Algebraic Biology: theory and applications

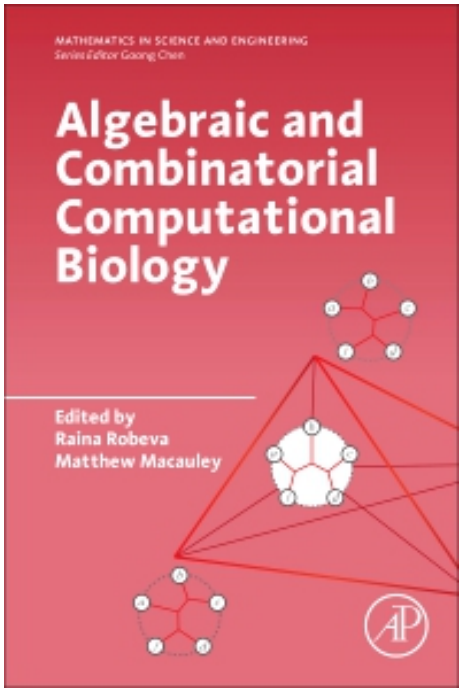
Matthew Macauley

School of Mathematical & Statistical Sciences
Clemson University
http://www.math.clemson.edu/~macaule/

January 2020

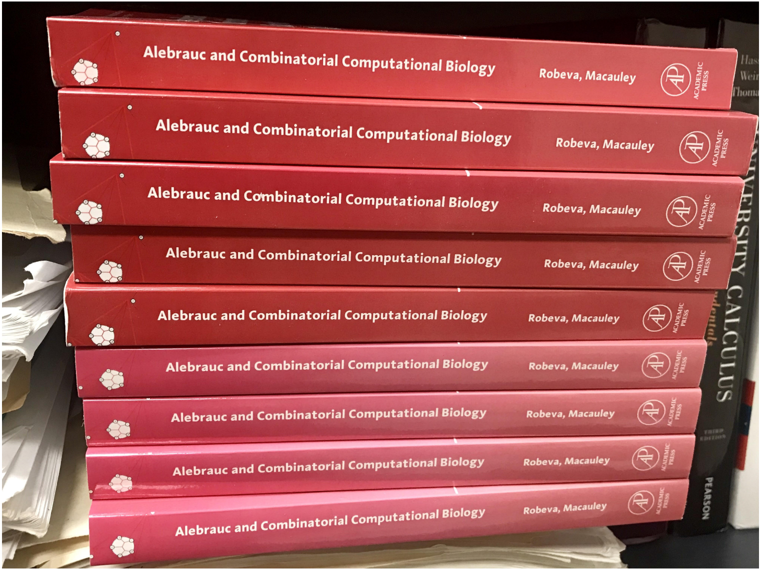
SIMONS FOUNDATION

Algebraic and Combinatorial Computational Biology



1. Multiscale graph-theoretic modeling of biomolecular structures. (Jungck, D. Knisley, Pangborn, Riehl, Wiesner)
2. Tile-based DNA nanostructures: mathematical design and problem encoding. (Ellis-Monaghan, Jonoska, Pangborn)
3. Graphis associated with DNA rearrangements and their polynomials. (Brijder, Hoogeboom, Jonoska, Saito)
4. The regulation of gene expression by operons and the local modeling framework. (Macauley, Jenkins, Davies)
5. Modeling the stochastic nature of gene regulation with Boolean networks. (Murrugarra, Aguilar)
6. Inferring interactions in molecular networks via primary decompositions of monomial ideals. (Macauley, Stigler)
7. Analysis of combinatorial neural codes: an algebraic approach. (Youngs, Curto, Veliz-Cuba)
8. Predicting neural network dynamics: insights from graph theory. (Morrison, Curto)
9. Multistationarity in biochemical networks: Results, analysis, & examples. (Conradi, Pantea)
10. Optimization problems in phylogenetics: Polytopes, programming and interpretation. (Hamerlinck, Forcey, Sands)
11. Clustering via self-organizing maps on biology and medicine. (Akman, Comar, Hrozencik, Gonzalez)
12. Toward revealing protein function: Identifying biologically relevant clusters with graph spectral methods. (Davies, Ghosh-Dastidar, J. Knisley and Samyono)

Alebrauc?


Alebrauc and Combinatorial Computational Biology
Robeva, Macauley
ACADEMIC PRESS
Alebrauc and Combinatorial Computational Biology
Robeva, Macauley
ACADEMIC PRESS
Alebrauc and Combinatorial Computational Biology
Robeva, Macauley
ACADEMIC PRESS
Alebrauc and Combinatorial Computational Biology
Robeva, Macauley
ACADEMIC PRESS
Alebrauc and Combinatorial Computational Biology
Robeva, Macauley
ACADEMIC PRESS
Alebrauc and Combinatorial Computational Biology
Robeva, Macauley
ACADEMIC PRESS
Alebrauc and Combinatorial Computational Biology
Robeva, Macauley
ACADEMIC PRESS
Alebrauc and Combinatorial Computational Biology
Robeva, Macauley
ACADEMIC PRESS
Alebrauc and Combinatorial Computational Biology
Robeva, Macauley
ACADEMIC PRESS
CALCULUS
PEARSON

Local models

Let $\mathbb{F}$ be a field of order $q = p^k$, $R = \mathbb{F}[x_1, \ldots, x_n]$, and $I = \langle x_1^q - x_1, \ldots, x_n^q - x \rangle$.

Definition

A local model over $\mathbb{F}$ is an n-tuple of functions $f = (f_1, \ldots, f_n)$, where each $f_i : \mathbb{F}^n \to \mathbb{F}$.

Remarks

Every local model $f = (f_1, \ldots, f_n)$ over $\mathbb{F}$...

1. can be associated with a unique element in $(R/I) \times \cdots \times (R/I)$.
2. defines a finite dynamical system (FDS), by iterating the map

$$f : \mathbb{F}^n \longrightarrow \mathbb{F}^n, \qquad x = (x_1, \ldots, x_n) \longmapsto \big(f_1(x), \ldots, f_n(x)\big).$$

3. has a unique asynchronous automata: the digraph with vertex set $\mathbb{F}^n$ and edges

$$E = \big\{(x, F_i(x)) \mid i = 1, \ldots, n;\ x \in \mathbb{F}^n\big\}.$$

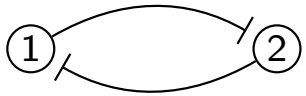
4. defines a wiring diagram.

If $|\mathbb{F}| = q$, then the number of items in (1), (2), (3) are all counted by $q^{(nq^n)}$.

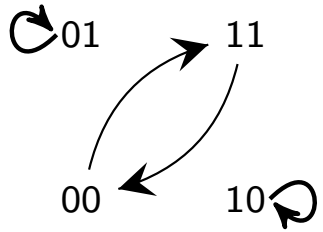
Examples: synchronous vs. asynchronous

$$\begin{cases} f_1(x_1, x_2) = \overline{x_2} \\ f_2(x_1, x_2) = \overline{x_1} \end{cases}$$

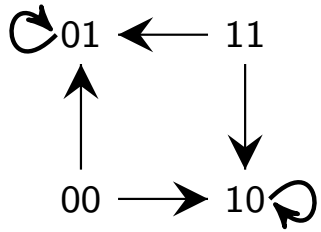
Functions



Wiring diagram



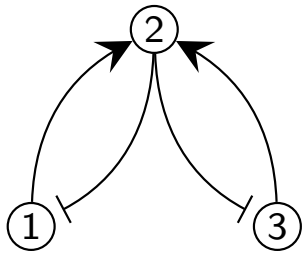
FDS map



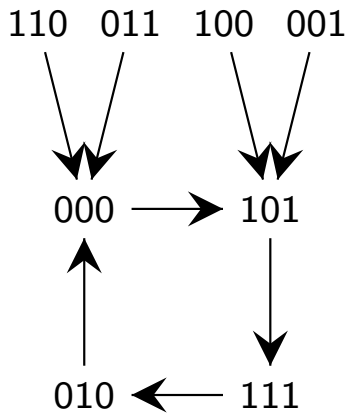
Asynchronous automata

$$\begin{cases} f_1 = \overline{x_2} \\ f_2 = x_1 \wedge x_3 \\ f_3 = \overline{x_2} \end{cases}$$

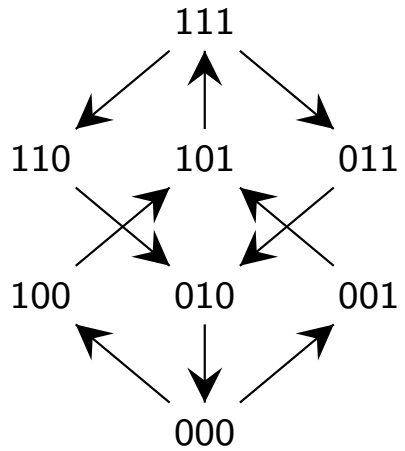
Functions



Wiring diagram



FDS map



Asynchronous automata (self-loops omitted)

Remarks

- The 2-cycle in the 1st FDS map is an "artifact of synchrony."
- In the 2nd asynchronous automata, there is a directed path between any two nodes.

