((\textbf{not } f_2100) \textbf{ and } f_2101)$$

$$\varphi^{-}_{R_G}(1,3) = f_3000 \textbf{ and } (\textbf{not } f_3100)$$

$$\varphi_{R_G} = \varphi^{+}_{R_G}(1,2) \textbf{ and } \varphi^{+}_{R_G}(2,1) \textbf{ and } \varphi^{+}_{R_G}(3,2) \textbf{ and } \varphi^{-}_{R_G}(1,3)$$

$$\varphi^{+}_{R_G}(2,1) = (\textbf{not } \mathrm{f}_1 000) \textbf{ and } \mathrm{f}_1 010$$

$$\varphi^{+}_{R_G}(1,2) = ((\textbf{not } \mathrm{f}_2 000) \textbf{ and } \mathrm{f}_2 100) \textbf{ or } ((\textbf{not } \mathrm{f}_2 001) \textbf{ and } \mathrm{f}_2 101)$$

$$\varphi^{+}_{R_G}(3,2) = (\textbf{not } \mathrm{f}_2 000) \textbf{ and } \mathrm{f}_2 001 \textbf{ or } ((\textbf{not } \mathrm{f}_2 100) \textbf{ and } \mathrm{f}_2 101)$$

$$\varphi^{-}_{R_G}(1,3) = \mathrm{f}_3 000 \textbf{ and } (\textbf{not } \mathrm{f}_3 100)$$

$$\varphi_{R_G} = \varphi^{+}_{R_G}(1,2) \textbf{ and } \varphi^{+}_{R_G}(2,1) \textbf{ and } \varphi^{+}_{R_G}(3,2) \textbf{ and } \varphi^{-}_{R_G}(1,3)$$

symbolic method (represents the set of all desired networks as a Boolean formula)

001 is a stationary state:

$$\varphi_{\mathrm{FP}}(G,x) = (\textbf{not } \mathrm{f}_1 000) \textbf{ and } (\textbf{not } \mathrm{f}_2 001) \textbf{ and } \mathrm{f}_3 000$$

001 is a stationary state:

$$\varphi_{\mathrm{FP}}(G, x) = (\textbf{not } \mathrm{f}_1 000) \textbf{ and } (\textbf{not } \mathrm{f}_2 001) \textbf{ and } \mathrm{f}_3 000$$

001 is the only stationary state:

$$\begin{aligned}
\varphi_{\mathrm{NFP}} = \varphi_{\mathrm{FP}} \textbf{ and}\\
&\textbf{not } ((\textbf{not } \mathrm{f}_1 000) \textbf{ and } (\textbf{not } \mathrm{f}_2 000) \textbf{ and } (\textbf{not } \mathrm{f}_3 000)) \textbf{ and}\\
&\textbf{not } ((\textbf{not } \mathrm{f}_1 010) \textbf{ and } \mathrm{f}_2 000 \textbf{ and } (\textbf{not } \mathrm{f}_3 000)) \textbf{ and}\\
&\textbf{not } ((\textbf{not } \mathrm{f}_1 010) \textbf{ and } \mathrm{f}_2 001 \textbf{ and } \mathrm{f}_3 000) \textbf{ and}\\
&\textbf{not } (\mathrm{f}_1 000 \textbf{ and } (\textbf{not } \mathrm{f}_2 100) \textbf{ and } (\textbf{not } \mathrm{f}_3 100)) \textbf{ and}\\
&\ldots\\
&\textbf{not } (\mathrm{f}_1 010 \textbf{ and } \mathrm{f}_2 101 \textbf{ and } \mathrm{f}_3 100)
\end{aligned}$$

Formulas:

- exact set of stationary states (exponential in number of genes)
- possibly other (hypothetical) stationary states
- exact set of interactions
- possibly other (hypothetical) interactions
- nonambiguity (no positive and negative interactions)
- stationary states of mutations

Formulas:

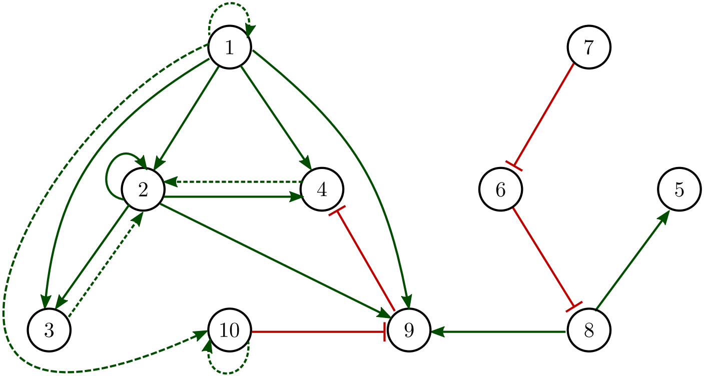
- exact set of stationary states (exponential in number of genes)
- possibly other (hypothetical) stationary states
- exact set of interactions
- possibly other (hypothetical) interactions
- nonambiguity (no positive and negative interactions)
- stationary states of mutations

For each interaction:

- one of $\{\text{mandatory}, \text{hypothetical}\}$,
- one of $\{\text{positive}, \text{negative}, \text{unknown sign}\}$,
- one of $\{\text{ambiguous}, \text{nonambiguous}\}$.

3. Practical example

Arabidopsis thaliana root stem cell niche



no mutations: By hand, 2 networks. *Griffin*, 74 networks.

mutations: By hand, 1 network. *Griffin*, 3 networks.

4. Concluding remarks
Related work

1. [Akutsu, Miyano and Kuhara, 1999]
   - entry variables
   - input: set of "state/next-state" pairs ($\sim$ time series)
   - output: Boolean networks
   - exhaustive search (nonsymbolic)

2. [Azpeitia, Weinstein, Benítez, Mendoza and Alvarez-Buylla, 2013]
   - input: interaction graph (topology)
   - output: Boolean networks
   - search with random component (nonsymbolic)

4. Concluding remarks

Boolean-network inference and symbolic methods

1. interaction graph (topology) → Boolean network is important
    1. root stem cell niche *A. thaliana*
    2. flower *A. thaliana*
    3. yeast cell cycle
    4. mammalian cell cycle
    5. T-cell receptor signaling
    6. T-cell differentiation
    7. vulva *C. elegans*
2. search space is *vast*
3. only possible with symbolic methods (SAT solvers, binary decision diagrams (BDDs))

Future work

1. multi-valued genes
2. more efficient method for exact set of stable states
3. asynchronous networks

Summary

1. Boolean-network inference from interaction graph is important
2. search space is vast
3. forthcoming networks will be large
4. symbolic methods (representing set of desired solutions as Boolean formula)
5. we presented a method for symbolically finding synchronous Boolean networks having a number of properties
6. we wish to extend our method to:
    1. multi-valued genes
    2. asynchronous networks