Reduction of Boolean models

Matthew Macauley

Department of Mathematical Sciences
Clemson University
http://www.math.clemson.edu/~macaule/

Algebraic Systems Biology

Motivating example: Th-cell differentiation

White blood cells or *leukocytes* are in the immune system and fight diseases and infections.

One subtype are the *lymphocytes*, which includes the natural killer (NK) cells, B cells, and T cells, all which have different cellular functions.

The T-cells circulate throughout our bodies in the lymph fluid, looking for cellular abnormalities, infections, and diseases.

Helper T-cells (Th-cells) are a certain type of T-cells. They begin as *naïve*, or *Th0 cells*, and then differentiate into one of two phenotypes:
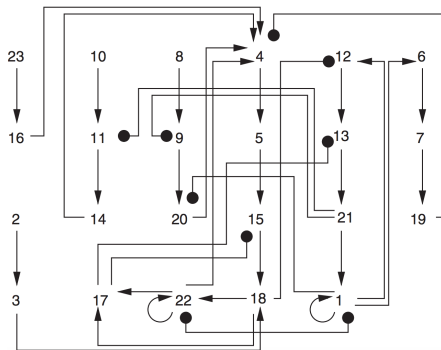
1. Type 1 are the *Th1 cells* which fight intracellular bacteria and protozoa.
2. Type 2 are the *Th2 cells* which fight extracellular parasites.

Malfunctions of immune responses involving Th1 phenotypes can result in autoimmune diseases, whereas malfunctions involving Th2 phenotypes can result in allergic reactions.

The biochemical signals that determine Th1 and Th2 differentiation act as a bistable switch, which permits either GATA3 or T-bet to be expressed, but not both. This was modeled using a 23-node Boolean network in Mendoza et. al (2006).

Boolean model of Th-cell differentiation (Mendoza, 2006)

$x_1 =$ GATA3 $f_1 = (x_1 \vee x_{21}) \wedge \overline{x_{22}}$

$x_2 =$ IFN-β $f_2 = 0$

$x_3 =$ IFN-βR $f_3 = x_2$

$x_4 =$ IFN-γ $f_4 = (x_{14} \vee x_{16} \vee x_{20} \vee x_{22}) \wedge \overline{x_{19}}$

$x_5 =$ IFN-γR $f_5 = x_4$

$x_6 =$ IL-10 $f_6 = x_1$

$x_7 =$ IL-10R $f_7 = x_6$

$x_8 =$ IL-12 $f_8 = 0$

$x_9 =$ IL-12R $f_9 = x_8 \wedge \overline{x_{21}}$

$x_{10} =$ IL-18 $f_{10} = 0$

$x_{11} =$ IL-18R $f_{11} = x_{10} \wedge \overline{x_{21}}$

$x_{12} =$ IL-4 $f_{12} = x_1 \wedge \overline{x_{18}}$

$x_{13} =$ IL-4R $f_{13} = x_{12} \wedge \overline{x_{17}}$

$x_{14} =$ IRAK $f_{14} = x_{11}$

$x_{15} =$ JAK1 $f_{15} = x_5 \wedge \overline{x_{17}}$

$x_{16} =$ NFAT $f_{16} = x_{23}$

$x_{17} =$ SOCS1 $f_{17} = x_{18} \vee x_{22}$

$x_{18} =$ STAT1 $f_{18} = x_3 \vee x_{15}$

$x_{19} =$ STAT3 $f_{19} = x_7$

$x_{20} =$ STAT4 $f_{20} = x_9 \wedge \overline{x_1}$

$x_{21} =$ STAT6 $f_{21} = x_{13}$

$x_{22} =$ T-bet $f_{22} = (x_{18} \vee x_{22}) \wedge \overline{x_1}$

$x_{23} =$ TCR $f_{23} = 0$

Analyzing the T helper cell model

The 23-node Th-cell Boolean model has a state space of size $2^{23} = 8,388,608$ nodes.

A Boolean model of the segment polarity genes in *Drosophila melanogaster* has 60 nodes, and a state space of size $2^{60} \approx 1.15 \times 10^{18}$.

In the previous lecture, we modeled time-delays and dilution & degradation by adding a number of Booleans variables.

This can causes the state space to grow enormously, though in many cases, this shouldn't affect the qualitative nature of the dynamics.

In this lecture, we'll see how large Boolean models can be "reduced" to much smaller models in a way that preserves the number of fixed points.