Local models over general finite fields: synchronous vs. asynchronous

Recall: $\mathbb{F}$ is a finite field of order $q = p^k$, and

$$R/I = \mathbb{F}[x_1, \ldots, x_n]/\langle x_1^q - x_1, \ldots, x_n^q - x_n \rangle.$$

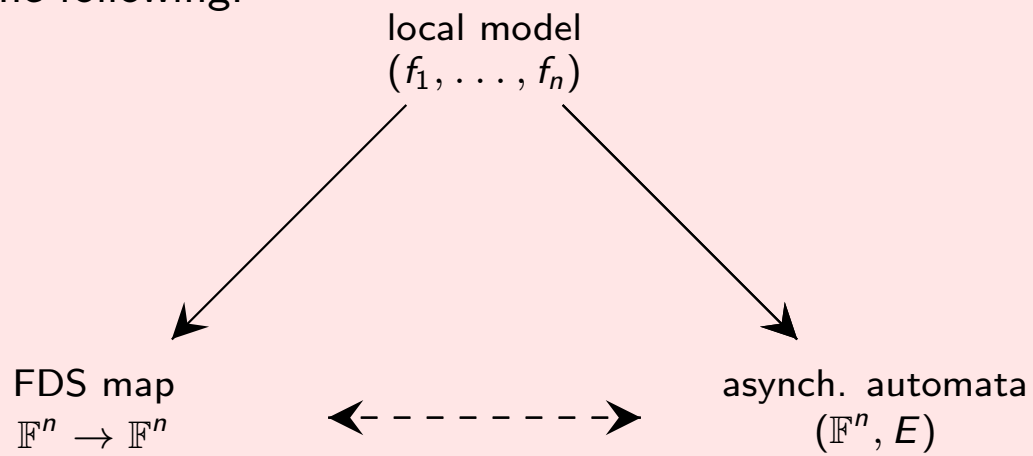
Summary

There are bijections between the following sets of cardinality $q^{(nq^n)}$:

- local models $(f_1, \ldots, f_n)$ over $\mathbb{F}$, i.e., elements of $(R/I)^n$
- FDS maps, $\mathbb{F}^n \to \mathbb{F}^n$;
- asynchronous automata: a digraph $G = (\mathbb{F}^n, E)$ with the "local property".

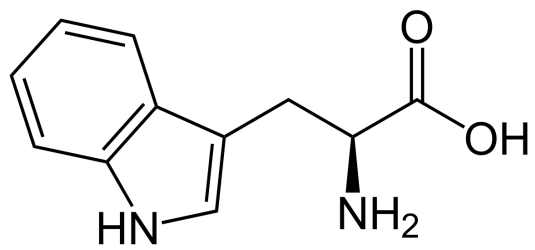
Open-ended question

Better understand the following:



Forward engineering: tryptophan synthesis and metabolism

Tryptophan (**W**) is one of the 21 amino acids that make up building blocks for proteins.



Humans are unable to synthesize it, so it must be obtained from their diets.

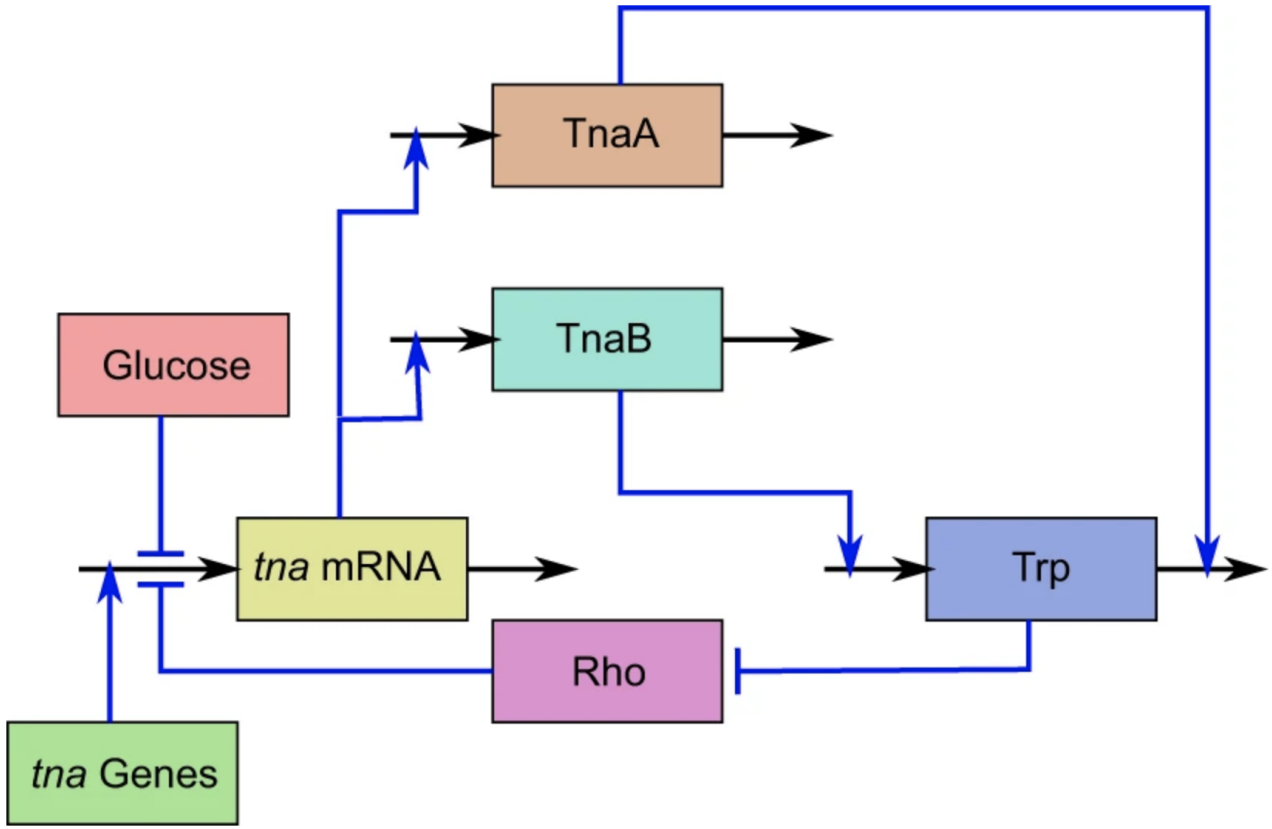
E. coli can synthesize it, via a repressible *trp operon*.

	Inducible	**Repressible**
Negative Control	**Lac operon** – remove a repressor to derepress transcription	**Trp operon** – add a corepressor to repress transcription
Positive Control	**Ara operon** – add an inducer to activate transcription	**???** – add an inhibitor to deactivate transcription

It is then metabolized by the tryptophanase (*tna*) operon.

The *tna* network

The tna operon codes for the proteins needed to metabolize tryptophan and use it as a carbon source in the absense of glucose.



An ODE model of tryptophan metabolism (Orozco-Gómez, 2019)

$$A' = k_A P_G(G_e) P_W(W) - (\gamma_A + \mu)A$$
$$B' = k_B P_G(G_e) P_W(W) - (\gamma_B + \mu)B$$
$$W' = (\alpha + \beta B)W_e - (\delta + \epsilon A P_A(G_e, W_e) + \mu)W$$

Variables:

- $A(t)$: concentration of TnaA protein
- $B(t)$: concentration of TnaB protein
- $W(t)$: concentration of intracellular tryptophan

Parameters:

- W_e: concentration of extracellular tryptophan
- G_e: concentration of extracellular glucose

Rate constants:

- k_A, k_B: from mass-action kinetics
- γ_A, γ_B: protein degradation
- μ cellular growth (causes dilution)

Functions:

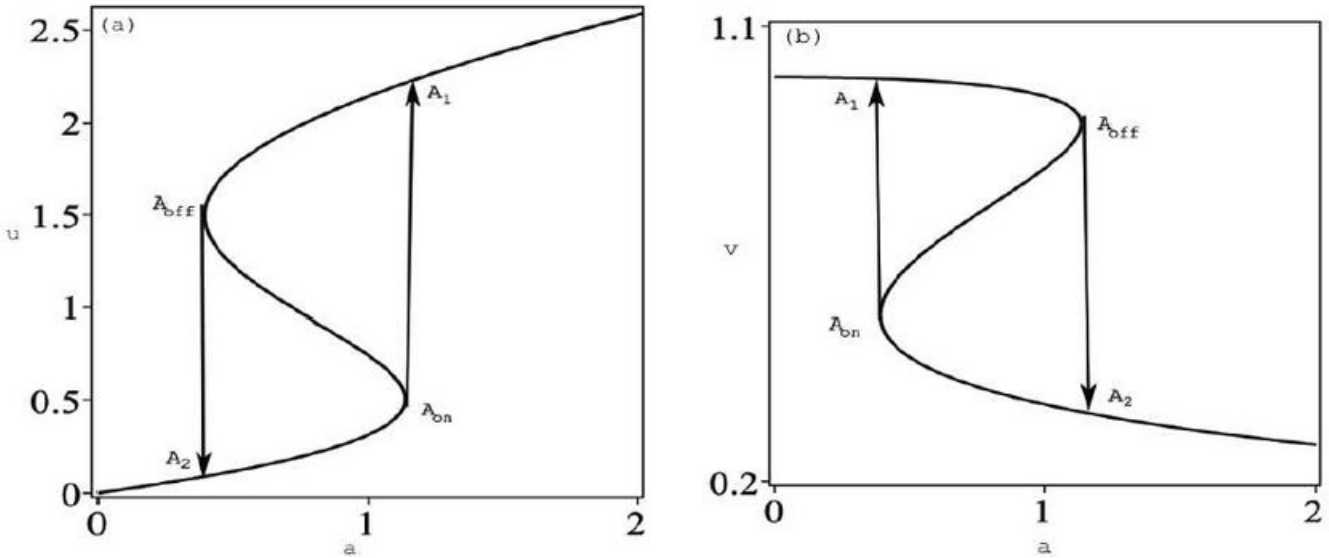
- $P_G(G_e) = \frac{K_G^{nG}}{K_G^{nG} + G_E^{nG}}$, $P_W(W) = \frac{W^{nW}}{K_W^{nW} + W^{nW}}$: sigmoidal Hill functions.

