ASKeD-BN: Automatic Synthesis of Boolean Networks from Biological Knowledge and Data

ola 2021

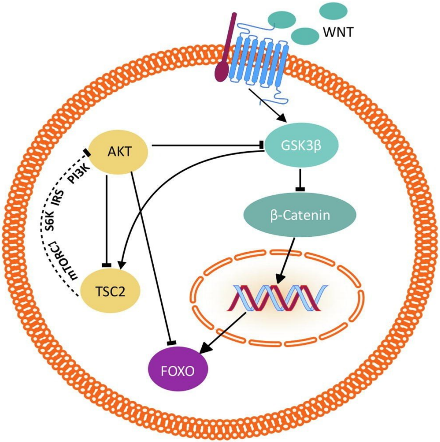
Athénaïs Vaginay, Taha Boukhobza, Malika Smaïl-Tabbone

Jun 21-23 2021

Modeling Biological Systems with Boolean Networks

Boolean Networks (BN) are:

- qualitative formalism, well-suited for biological systems
- built from experimental data and knowledge from literature
- the **automatic** synthesis of BNs from biological data and knowledge is still a challenge



from Schwab et al. 2020.

Boolean Networks – Generalities

$$\mathscr{B} = \begin{cases} f_{\mathsf{A}} : a_{t+1} & = & c_t \\ f_{\mathsf{B}} : b_{t+1} & = & b_t \wedge \neg c_t \\ f_{\mathsf{C}} : c_{t+1} & = & \neg c_t \end{cases}$$

in the Boolean world: $\mathbb{B} = \{0, 1\}$

n Boolean **components**

BN = set of n **update functions**

negation: $\neg$ disjunction: $\vee$ conjunction: $\wedge$

Boolean Networks – Generalities

$$\mathscr{B} = \begin{cases} f_{\mathsf{A}} : a_{t+1} & = & c_t \\ f_{\mathsf{B}} : b_{t+1} & = & b_t \wedge \neg c_t \\ f_{\mathsf{C}} : c_{t+1} & = & \neg c_t \end{cases}$$

in the Boolean world: $\mathbb{B} = \{0, 1\}$

n Boolean **components**

BN = set of n **update functions**

negation: $\neg$ disjunction: $\vee$ conjunction: $\wedge$

Boolean Networks – Generalities

$$\mathscr{B} = \begin{cases} f_{\mathsf{A}} : a_{t+1} &= c_t \\ f_{\mathsf{B}} : b_{t+1} &= b_t \wedge \neg c_t \\ f_{\mathsf{C}} : c_{t+1} &= \neg c_t \end{cases}$$

in the Boolean world: $\mathbb{B} = \{0, 1\}$

n Boolean **components**

BN = set of n **update functions**

negation: $\neg$ disjunction: $\vee$ conjunction: $\wedge$

Boolean Networks – Generalities

$$\mathscr{B} = \begin{cases} f_{\mathsf{A}} : a_{t+1} &= c_t \\ f_{\mathsf{B}} : b_{t+1} &= b_t \wedge \neg c_t \\ f_{\mathsf{C}} : c_{t+1} &= \neg c_t \end{cases}$$

in the Boolean world: $\mathbb{B} = \{0, 1\}$

n Boolean **components**

BN = set of n **update functions**

negation: $\neg$ disjunction: $\vee$ conjunction: $\wedge$

Boolean Networks – More About Update Functions

$$\mathscr{B} = \begin{cases} f_{\mathsf{A}} : a_{t+1} & = & c_t \\ f_{\mathsf{B}} : b_{t+1} & = & b_t \wedge \neg c_t \\ f_{\mathsf{C}} : c_{t+1} & = & \neg c_t \end{cases}$$

Boolean Networks – More About Update Functions

$$\mathscr{B} = \begin{cases} f_{\mathsf{A}} : a_{t+1} = c_t & \text{“C activates A”} \\ f_{\mathsf{B}} : b_{t+1} = b_t \wedge \neg c_t \\ f_{\mathsf{C}} : c_{t+1} = \neg c_t \end{cases}$$