Other methods have been developed for asynchronous Boolean or logical models that preserve more general attractors.

A motivating example

Toy model of the *lac* operon

$f_M = \overline{R}$	R inhibits mRNA transcription
$f_P = M$	P is translated from *lac* mRNA
$f_B = M$	B is translated from *lac* mRNA
$f_R = \overline{A}$	A inactivates the repressor protein
$f_A = L \wedge B$	A is synthesized by lactose and β-galactosidase
$f_L = P$	*lac* permease transports lactose into the cell

Here is the wiring diagram:
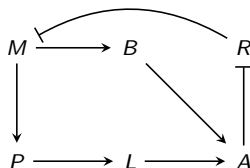
We won't show the state space because it's large (64 nodes), but it has two fixed points, both of which are biologically reasonable:
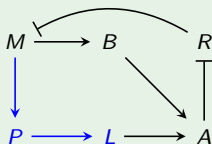
$$(M, P, B, R, A, L) = (0, 0, 0, 1, 0, 0) \qquad \text{and} \qquad (1, 1, 1, 0, 1, 1).$$

Our goal is to "reduce" this model in a way that in some senes, *preserves the fixed points*.

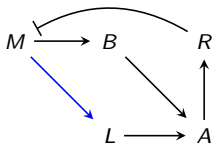# A motivating example (cont.)

Let's eliminate P

$$f_M = \overline{R}$$
$$f_P = M$$
$$f_B = M$$
$$f_R = \overline{A}$$
$$f_A = L \wedge B$$
$$f_L = P$$

At equilibrium, $P(t) = P(t+1) = f_P(x(t)) = M(t)$.

Thus, we can replace every instance of P with M:

$$f_M = \overline{R}$$
$$\cancel{f_P = M}$$
$$f_B = M$$
$$f_R = \overline{A}$$
$$f_A = L \wedge B$$
$$f_L = M$$
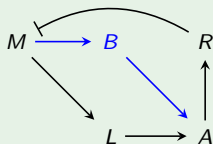

This reduced network has two fixed points: $(M, B, R, A, L) = (0, 0, 1, 0, 0)$, $(1, 1, 0, 1, 1)$.

Since $P = M$, we can recover the fixed points of the original network.
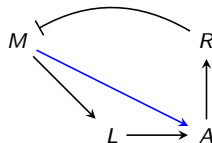
A motivating example (cont.)

Let's eliminate B

$$f_M = \overline{R}$$
$$f_B = M$$
$$f_R = \overline{A}$$
$$f_A = L \wedge B$$
$$f_L = M$$

At equilibrium, $B = f_B = M$, so we can replace every instance of B with M.

Thus, we can replace every instance of P with M:

$$f_M = \overline{R}$$
$$\cancel{f_B = M}$$
$$f_R = \overline{A}$$
$$f_A = L \wedge M$$
$$f_L = M$$

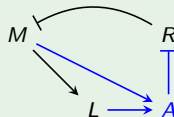This reduced network has two fixed points: $(M, R, A, L) = (0, 1, 0, 0)$, $(1, 0, 1, 1)$.

Since $B = P = M$, we can recover the fixed points of the original network.

$$(M, P, B, R, A, L) = (M, M, M, \overline{A}, L \wedge M, M) = (0, 0, 0, 1, 0, 0), \quad \text{and} \quad (1, 1, 1, 0, 1, 1).$$
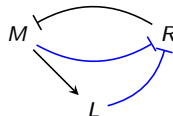
A motivating example (cont.)

Let's eliminate A

$f_M = \overline{R}$
$f_R = \overline{A}$
$f_A = L \wedge M$
$f_L = M$

At equilibrium, $A = f_A = L \wedge M$, so we can replace every instance of A with $L \wedge M$:

$f_M = \overline{R}$
$f_R = \overline{L \wedge M} = \overline{L} \vee \overline{M}$
$\cancel{f_A = L \wedge M}$
$f_L = M$

There are two fixed points of this reduced network: $(M, R, L) = (0, 1, 0),\ (1, 0, 1)$.

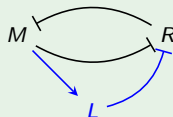Since $B = P = M$, $A = L \wedge M$, we can recover the fixed points of the original network by *back-substituting*.

$$(M, P, B, R, A, L) = (M, M, M, R, L \wedge M, L) = (0, 0, 0, 1, 0, 0), \quad \text{and} \quad (1, 1, 1, 0, 1, 1).$$

A motivating example (cont.)