An ODE model of tryptophan metabolism (Orozco-Gómez, 2019)

The authors developed this model using known regulatory mechanisms and experimental data.

$$A' = k_A P_G(G_e) P_W(W) - (\gamma_A + \mu)A$$
$$B' = k_B P_G(G_e) P_W(W) - (\gamma_B + \mu)B$$
$$W' = (\alpha + \beta B)W_e - (\delta + \epsilon A P_A(G_e, W_e) + \mu)W$$

They showed both mathematically and experimentally that the operon is bistable for a specifc range of parameter values.



We showed that Boolean model can predict the same qualitative behavior.

A Boolean model of the *tna* operon

Variables:

- TnaA protein: $f_A = M$
- TnaB protein: $f_B = M$
- cAMP–CAP protein complex: $f_C = \overline{G_e}$.
- Tna mRNA: $f_M = C \wedge \overline{R}$.
- Rho protein (repressor): $f_R = \overline{W} \wedge \overline{W_m}$
- Intracellular tryptophan (high levels): $f_W = W_e \wedge B$
- Intracellular tryptophan: $f_{W_m} = (W_{em} \wedge B) \vee W_e \vee W$

Parameters:

- G_e: extracellular glucose
- W_e: extracellular tryptophan (high levels)
- W_{em}: extracellular tryptophan

Fixed point analysis

Rename our variables:

$$(A, B, C, M, R, W, W_m) = (x_1, x_2, x_3, x_4, x_5, x_6, x_7).$$

To find the fixed points we must solve the system $\{f_{x_i} = x_i \mid i = 1, \ldots, 7\}$ of equations.

This is easiest by first writing functions as polynomials in $\mathbb{F}_2[x_1, \ldots, x_7]$:

$$\begin{cases} x_1 + x_4 = 0 \\ x_2 + x_4 = 0 \\ x_3 + G_e + 1 = 0 \\ x_4 + x_3(1 + x_5) = 0 \\ x_5 + (1 + x_6)(1 + x_7) = 0 \\ x_6 + x_2 \cdot W_3 = 0 \\ x_7 + x_2(x_6 \cdot W_e \cdot W_{em} + x_6 W_{em} + W_e \cdot W_{em} + W_{em}) + x_6(1 + \cdot W_e) + W_e + x_6. \end{cases}$$

We must solve this system for 6 parameter combinations of $(G_e, W_e, W_{em}) \in \mathbb{F}_2^3$.

Fixed point analysis using Macaulay2

-- Define a polynomial ring over $\mathbb{F}_2$:

```
R = ZZ/2[A,B,C,M,R,W,Wm,We,Wem]
```

```
J = ideal(A^2-A,B^2-B,C^2-C,M^2-M,R^2-R,W^2-W,Wm^2-Wm,We^2-We,Wem^2-Wem)
```

```
Q = R/J
```

-- Set shortcuts for AND and OR operations:

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

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

-- Define the Boolean functions

```
f1 = M; f2 = M; f3 = 1+G; f4 = C & (1+R); f5 = (1+W) & (1+Wm); f6 = We & B; f7
= (Wem & B) | We | W;
```

-- Set the parameters (in this case, no glucose, medium levels of tryptophan)

```
G = 0_Q; We = 0_Q; Wem = 1_Q;
```

-- Define the ideal generated by $\{f_{x_i} + x_i \mid i = 1, \ldots, 7\}$ in the quotient ring:

```
I = ideal(f1+A, f2+B, f3+C, f4+M, f5+R, f6+W, f7+Wm)
```

Fixed point analysis

-- Compute a Gröbner basis of I:

```
G = gens gb I;
```

-- This gives the output

```
| W R+Wm+1 M+Wm C+1 B+Wm A+Wm |
```

Which means: $W = 0, \quad C = 1, \quad W_m = A = B = M = R + 1.$

Parameters $x = (G_e, W_e, W_{em})$	Fixed point(s) (A, B, C, M, R, W, W_m)	Operon ON or OFF ?
(0,0,0)	$(0,0,1,0,1,0,0)$	OFF
(0,1,1)	$(0,0,1,0,1,0,1)$	OFF
(0,0,1)	$(0,0,1,0,0,0,0)$	OFF
	$(1,1,1,1,0,0,1)$	ON
(1,0,0)	$(0,0,0,0,1,0,0)$	OFF
(1,1,1)	$(0,0,0,0,0,0,1)$	OFF
(1,0,1)	$(0,0,0,0,0,0,0)$	OFF

Table : Fixed points of our *tna* operon Boolean network model for each choice of parameters.

Summary

All of the fixed points make sense biologically and predict bistability.

Forward vs. reverse engineering

The previous model is an example of forward engineering: Given biological knowledge, proposal a model, generate data, and analyze it.

The reverse engineering problem does the opposite: given data, use it to generate a model.

There are many modeling frameworks:

- Differential equations
- Difference equations
- Statistical models
- Boolean or logical networks

All of these utilize different techniques.

We'll look at this last framework. Computational algebraic techniques tend to arise in their analysis.

Mammalian signaling pathways

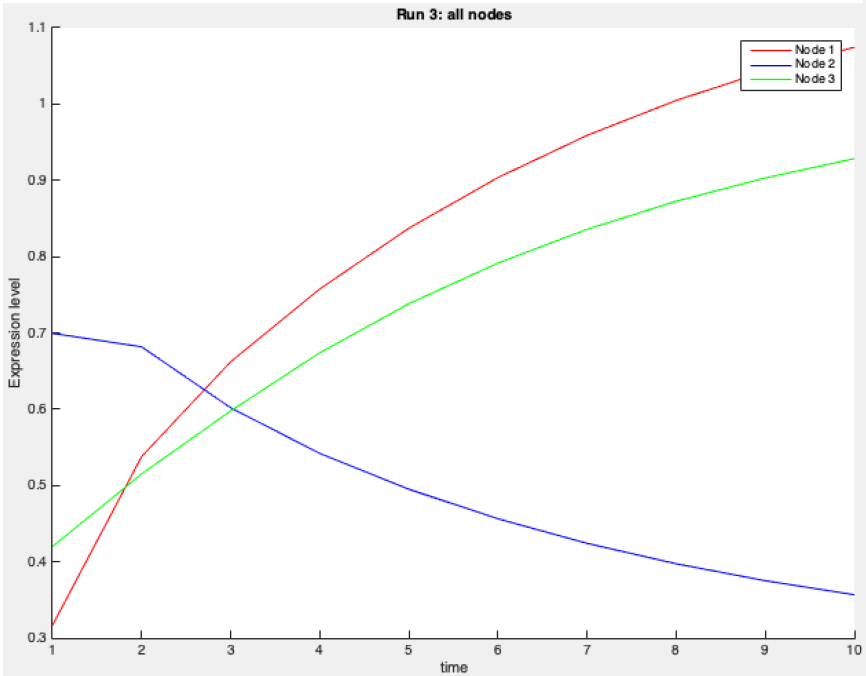
From "*Network Reconstruction from Perturbation Time Course Data*" (Smith et al., 2019).