Boolean Networks – More About Update Functions

$$\mathscr{B} = \begin{cases} f_{\mathsf{A}} : a_{t+1} \;=\; c_t \\ f_{\mathsf{B}} : b_{t+1} \;=\; b_t \wedge \neg c_t \\ f_{\mathsf{C}} : c_{t+1} \;=\; \neg c_t \end{cases}$$

status of child component at $t+1$
= f(status of parents components at t)

Boolean Networks – More About Update Functions

$$\mathscr{B} = \begin{cases} f_{\mathsf{A}} : a_{t+1} \ = \ c_t \\ f_{\mathsf{B}} : b_{t+1} \ = \ b_t \wedge \neg c_t \\ f_{\mathsf{C}} : c_{t+1} \ = \ \neg c_t \end{cases}$$

status of child component at $t+1$
= f(status of parents components at t)

Boolean Networks – More About Update Functions

$$\mathscr{B} = \begin{cases} f_{\mathsf{A}} : a_{t+1} & = & c_t \\ f_{\mathsf{B}} : b_{t+1} & = & b_t \wedge \neg c_t \\ f_{\mathsf{C}} : c_{t+1} & = & \neg c_t \end{cases}$$

status of child component at $t+1$
= f(status of parents components at t)

Boolean Networks – More About Update Functions

$$\mathscr{B} = \begin{cases} f_{\mathsf{A}} : a_{t+1} &= c_t \\ f_{\mathsf{B}} : b_{t+1} &= b_t \wedge \neg c_t \\ f_{\mathsf{C}} : c_{t+1} &= \neg c_t \end{cases}$$

status of child component at $t+1$
= f(status of parents components at t)

Boolean Networks – More About Update Functions

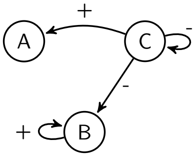
$$\mathscr{B} = \begin{cases} f_{\mathsf{A}} : a_{t+1} & = & c_{t} \\ f_{\mathsf{B}} : b_{t+1} & = & b_{t} \wedge \neg c_{t} \\ f_{\mathsf{C}} : c_{t+1} & = & \neg c_{t} \end{cases}$$

status of child component at $t+1$
= f(status of parents components at t)

Boolean Networks – Their Structure and Dynamics

$$\mathscr{B} = \begin{cases} f_{\mathsf{A}} : a_{t+1} & = & c_t \\ f_{\mathsf{B}} : b_{t+1} & = & b_t \wedge \neg c_t \\ f_{\mathsf{C}} : c_{t+1} & = & \neg c_t \end{cases}$$

Boolean Networks – Their Structure and Dynamics



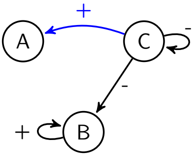
$$\mathscr{B} = \begin{cases} f_{\mathsf{A}} : a_{t+1} \;=\; c_t \\ f_{\mathsf{B}} : b_{t+1} \;=\; b_t \land \neg c_t \\ f_{\mathsf{C}} : c_{t+1} \;=\; \neg c_t \end{cases}$$

Interaction Graph (IG)

nodes: components of the BN

edges: influences + polarity

Boolean Networks – Their Structure and Dynamics



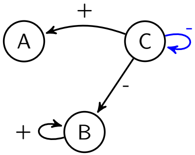
$$\mathscr{B} = \begin{cases} f_{\mathsf{A}} : a_{t+1} & = & c_t \\ f_{\mathsf{B}} : b_{t+1} & = & b_t \wedge \neg c_t \\ f_{\mathsf{C}} : c_{t+1} & = & \neg c_t \end{cases}$$

Interaction Graph (IG)

nodes: components of the BN

edges: influences + polarity

Boolean Networks – Their Structure and Dynamics



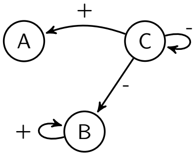
$$\mathscr{B} = \begin{cases} f_{\mathsf{A}} : a_{t+1} &= c_t \\ f_{\mathsf{B}} : b_{t+1} &= b_t \wedge \neg c_t \\ f_{\mathsf{C}} : c_{t+1} &= \neg c_t \end{cases}$$