Let's eliminate L

$$f_M = \overline{R}$$
$$f_R = \overline{L} \vee \overline{M}$$
$$f_L = M$$

At equilibrium, $L = f_L = M$, so we can replace every instance of L with M:

$$f_M = \overline{R}$$
$$f_R = \overline{M} \vee \overline{M} = \overline{M}$$
$$\cancel{f_L = M}$$

There are two fixed points of this reduced network $(M, R) = (0, 1)$, $(1, 0)$.

Since $L = B = P = M$ and $A = L \wedge M = M$, we can recover the steady-states of the original network by *back-substituting*.

$$(M, P, B, R, A, L) = (M, M, M, R, M, M) = (0, 0, 0, 1, 0, 0), \quad \text{and} \quad (1, 1, 1, 0, 1, 1).$$

A motivating example (cont.)

Let's eliminate R

$$f_M = \overline{R}$$
$$f_R = \overline{M}$$

$M \rightharpoonup R$ (bidirectional arc between M and R)

At equilibrium, $R = f_R = \overline{M}$, so we can replace every instance of R with $\overline{M}$:

$$f_M = \overline{\overline{M}} = M$$
$$\cancel{f_R = \overline{N}}$$

M (with self-loop)

There are two fixed points of this reduced network $M = 0, 1$.

Since $L = B = P = M$ and $A = L \wedge M = M$, we can recover the steady-states of the original network by *back-substituting*.

$$(M, P, B, R, A, L) = (M, M, M, \overline{M}, M, M) = (0, 0, 0, 1, 0, 0), \quad \text{and} \quad (1, 1, 1, 0, 1, 1).$$

General reduction

Reduction steps

1. Simplify the Boolean functions and wiring diagram.

   1.1 Reduce / simplfy Boolean expressions using Boolean algebra.

   1.2 Remove unnecessary edges from the wiring diagram.

2. Delete vertices x_i with no self-loop (equivalently, f_{x_i} doesn't depend on x_i), by doing the following:

   2.1 For all vertices y such that $x_i \longrightarrow y$, substitute f_{x_i} into x_i:

   $$f_y(x_1 \ldots, \underbrace{\cdots x_i \cdots}_{\text{pos. } x_i}, \ldots, x_n) \qquad \text{becomes} \qquad f_y(x_1 \ldots, \underbrace{\cdots f_{x_i} \cdots}_{\text{pos. } x_i}, \ldots, x_n).$$

   2.2 Replace edges $v \longrightarrow x_i \longrightarrow y$ by $v \longrightarrow y$ and remove x_i (and all edges to/from x_i).
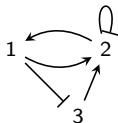
Exercise

In Step 2.2 above, how should you replace replace edges of the form:

- $v \longrightarrow x_i \longrightarrow\!\mid y$

- $v \longrightarrow\!\mid x_i \longrightarrow y$

- $v \longrightarrow\!\mid x_i \longrightarrow\!\mid y$

General reduction: an example

Consider the Boolean network $f(x) = (x_2, (x_1 \wedge x_3) \vee \overline{x_2}, \overline{x_1})$.

Let's remove $x_3 = \overline{x_1}$. The new Boolean functions are

$$h_1(x_1, x_2) = f_1(x_1, x_2, x_3) = f_1(x_1, x_2, \overline{x_1}) = x_2 \,,$$
$$h_2(x_1, x_2) = f_2(x_1, x_2, x_3) = f_2(x_1, x_2, \overline{x_1}) = (x_1 \wedge \overline{x_1}) \vee \overline{x_2}$$

However, $x_1 \wedge \overline{x_1} = 0$, and so

$$h_2(x_1, x_2) = (x_1 \wedge \overline{x_1}) \vee \overline{x_2} = 0 \vee \overline{x_2} = \overline{x_2} \,.$$

The reduced Boolean network is thus $h(x_1, x_2) = (x_2, \overline{x_2})$

To find the fixed points, we must solve the system $h_i = x_i$ for $i = 1, 2$:
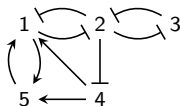
$$\begin{cases} h_1(x_1, x_2) = x_2 = x_1 \\ h_2(x_1, x_2) = \overline{x_2} = x_2 \,. \end{cases}$$

Since $x_2 \neq \overline{x_2}$, there are no fixed points in the reduced BN, and thus none in the original BN.
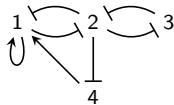
General reduction: an example