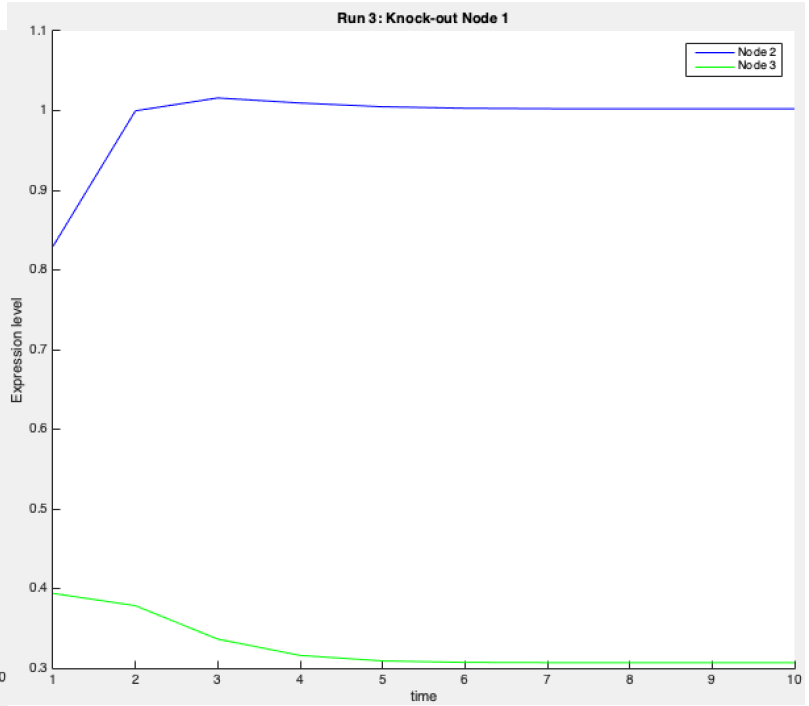
They propose an ODE-based algorithm that analyzes data from gene knockouts. They train it on synthetic 2-node and 3-node networks constructed in Matlab.

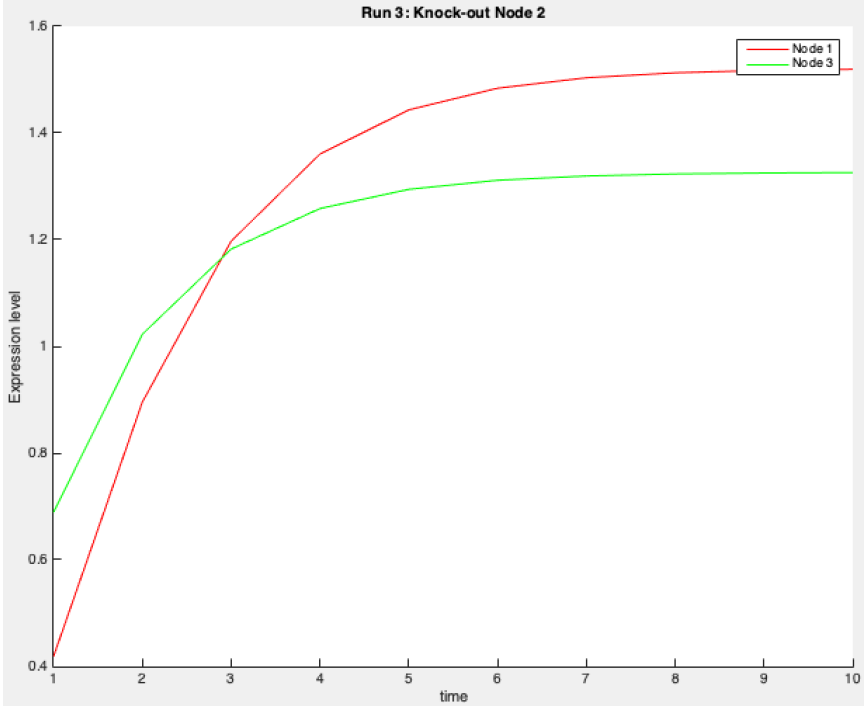
Then, they apply their methods to the ERK and AKT pathways, widely important in mammalian signaling.

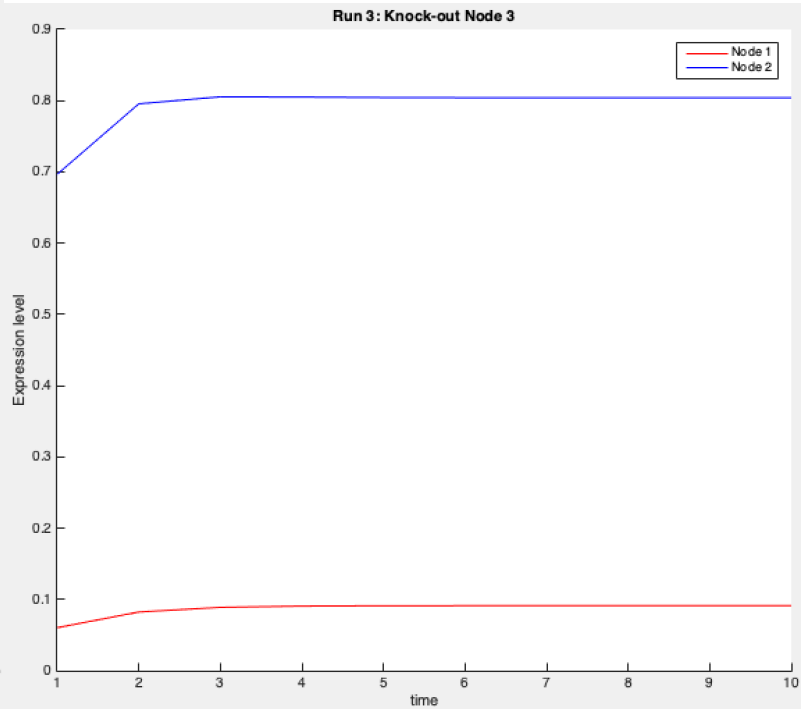
Goal

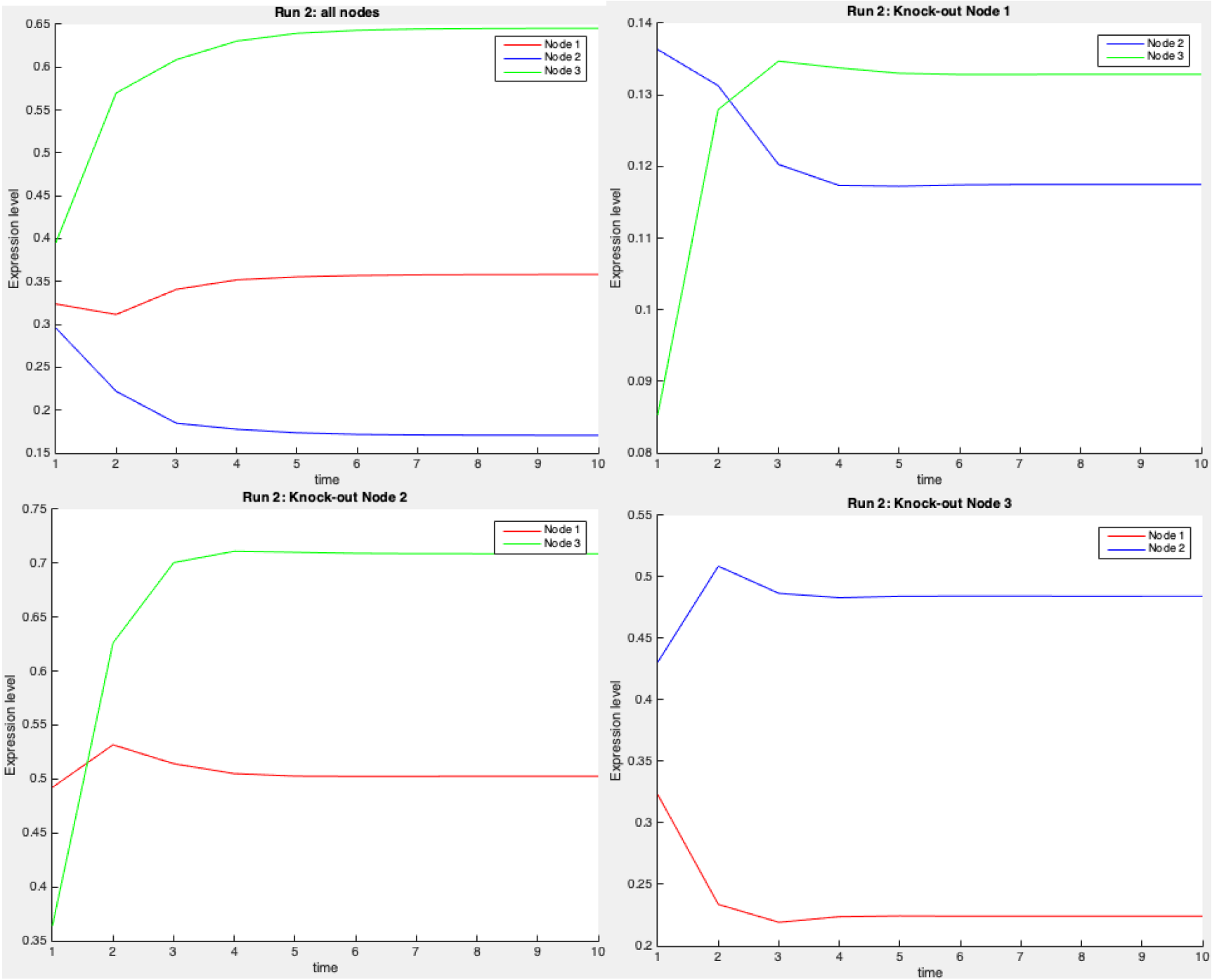
Can algebraic methods do better?