Interaction Graph (IG)

nodes: components of the BN

edges: influences + polarity

Boolean Networks – Their Structure and Dynamics

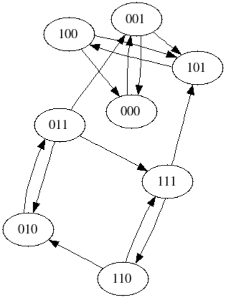


Interaction Graph (IG)

nodes: components of the BN

edges: influences + polarity

$$\mathscr{B} = \begin{cases} f_{\mathsf{A}} : a_{t+1} &= c_t \\ f_{\mathsf{B}} : b_{t+1} &= b_t \wedge \neg c_t \\ f_{\mathsf{C}} : c_{t+1} &= \neg c_t \end{cases}$$

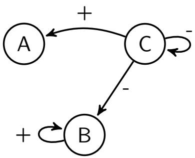


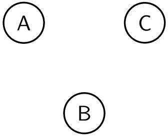
State Transition Graph (STG)

nodes: configurations of the BN (vector $\in \mathbb{B}^n$)
e.g. 001, 010, 111, ...

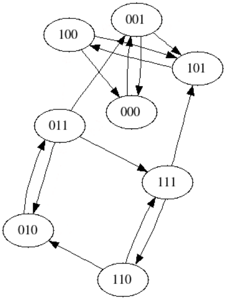
edge from c to c' if c' = f(c)

Boolean Networks – Synthesis from Knowledge and Data

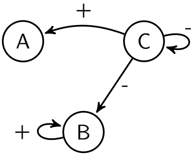




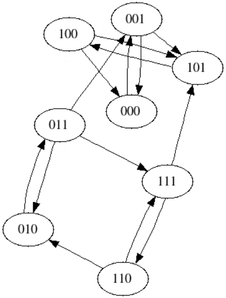
$$\mathscr{B} = \begin{cases} f_{\mathsf{A}} : a_{t+1} &= c_t \\ f_{\mathsf{B}} : b_{t+1} &= b_t \wedge \neg c_t \\ f_{\mathsf{C}} : c_{t+1} &= \neg c_t \end{cases}$$



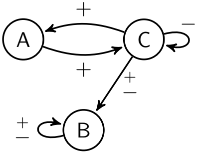
Boolean Networks – Synthesis from Knowledge and Data



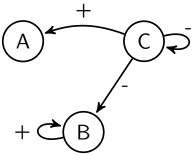
$$\mathscr{B} = \begin{cases} f_{\mathsf{A}} : a_{t+1} &= c_t \\ f_{\mathsf{B}} : b_{t+1} &= b_t \wedge \neg c_t \\ f_{\mathsf{C}} : c_{t+1} &= \neg c_t \end{cases}$$



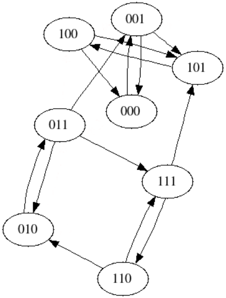
"A activates C"
"B interacts with itself"
"C activates A"
"C interacts with B"
"C inhibits itself"



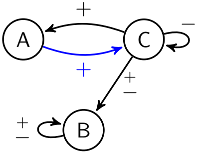
Boolean Networks – Synthesis from Knowledge and Data



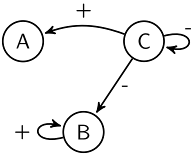
$$\mathscr{B} = \begin{cases} f_{\mathsf{A}} : a_{t+1} &= c_t \\ f_{\mathsf{B}} : b_{t+1} &= b_t \wedge \neg c_t \\ f_{\mathsf{C}} : c_{t+1} &= \neg c_t \end{cases}$$