Consider the Boolean network:

$$f = (x_5 \vee \overline{x_2} \vee x_4, \ \overline{x_1} \wedge \overline{x_3}, \ \overline{x_2}, \ \overline{x_2}, \ x_1 \vee x_4).$$
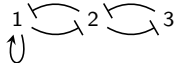

Remove $x_5 = x_1 \vee x_4$:

$$f = ((x_1 \vee x_4) \vee \overline{x_2} \vee x_4, \ \overline{x_1} \wedge \overline{x_3}, \ \overline{x_2}, \ \overline{x_2})$$
$$= (x_1 \vee \overline{x_2} \vee x_4, \ \overline{x_1} \wedge \overline{x_3}, \ \overline{x_2}, \ \overline{x_2}).$$

Remove $x_4 = \overline{x_2}$:

$$f = (x_1 \vee \overline{x_2} \vee \overline{x_2}, \ \overline{x_1} \wedge \overline{x_3}, \ \overline{x_2}) = (x_1 \vee \overline{x_2}, \ \overline{x_1} \wedge \overline{x_3}, \ \overline{x_2})$$

Remove $x_3 = \overline{x_2}$:

$$f = (x_1 \vee \overline{x_2}, \ \overline{x_1} \wedge \overline{\overline{x_2}}) = (x_1 \vee \overline{x_2}, \ \overline{x_1} \wedge x_2)$$

This yields the system:

$$\begin{cases} h_1(x_1, x_2) = x_1 \vee \overline{x_2} \\ h_2(x_1, x_2) = \overline{x_1} \wedge x_2 \\ x_3 = \overline{x_2} \\ x_4 = \overline{x_2} \\ x_5 = x_1 \vee x_4 \end{cases}$$

The reduced system (h_1, h_2) has 2 fixed points:

$$(x_1, x_2) = (1, 0), \ (0, 1).$$

Thus, the original system has two fixed points:

$$(x_1, x_2, x_3, x_4, x_5) = (1, 0, 1, 1, 1), \ (0, 1, 0, 0, 0).$$

Boolean model reduction in Macaulay2

We want polynomials in variables $x_1, \ldots, x_5$, over the field $\mathbb{F}_2$, and $x_i^2 = x_i$:

```
R = ZZ/2[x1,x2,x3,x4,x5] / ideal(x1^2-x1, x2^2-x2, x3^2-x3, x4^2-x4, x5^2-x5);
```

For convenience, let's define $a|b := a + b + ab$ and $a\&b := a * b$:

```
RingElement | RingElement :=(x,y)->x+y+x*y;
RingElement & RingElement :=(x,y)->x*y;
```

Input the Boolean model $f = (f_1, f_2, f_3, f_4, f_5) = (x_5 \vee \overline{x_2} \vee x_4, \; \overline{x_1} \wedge \overline{x_3}, \; \overline{x_2}, \; \overline{x_2}, \; x_1 \vee x_4)$:

```
f1 = x5 | (1+x2) | x4;
f2 = (1+x1) & (1+x3);
f3 = 1+x2;
f4 = 1+x2;
f5 = x1 | x4;
```

Typing (f1,f2,f3,f4,f5) gives the following output:

```
(x2x4x5+x2x4+x2x5+x2+1, x1x3+x1+x3+1, x2+1, x2+1, x1x4+x1+x4)
```

Boolean model reduction in Macaulay2

Let's eliminate $x_5 = x_1 \lor x_4$, which appears in the function f_1.

```
f1=sub(f1,{x5=>f5});
```

Now, typing (f1,f2,f3,f4) gives the following output:

```
(x1x2x4+x1x2+x2x4+x2+1, x1x3+x1+x3+1, x2+1, x2+1)
```

Let's eliminate $x_4 = \overline{x_2}$, which appears in the function f_1.

```
f1=sub(f1,{x4=>f4});
```

Typing (f1,f2,f3) gives the following output:

```
(x1x2+x2+1, x1x3+x1+x3+1, x2+1)
```

Boolean model reduction in Macaulay2

Let's eliminate $x_3 = \overline{x_2}$, which appears in the function f_2.

```
f2=sub(f1,{x3=>f3});
```

Typing (f1,f2) gives the following output:

```
(x1x2+x2+1, x1x2+x2+1)
```

gens gb ideal(f1+x2, f2+x2) gives the following output:

```
(x1+x2+1)
```

which means that $x_1 = \overline{x_2}$.