Run 3: all nodes
Node 1
Node 2
Node 3
Expression level
time


Run 3: Knock-out Node 1
Node 2
Node 3
Expression level
time


Run 3: Knock-out Node 2
Node 1
Node 3
Expression level
time


Run 3: Knock-out Node 3
Node 1
Node 2
Expression level
time


Run 2: all nodes
Node 1
Node 2
Node 3
Expression level
time
Run 2: Knock-out Node 1
Node 2
Node 3
Expression level
time
Run 2: Knock-out Node 2
Node 1
Node 3
Expression level
time
Run 2: Knock-out Node 3
Node 1
Node 2
Expression level
time

Broad goal

Suppose we have an unknown Boolean function $f_i : \mathbb{F}_2^3 \to \mathbb{F}_2$:

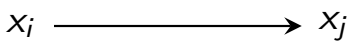
$x_1x_2x_3$	111	110	101	100	011	010	001	000
$f_i(x)$	0	0	?	?	?	?	?	1

Goal

Find all "minimal wiring diagrams".

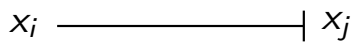
Different types of interactions are indicated in the wiring diagram:

$f_j = x_i \wedge x_k$



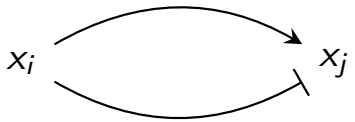
"x_i activates x_j"

$f_j = \overline{x_i} \wedge x_k$



"x_i inhibits x_j"

$f_j = x_i + x_k$



"x_i affects x_j positively & negatively"

Algebraic methods have been proposed in the following papers:

1. Unsigned version, using monomial ideals: *Reverse engineering of dynamics networks* (Stigler, Jarrah, Stillman, Laubenbacher, 2007)
2. Signed version, using pseudomonomial ideals: *An algebraic approach to reverse engineering finite dynamical systems arising from biology* (Veliz-Cuba, 2012)

The basic idea

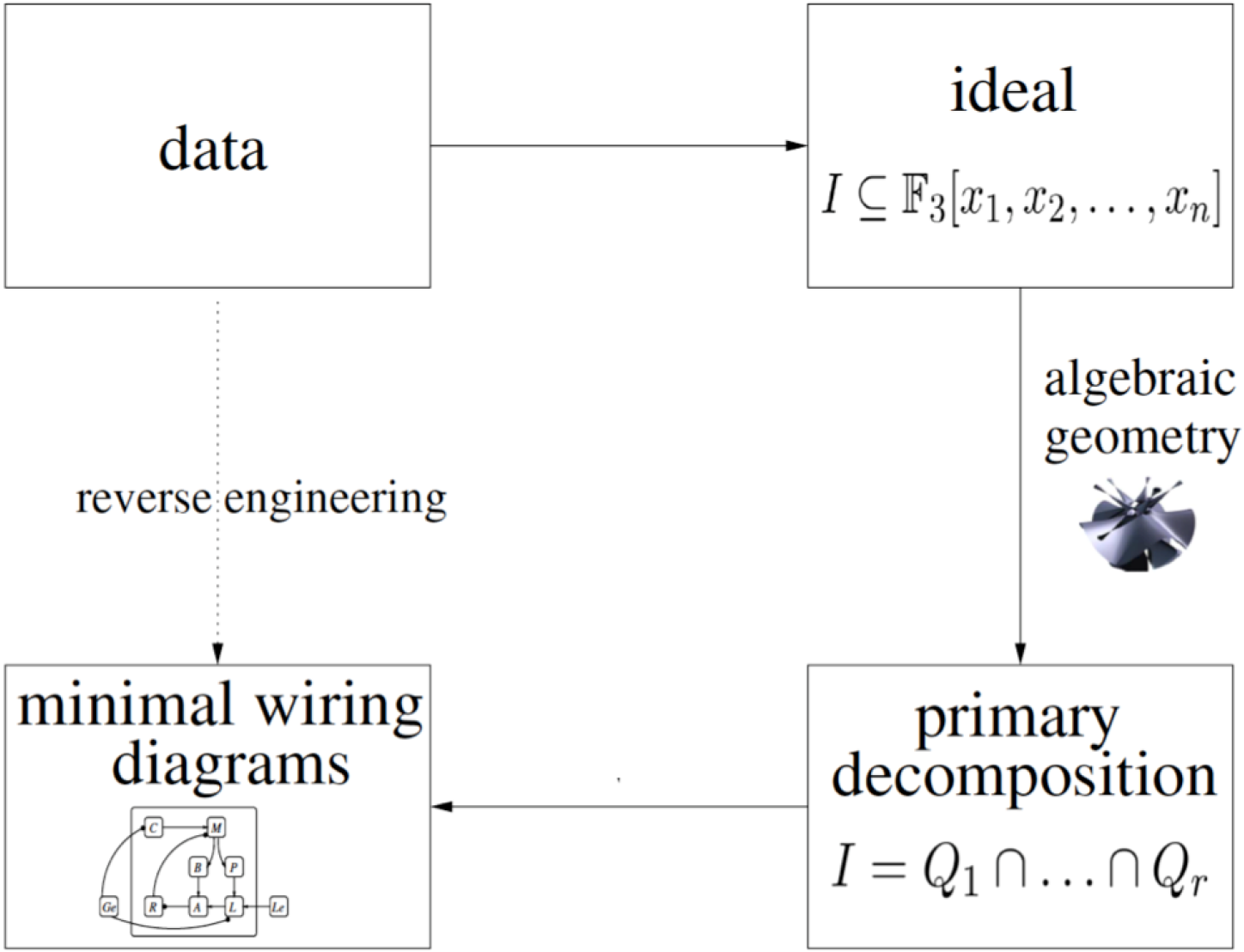


Figure : Image courtesy of Alan Veliz-Cuba.

Data and model spaces

Let $f : \mathbb{F}^n \to \mathbb{F}$ be a function, where $\mathbb{F} = \mathbb{F}_p$.

Definition

Consider a set of data

$$\mathcal{D} = \{(\mathbf{s}_1, t_1), \ldots, (\mathbf{s}_m, t_m)\}, \qquad \mathbf{s}_i \in \mathbb{F}^n, \quad t_i \in \mathbb{F}$$

of input-output pairs, all $\mathbf{s}_i$ are distinct. We say that f fits the data $\mathcal{D}$ if

$$f(\mathbf{s}_i) = f(s_{i1}, \ldots, s_{in}) = t_i, \qquad \text{for all } i = 1, \ldots, m.$$

The model space of $\mathcal{D}$ is the set $\text{Mod}(\mathcal{D})$ of all functions that fit the data, i.e.,

$$\text{Mod}(\mathcal{D}) = \{f : \mathbb{F}^n \to \mathbb{F} \mid f(\mathbf{s}_i) = t_i \text{ for all } i = 1, \ldots, m\}.$$

For any f in $\text{Mod}(\mathcal{D})$, the support of f is the set of variables on which f depends.

Under a slight abuse of notation, we can think of the support as a subset of $\{x_1, \ldots, x_n\}$ or as a subset $\alpha \subseteq [n] = \{1, \ldots, n\}$.

Feasible, disposable, and min-sets

Definition

With respect to a set $\mathcal{D}$ of data, a set $\alpha \subseteq [n]$ is:

- feasible if there is there is some $f \in \text{Mod}(\mathcal{D})$ for which $\text{supp}(f) \subseteq \alpha$.
- disposable if there is some $f \in \text{Mod}(\mathcal{D})$ for which $\text{supp}(f) \cap \alpha = \emptyset$.

Note that a set α is feasible if and only if its complement $\overline{\alpha} := [n] - \alpha$ is disposable.

Remark

These are *not* opposite concepts; a set can be both feasible and disposable, or neither.

Key point

Let $\mathcal{D}$ be a set of data, and $\alpha, \beta \subseteq [n]$.

(i) If α and β are feasible with respect to $\mathcal{D}$, then so is $\alpha \cup \beta$.

(ii) If α and β are disposable with respect to $\mathcal{D}$, then so is $\alpha \cap \beta$.

In particular, the disposable sets of $\mathcal{D}$ form a simplicial complex $\Delta_{\mathcal{D}}$.

Definition

A subset $\alpha \subseteq [n]$ is a min-set of $\mathcal{D}$ if its complement $\overline{\alpha} := [n] - \alpha$ is a maximal disposable set of $\mathcal{D}$.

Alexander duality

Definition

Given an ideal I in $\mathbb{F}[x_1, \ldots, x_n]$, define the simplicial complex

$$\Delta_{I^c} := \{\alpha \mid x^\alpha \notin I\}.$$

Given a simplicial complex Δ, define a square-free monomial ideal

$$I_{\Delta^c} := \langle x^\alpha \mid \alpha \notin \Delta \rangle.$$

This is called the Stanley-Reisner ideal of Δ.

Theorem

The correspondence $I \mapsto \Delta_{I^c}$ and $\Delta \mapsto I_{\Delta^c}$ is a bijection between:

(i) simplicial complexes on $[n] = \{1, \ldots, n\}$,

(ii) square-free monomial ideals in $\mathbb{F}[x_1, \ldots, x_n]$.