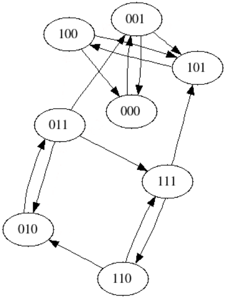
"A activates C"
"B interacts with itself"
"C activates A"
"C interacts with B"
"C inhibits itself"



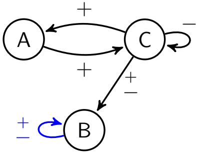
Boolean Networks – Synthesis from Knowledge and Data



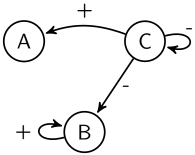
$$\mathscr{B} = \begin{cases} f_{\mathsf{A}} : a_{t+1} &= c_t \\ f_{\mathsf{B}} : b_{t+1} &= b_t \wedge \neg c_t \\ f_{\mathsf{C}} : c_{t+1} &= \neg c_t \end{cases}$$



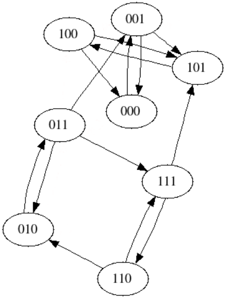
"A activates C"
"B interacts with itself"
"C activates A"
"C interacts with B"
"C inhibits itself"



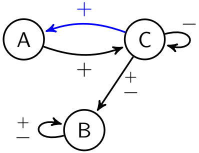
Boolean Networks – Synthesis from Knowledge and Data



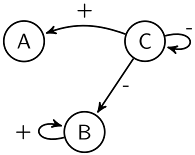
$$\mathscr{B} = \begin{cases} f_{\mathsf{A}} : a_{t+1} &= c_t \\ f_{\mathsf{B}} : b_{t+1} &= b_t \wedge \neg c_t \\ f_{\mathsf{C}} : c_{t+1} &= \neg c_t \end{cases}$$



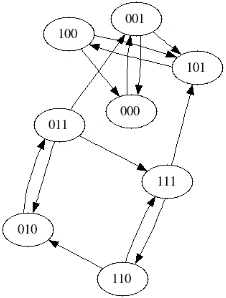
"A activates C"
"B interacts with itself"
"C activates A"
"C interacts with B"
"C inhibits itself"



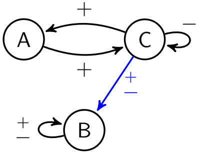
Boolean Networks – Synthesis from Knowledge and Data



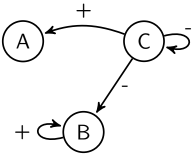
$$\mathscr{B} = \begin{cases} f_{\mathsf{A}} : a_{t+1} &= c_t \\ f_{\mathsf{B}} : b_{t+1} &= b_t \wedge \neg c_t \\ f_{\mathsf{C}} : c_{t+1} &= \neg c_t \end{cases}$$



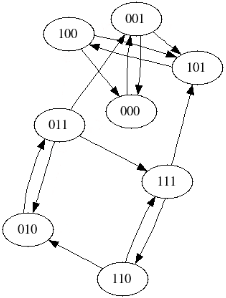
"A activates C"
"B interacts with itself"
"C activates A"
"C interacts with B"
"C inhibits itself"



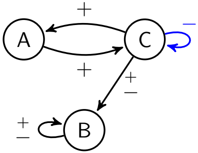
Boolean Networks – Synthesis from Knowledge and Data



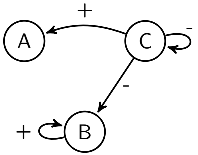
$$\mathscr{B} = \begin{cases} f_{\mathsf{A}} : a_{t+1} & = & c_t \\ f_{\mathsf{B}} : b_{t+1} & = & b_t \wedge \neg c_t \\ f_{\mathsf{C}} : c_{t+1} & = & \neg c_t \end{cases}$$



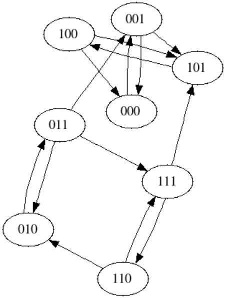
"A activates C"
"B interacts with itself"
"C activates A"
"C interacts with B"
"C inhibits itself"