Back-substituting $x_3 = \overline{x_2}$, and $x_4 = \overline{x_2}$, and $x_5 = x_1 \vee x_4 = \overline{x_2} \vee \overline{x_2} = \overline{x_2}$ gives

$$(x_1, x_2, x_3, x_4, x_5) = (\overline{x_2}, x_2, \overline{x_2}, \overline{x_2}, \overline{x_2}),$$
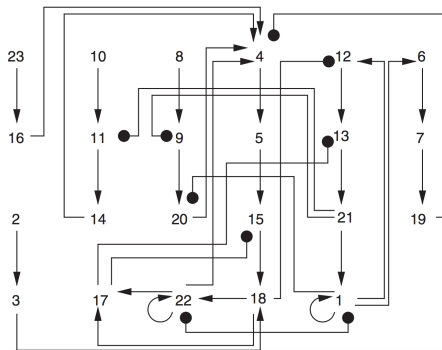
and thus we have two fixed points of our original Boolean model:

$$(x_1, x_2, x_3, x_4, x_5) = (1, 0, 1, 1, 1) \quad \text{and} \quad (0, 1, 0, 0, 0).$$

Finding the fixed points of the T helper cell model

$x_1 = $ GATA3 $f_1 = (x_1 \lor x_{21}) \land \overline{x_{22}}$
$x_2 = $ IFN-β $f_2 = 0$
$x_3 = $ IFN-βR $f_3 = x_2$
$x_4 = $ IFN-γ $f_4 = (x_{14} \lor x_{16} \lor x_{20} \lor x_{22}) \land \overline{x_{19}}$
$x_5 = $ IFN-γR $f_5 = x_4$
$x_6 = $ IL-10 $f_6 = x_1$
$x_7 = $ IL-10R $f_7 = x_6$
$x_8 = $ IL-12 $f_8 = 0$
$x_9 = $ IL-12R $f_9 = x_8 \land \overline{x_{21}}$
$x_{10} = $ IL-18 $f_{10} = 0$
$x_{11} = $ IL-18R $f_{11} = x_{10} \land \overline{x_{21}}$
$x_{12} = $ IL-4 $f_{12} = x_1 \land \overline{x_{18}}$
$x_{13} = $ IL-4R $f_{13} = x_{12} \land \overline{x_{17}}$
$x_{14} = $ IRAK $f_{14} = x_{11}$
$x_{15} = $ JAK1 $f_{15} = x_5 \land \overline{x_{17}}$
$x_{16} = $ NFAT $f_{16} = x_{23}$
$x_{17} = $ SOCS1 $f_{17} = x_{18} \lor x_{22}$
$x_{18} = $ STAT1 $f_{18} = x_3 \lor x_{15}$
$x_{19} = $ STAT3 $f_{19} = x_7$
$x_{20} = $ STAT4 $f_{20} = x_9 \land \overline{x_1}$
$x_{21} = $ STAT6 $f_{21} = x_{13}$
$x_{22} = $ T-bet $f_{22} = (x_{18} \lor x_{22}) \land \overline{x_1}$
$x_{23} = $ TCR $f_{23} = 0$

Reducing the T helper cell model
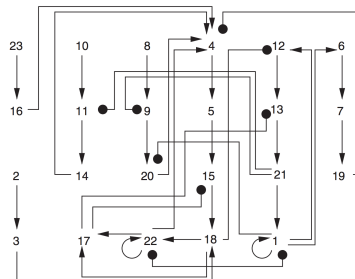
If we eliminate the variables in the order, $x_{23}, x_{21}, x_{20}, \ldots$, we get

Variable	Boolean function	Polynomial function
x_1 = GATA3	$h_1(x_1, x_{22}) = x_1 \wedge \overline{x_{22}}$	$h_1(x_1, x_{22}) = x_1 x_{22} + x_1$
x_{22} = T-bet	$h_{22}(x_1, x_{22}) = \overline{x_1} \wedge x_{22}$	$h_{22} = x_1 x_{22} + x_{22}$

There are three fixed points:

- $(0, 0)$: GATA3 and T-bet are inactive, the "signature" of Th0 cells.
- $(0, 1)$: Only T-bet is active, the signature of Th1 cells.
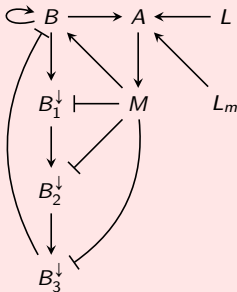- $(1, 0)$: Only GATA3 is active, the signature of Th2-cells.