This correspondence is called Alexander duality.

Min-sets and Stanley-Reisner theory applied to min-sets

Theorem

There is a bijective correspondence between:

- the simplicial complex $\Delta_{\mathcal{D}}$ of disposable sets,
- the square-free monomial ideal $I_{\Delta_{\mathcal{D}}^c}$ in $\mathbb{F}[x_1, \ldots, x_n]$ of non-disposable sets.

In other words, α is a min-set of $\mathcal{D}$ if and only if $\overline{\alpha}$ is a maximal disposable set, and

$$x^{\alpha} \in I_{\Delta_{\mathcal{D}}^c} \quad \text{if and only if } \alpha \text{ is non-disposable.}$$

For each pair $(\mathbf{s}, t), (\mathbf{s}', t') \in \mathcal{D}$, define the monomial

$$m(\mathbf{s}, \mathbf{s}') := \prod_{s_i \neq s_i'} x_i.$$

By construction, if $t \neq t'$, then $\operatorname{supp}(m(\mathbf{s}, \mathbf{s}'))$ *must* be non-disposable.

Theorem

The ideal of non-disposable sets is the ideal in $\mathbb{F}_2[x_1, \ldots, x_n]$ defined by

$$I_{\Delta_{\mathcal{D}}^c} = \langle m(\mathbf{s}, \mathbf{s}') \mid t \neq t' \rangle.$$

The generators of the primary components of $I_{\Delta_D^c}$ are the min-sets of $\mathcal{D}$.

Example 2 (continued)

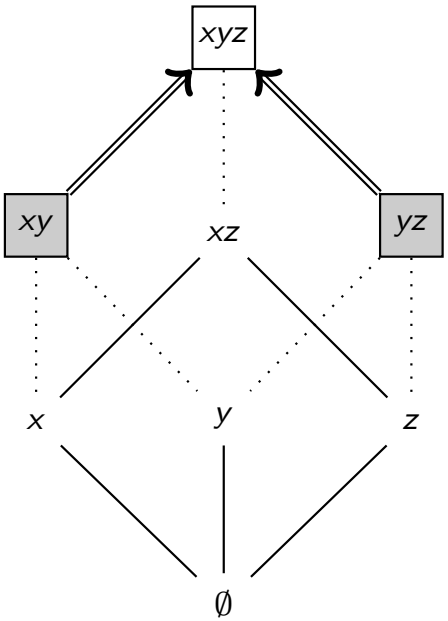
Consider a Boolean function $f : \mathbb{F}_2^3 \to \mathbb{F}_2$ with the following partial data:

xyz	101	000	110
$f(x, y, z)$	0	0	1

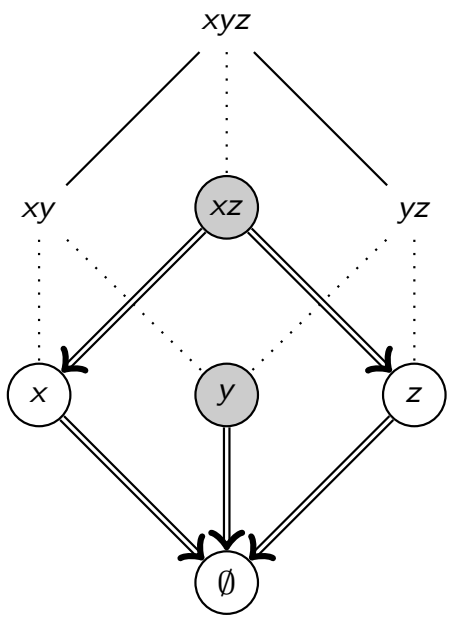
Using our notation, the *data* $\mathcal{D}$, grouped by output value, is

$$\mathcal{D} = \{(\mathbf{s}_1, t_1), (\mathbf{s}_2, t_2), (\mathbf{s}_3, t_3)\} = \{(101, 0), (000, 0), (110, 1)\}.$$

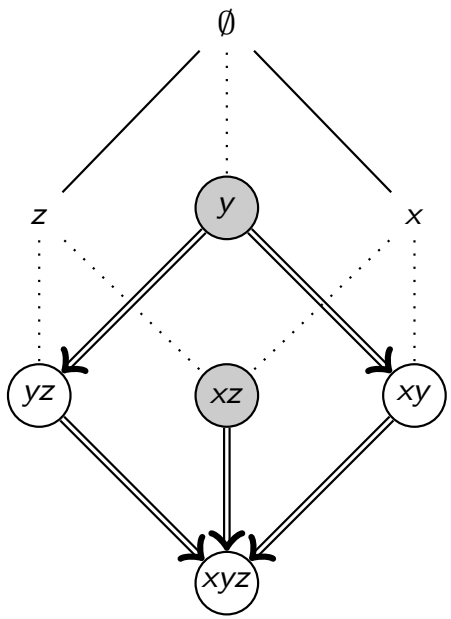
Since $t_1 = t_2 \neq t_3$, we compute $m(\mathbf{s}_1, \mathbf{s}_3) = yz$ and $m(\mathbf{s}_2, \mathbf{s}_3) = xy$.



Non-disposable sets $\Delta_{\mathcal{D}}^c$;
Monomials in $I_{\Delta_{\mathcal{D}}^c}$



Disposable sets $\Delta_{\mathcal{D}}$



Feasible sets of $\mathcal{D}$

The min-sets are shaded

Example 3

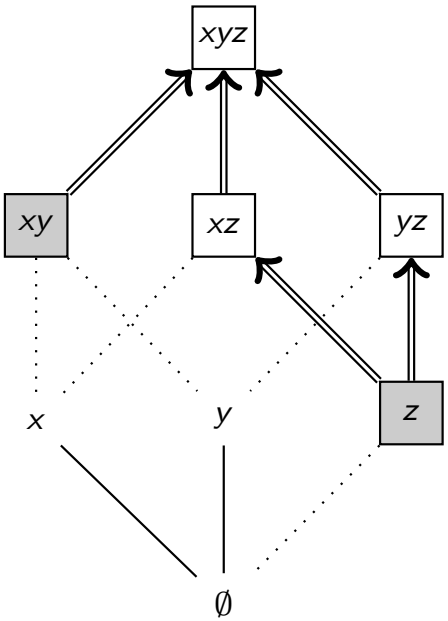
Consider a Boolean function $f : \mathbb{F}_2^3 \to \mathbb{F}_2$ with the following partial data:

xyz	111	000	110
$f(x, y, z)$	0	0	1

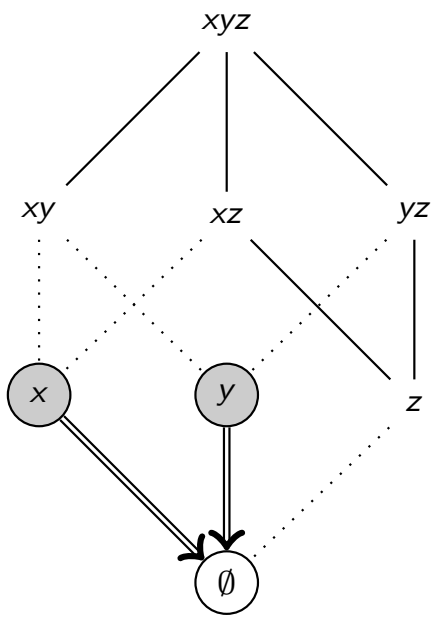
Using our notation, the *data* $\mathcal{D}$, grouped by output value, is

$$\mathcal{D} = \big\{(\mathbf{s}_1, t_1), (\mathbf{s}_2, t_2), (\mathbf{s}_3, t_3)\big\} = \big\{(111, 0), (000, 0), (110, 1)\big\}.$$

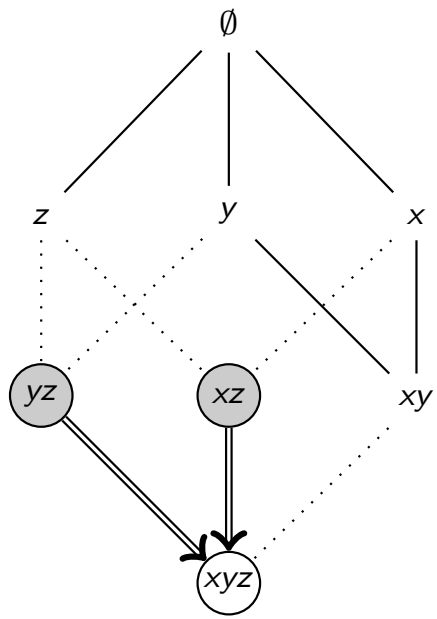
Since $t_1 = t_2 \neq t_3$, we compute $m(\mathbf{s}_1, \mathbf{s}_3) = z$ and $m(\mathbf{s}_2, \mathbf{s}_3) = xy$.



Non-disposable sets $\Delta_{\mathcal{D}}^c$;
Monomials in $I_{\Delta_{\mathcal{D}}^c}$



Disposable sets $\Delta_{\mathcal{D}}$



Feasible sets of $\mathcal{D}$

The min-sets are shaded

Summary so far

The following table summarizes the correspondence between the combinatorial structures in the Boolean network problem to Stanley-Reisner theory and Alexander duality.