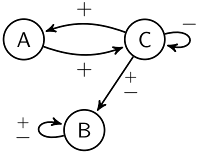
Boolean Networks – Synthesis from Knowledge and Data



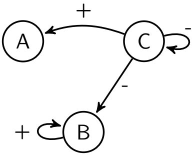
$$\mathscr{B} = \begin{cases} f_{\mathsf{A}} : a_{t+1} &= c_t \\ f_{\mathsf{B}} : b_{t+1} &= b_t \wedge \neg c_t \\ f_{\mathsf{C}} : c_{t+1} &= \neg c_t \end{cases}$$



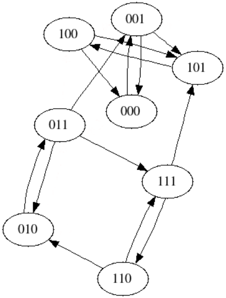
"A activates C"
"B interacts with itself"
"C activates A"
"C interacts with B"
"C inhibits itself"



Boolean Networks – Synthesis from Knowledge and Data



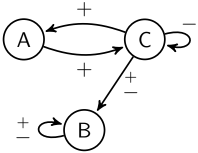
$$\mathscr{B} = \begin{cases} f_{\mathsf{A}} : a_{t+1} &= c_t \\ f_{\mathsf{B}} : b_{t+1} &= b_t \wedge \neg c_t \\ f_{\mathsf{C}} : c_{t+1} &= \neg c_t \end{cases}$$



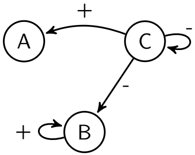
Prior Knowledge Network (PKN)

Super-set of influences allowed

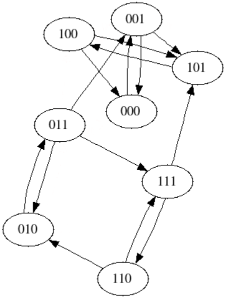
"A activates C"
"B interacts with itself"
"C activates A"
"C interacts with B"
"C inhibits itself"



Boolean Networks – Synthesis from Knowledge and Data



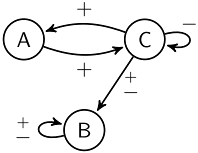
$$\mathscr{B} = \begin{cases} f_{\mathsf{A}} : a_{t+1} & = & c_t \\ f_{\mathsf{B}} : b_{t+1} & = & b_t \wedge \neg c_t \\ f_{\mathsf{C}} : c_{t+1} & = & \neg c_t \end{cases}$$



Prior Knowledge Network (PKN)

Super-set of influences allowed

"A activates C"
"B interacts with itself"
"C activates A"
"C interacts with B"
"C inhibits itself"

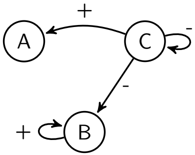


Multivariate Time Series (TS)

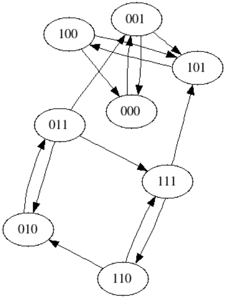
Concentrations of the components over time

t	1	2	3	4	5	6	7	8	9	10	11	12	13	...
A	0	3	7	13	20	30	49	61	100	63	36	25	2	...
B	100	86	64	57	54	53	51	49	45	37	33	28	22	...
C	0	27	36	42	60	75	54	44	38	48	60	72	88	...