Application: Modeling time delays and degradration & dilution

In the last lecture, we saw how to add Boolean variables to model time delays and loss of concentration due to degradation / dilution.

Consider the following model of the *lac* operon (slightly modified from last lecture) that assumes that β-galactosidase takes several time-steps to degrade.

Example model

$$
\begin{cases}
f_M = A \\
f_A = (B \wedge L_m) \vee L \\
f_B = M \vee (B \wedge \overline{B}_3^{\downarrow}) \\
f_{B_1^{\downarrow}} = \overline{M} \wedge B \\
f_{B_2^{\downarrow}} = \overline{M} \wedge B_1^{\downarrow} \\
f_{B_3^{\downarrow}} = \overline{M} \wedge B_2^{\downarrow}
\end{cases}
$$

Do you see why the precise number of $B_i^{\downarrow}$ variables is unimportant, regarding the number and qualitative nature of the fixed points?

Other reduction methods

The reduction methods in this lectures is from A. Veliz-Cuba (2011), and the subject of Chapter 6 in Robeva (2015).

In several papers (2013, 2015), Veliz-Cuba et al. reduce Boolean networks by transforming them to larger AND-NOT networks, where methods become easier.

In 2013, a reduction method for asynchronous Boolean networks that preserves attractors (not just fixed points) was done by R. Albert et al. in several papers.

A reduction method for (asynchronous) logical networks was done by A. Naldi et al. in 2009.

A 2023 paper by G.A. Argyris shows to to reduce Boolean networks with "backwards equivalence."
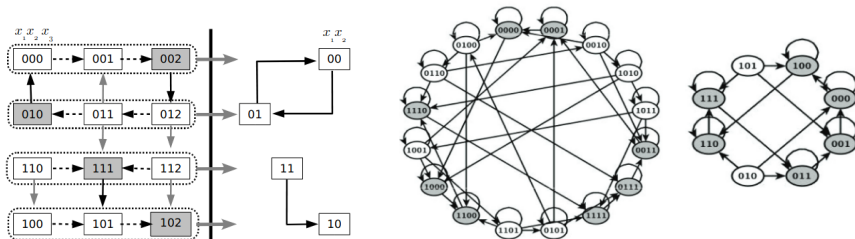
References

- A. Veliz-Cuba and D. Murrugarra. Steady State Analysis of Boolean Models: A Dimension Reduction Approach. Chapter 6 of *Algebraic and Discrete Mathematical Methods for Modern Biology*. R. Robeva (ed.). Academic Press, pp. 121-139, 2015.

- A. Veliz-Cuba. Reduction of Boolean network models. J. Theor. Biol. **289**:167–172, 2011.

- A. Veliz-Cuba, K. Buschur, R. Hamershock, A. Kniss, E. Wolff, and R. Laubenbacher. AND-NOT logic framework for steady state analysis of Boolean network models. *Appl. Math. Inform. Sci.* **7**(4):1263–74, 2013.

- A. Veliz-Cuba, R. Laubenbacher, and B. Aguilar. Dimension reduction of large sparse AND-NOT network models. Electron. Notes Theor. Comput. Sci. **316**:83–95, 2015.

- A. Naldi, E. Remy, D. Thieffry, and C. Chaouiya. A reduction of logical regulatory graphs preserving essential dynamical properties. In *Comput. Methods Syst. Biol.*: 7th Intl. Conf., CMSB 2009, Bologna, Italy. Proceedings 7. Springer Berlin Heidelberg, 2009.

- A. Veliz-Cuba, B. Aguilar, F. Hinkelmann, R. Laubenbacher. Steady state analysis of Boolean molecular network models via model reduction and computational algebra. *BMC Bioinformatics* **15**:1–8, 2014.

- J.G.T. Zañudo and R. Albert. An effective network reduction approach to find the dynamical repertoire of discrete dynamic networks. *Chaos* **23**(2), 2013.

- A. Saadatpour, R. Albert, and T.C. Reluga. A reduction method for Boolean network models proven to conserve attractors. *SIAM J. Appl. Dyn. Syst.* **12**(4):1997–2011, 2013.

- G.A. Argyris, A.L. Lafuente, A. Tribastone, M. Tschaikowski, and A. Vandin. Reducing Boolean networks with backward equivalence. *BMC Bioinformatics* **24**(212), 20 pp, 2023.