Reverse engineering of local models	Stanley-Reisner theory
Disposable sets of $\mathcal{D}$	Faces of the simplicial complex $\Delta_{\mathcal{D}}$
Non-disposable sets of $\mathcal{D}$	The non-faces, $\Delta_{\mathcal{D}}^{c}$
The ideal $\langle m(\mathbf{s}, \mathbf{s}') \mid t \neq t' \rangle$ of non-disposable sets	The Stanley-Reisner ideal $I_{\Delta_{\mathcal{D}}^{c}}$
Feasible sets of $\mathcal{D}$	Complements of faces of $\Delta_{\mathcal{D}}$
Min-sets of $\mathcal{D}$	Complements of max'l faces of $\Delta_{\mathcal{D}}$ $\leftrightarrow$ primary components of $I_{\Delta_{\mathcal{D}}^{c}}$

Finding signed min-sets of local models

Consider a set of data (i.e., input-output pairs) with all $\mathbf{s}_i$ distinct:

$$\mathcal{D} = \{(\mathbf{s}_1, t_1), \ldots, (\mathbf{s}_m, t_m)\}, \qquad \mathbf{s}_i \in \mathbb{F}^n, \quad t_i \in \mathbb{F}.$$

Order the data so the output values are non-decreasing, i.e., $t_1 \leq \cdots \leq t_m$.

For each pair $(\mathbf{s}, t), (\mathbf{s}', t') \in \mathcal{D}$ with different outputs, i.e., $t < t'$, we encode each coordinate x_i where they differ with $x_i \pm 1$.

- $(x_i - 1)$ if the interaction is positive ($s_i < s_i'$),
- $(x_i + 1)$ if the interaction is negative ($s_i > s_i'$).

Then define the pseudomonomial

$$p(\mathbf{s}, \mathbf{s}') := \prod_{s_i \neq s_i'} \big(x_i - \operatorname{sign}(s_i' - s_i)\big).$$

Theorem

The ideal of signed non-disposable sets is the ideal in $\mathbb{F}_3[x_1, \ldots, x_n]$ defined by

$$J_{\Delta_{\mathcal{D}}^c} = \big\langle p(\mathbf{s}_i, \mathbf{s}_j) \mid i < j,\ t_i \neq t_j \big\rangle.$$

The primary components of $J_{\Delta_{\mathcal{D}}^c}$ give the signed min-sets.

Example (from "Broad goal" slide)

Consider a Boolean function $f : \mathbb{F}_2^3 \to \mathbb{F}_2$ with the following partial data:

xyz	111	000	110
$f(x, y, z)$	0	0	1

The *data* $\mathcal{D}$ is

$$\mathcal{D} = \big\{(\mathbf{s}_1, t_1), (\mathbf{s}_2, t_2), (\mathbf{s}_3, t_3)\big\} = \big\{(111, 0), (000, 0), (110, 1)\big\}.$$

Note that

$$p(\mathbf{s}_1, \mathbf{s}_3) = z - (\text{sign}(s_{33} - s_{13})) = z + 1, \qquad p(\mathbf{s}_2, \mathbf{s}_3) = (x - 1)(y - 1).$$

The ideal of signed non-disposable sets for $\mathcal{D}$ is thus

$$J_{\Delta_{\mathcal{D}}^c} = \big\langle p(\mathbf{s}_1, \mathbf{s}_3),\ p(\mathbf{s}_2, \mathbf{s}_3) \big\rangle = \big\langle z + 1,\ (x - 1)(y - 1) \big\rangle.$$

The following Macaulay2 commands compute the primary decomposition of $J_{\Delta_{\mathcal{D}}^c}$:

```
R = ZZ/3[x,y,z];
J_nonDisp = ideal(z+1, (x-1)*(y-1));
primaryDecomposition J_nonDisp
```

Output: `{ideal (z + 1, y - 1), ideal (z + 1, x - 1)}`

- Primary decomposition: $J_{\Delta_{\mathcal{D}}^c} = \big\langle x - 1, z + 1 \big\rangle \cap \big\langle y - 1, z + 1 \big\rangle$.
- Signed min-sets: $\{x, \overline{z}\}$ and $\{y, \overline{z}\}$.

Summary

We call these *pseudomonimal ideals* because

$$J_{\Delta^c_{\mathcal{D}}} = \langle z+1,\ (x-1)(y-1)\rangle = \langle x-1, z+1\rangle \cap \langle y-1, z+1\rangle \subseteq \mathbb{F}_3[x,y,z]$$

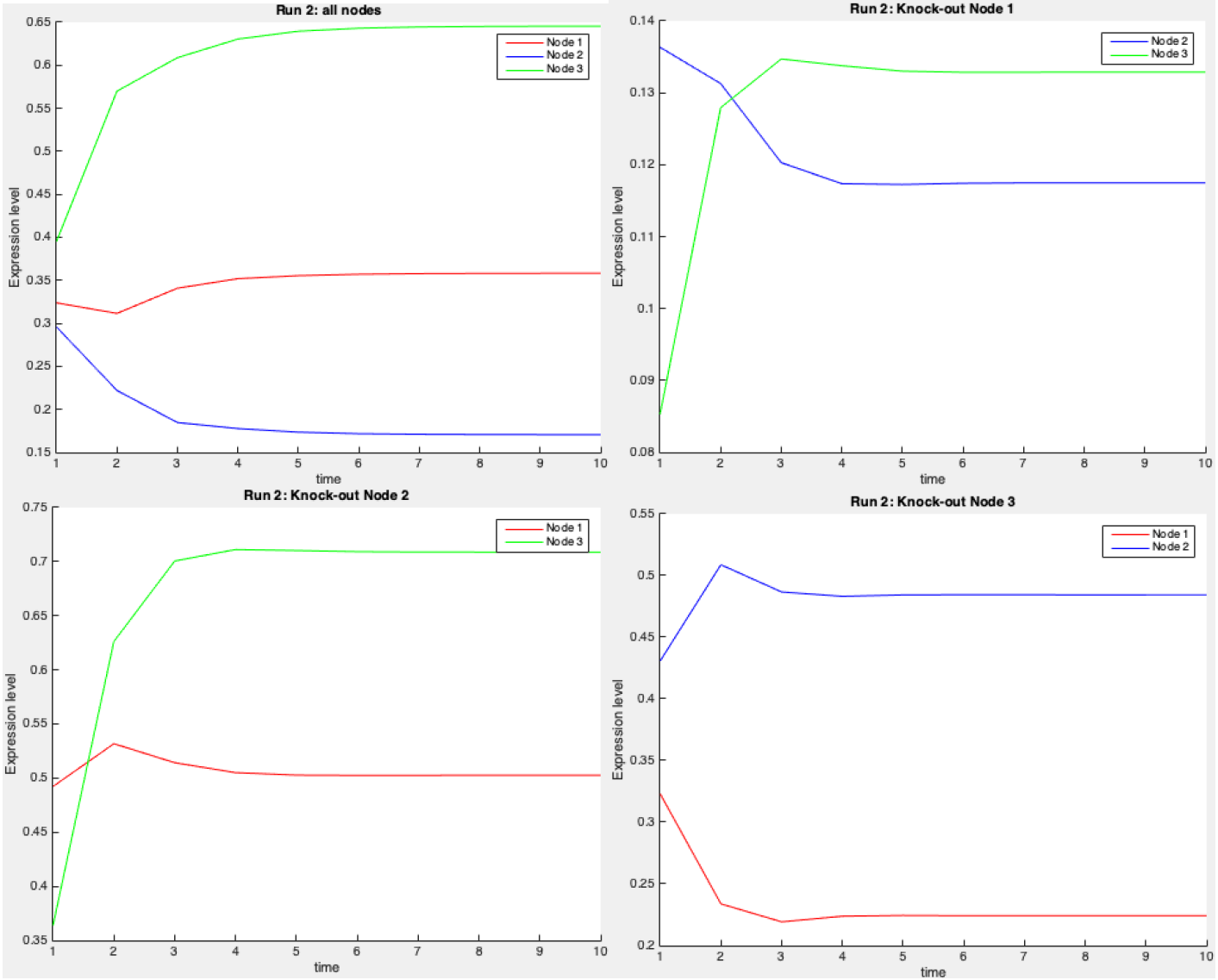
can be thought of as

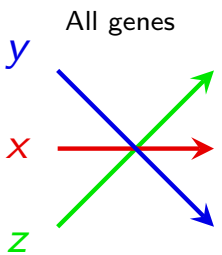
$$J_{\Delta^c_{\mathcal{D}}} = \langle \overline{z},\ xy\rangle = \langle x, \overline{z}\rangle \cap \langle y, \overline{z}\rangle \subseteq \mathbb{F}_3[x,y,z,\overline{x},\overline{y},\overline{z}],$$

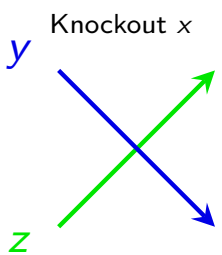
which become monomial ideals under polarization.

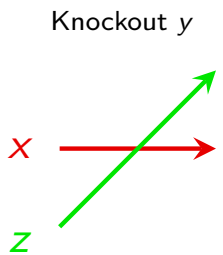
In the "unsigned" case, Stanley-Reisner theory and Alexander duality provides a correspondence between combinatorial structures in monomial ideals and Boolean networks.

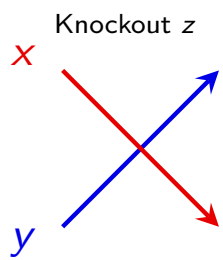
Reverse engineering of local models	Stanley-Reisner theory
Disposable sets of $\mathcal{D}$	Faces of the simplicial complex $\Delta_{\mathcal{D}}$
Non-disposable sets of $\mathcal{D}$	The non-faces, $\Delta^c_{\mathcal{D}}$
The ideal $\langle m(\mathbf{s},\mathbf{s}') \mid t \neq t'\rangle$ of non-disposable sets	The Stanley-Reisner ideal $I_{\Delta^c_{\mathcal{D}}}$
Feasible sets of $\mathcal{D}$	Complements of faces of $\Delta_{\mathcal{D}}$
Min-sets of $\mathcal{D}$	Complements of max'l faces of $\Delta_{\mathcal{D}}$ $\leftrightarrow$ primary components of $I_{\Delta^c_{\mathcal{D}}}$


Run 2: all nodes
Node 1
Node 2
Node 3
Expression level
time
Run 2: Knock-out Node 1
Node 2
Node 3
Expression level
time
Run 2: Knock-out Node 2
Node 1
Node 3
Expression level
time
Run 2: Knock-out Node 3
Node 1
Node 2
Expression level
time


All genes
y
x
z

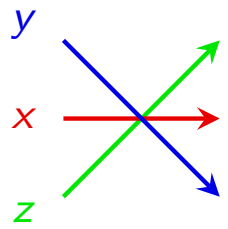

Knockout x
y
z


Knockout y
x
z


Knockout z
x
y

An alternative to discretization

All genes



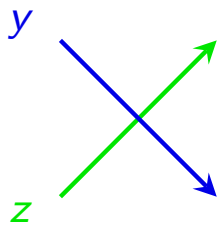
		t	x	y	z
$\vec{\mathbf{s}}$		0	1	2	0
$\vec{\mathbf{t}}$	$\vec{\mathbf{s}}'$	1	1	1	1
	$\vec{\mathbf{t}}'$	2	1	0	2

$p(\vec{\mathbf{s}}, \vec{\mathbf{s}}') = 0$

$p(\vec{\mathbf{s}}, \vec{\mathbf{s}}') = (y+1)(z-1)$

$p(\vec{\mathbf{s}}, \vec{\mathbf{s}}') = (y+1)(z-1)$

Knockout x

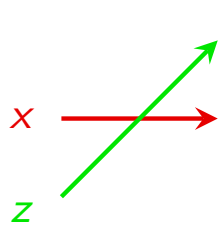


t	y	z
0	2	0
1	1	1
2	0	2

$p(\vec{\mathbf{s}}, \vec{\mathbf{s}}') = (y+1)(z-1)$

$p(\vec{\mathbf{s}}, \vec{\mathbf{s}}') = (y+1)(z-1)$

Knockout y

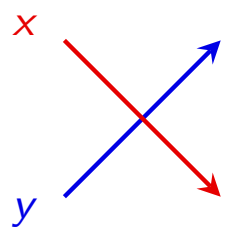


t	x	z
0	1	0
1	1	1
2	1	2

$p(\vec{\mathbf{s}}, \vec{\mathbf{s}}') = 0$

$p(\vec{\mathbf{s}}, \vec{\mathbf{s}}') = (z-1)$

Knockout z



t	x	y
0	2	0
1	1	1
2	0	2

$p(\vec{\mathbf{s}}, \vec{\mathbf{s}}') = (x+1)(y-1)$

$p(\vec{\mathbf{s}}, \vec{\mathbf{s}}') = (x+1)(y-1)$

We get the following ideals in $\mathbb{F}_3[x, y, z]$:

$$J_x = \langle (x+1)(y-1) \rangle, \qquad J_y = \langle (y+1)(z-1), (x+1)(y-1) \rangle, \qquad J_z = \langle (y+1)(z-1), (z-1) \rangle.$$

Min-sets from primary decompositions

The primary decompositions of these pseudomonimal ideals are:

$$J_x = \langle (x+1)(y-1) \rangle = \langle x+1 \rangle \cap \langle y-1 \rangle$$

$$J_y = \langle (y+1)(z-1), (x+1)(y-1) \rangle = \langle x+1, z+1 \rangle \cap \langle y-1 \rangle$$

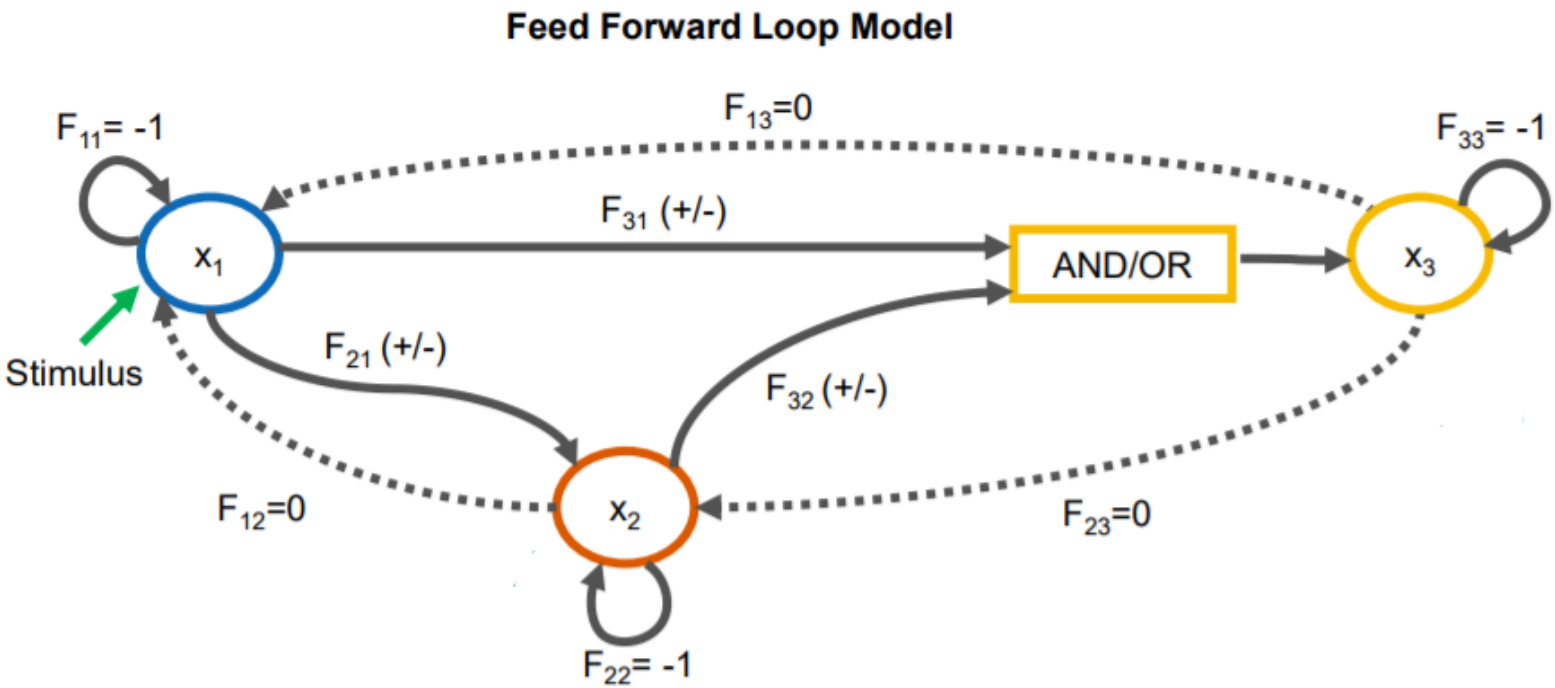
$$J_z = \langle (y+1)(z-1), (z-1) \rangle = \langle z-1 \rangle$$

This means that the signed min-sets are:

- Gene x: $\{\overline{x}\}$, $\{\overline{y}\}$
- Gene y: $\{\overline{x}, \overline{z}\}$, $\{y\}$
- Gene z: $\{z\}$

What to try next

In the same paper "*Network Reconstruction from Perturbation Time Course Data*" (Smith et al., 2019), they try out their reverse engineering algorithm on synthetic data from an AND/OR network:



We are optimistic that the algebraic methods will perform better, because AND/OR are monotone functions.

Current & future work

Biological

- Apply these methods to experimental mammalian signaling data.

Mathematical

- Develop a "signed Stanley-Reisner theory".
- Better understand what the primary decomposition means for pseudomonomial ideals.
- Use gene knockouts to reverse-engineer canalizing functions.