Boolean Networks – Synthesis from Knowledge and Data



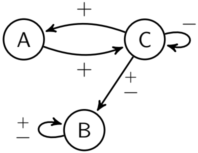
$$\mathscr{B} = \begin{cases} f_{\mathsf{A}} : a_{t+1} & = & c_t \\ f_{\mathsf{B}} : b_{t+1} & = & b_t \wedge \neg c_t \\ f_{\mathsf{C}} : c_{t+1} & = & \neg c_t \end{cases}$$



Prior Knowledge Network (PKN)

Super-set of influences allowed

"A activates C"
"B interacts with itself"
"C activates A"
"C interacts with B"
"C inhibits itself"

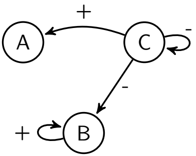


Multivariate Time Series (TS)

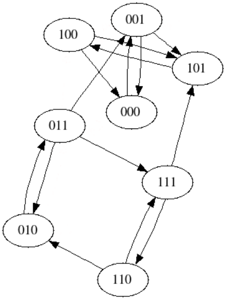
Concentrations of the components over time

t	1	2	3	4	5	6	7	8	9	10	11	12	13	...
A	0	3	7	13	20	30	49	61	100	63	36	25	2	...
B	100	86	64	57	54	53	51	49	45	37	33	28	22	...
C	0	27	36	42	60	75	54	44	38	48	60	72	88	...

Boolean Networks – Synthesis from Knowledge and Data

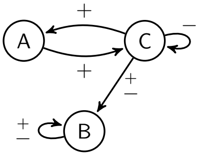


$$\mathscr{B} = \begin{cases} f_{\mathsf{A}} : a_{t+1} & = & c_t \\ f_{\mathsf{B}} : b_{t+1} & = & b_t \wedge \neg c_t \\ f_{\mathsf{C}} : c_{t+1} & = & \neg c_t \end{cases}$$



Prior Knowledge Network (PKN)
Super-set of influences allowed

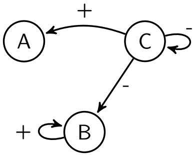
"A activates C"
"B interacts with itself"
"C activates A"
"C interacts with B"
"C inhibits itself"



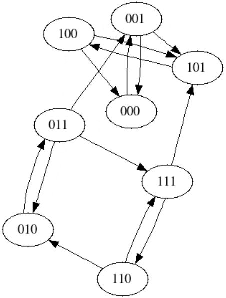
Multivariate Time Series (TS)
Concentrations of the components over time

t	1	2	3	4	5	6	7	8	9	10	11	12	13	...
A	0	3	7	13	20	30	49	61	100	63	36	25	2	...
B	100	86	64	57	54	53	51	49	45	37	33	28	22	...
C	0	27	36	42	60	75	54	44	38	48	60	72	88	...

Boolean Networks – Synthesis from Knowledge and Data



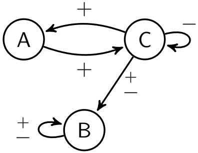
$$\mathscr{B} = \begin{cases} f_{\mathsf{A}} : a_{t+1} & = & c_t \\ f_{\mathsf{B}} : b_{t+1} & = & b_t \wedge \neg c_t \\ f_{\mathsf{C}} : c_{t+1} & = & \neg c_t \end{cases}$$



Prior Knowledge Network (PKN)

Super-set of influences allowed

"A activates C"
"B interacts with itself"
"C activates A"
"C interacts with B"
"C inhibits itself"

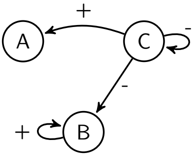


Multivariate Time Series (TS)

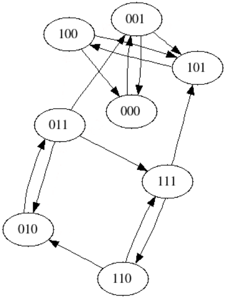
Concentrations of the components over time

t	1	2	3	4	5	6	7	8	9	10	11	12	13	...
A	0	3	7	13	20	30	49	61	100	63	36	25	2	...
B	100	86	64	57	54	53	51	49	45	37	33	28	22	...
C	0	27	36	42	60	75	54	44	38	48	60	72	88	...

Boolean Networks – Synthesis from Knowledge and Data



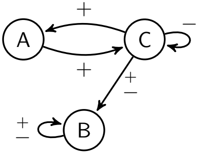
$$\mathscr{B} = \begin{cases} f_{\mathsf{A}} : a_{t+1} & = & c_t \\ f_{\mathsf{B}} : b_{t+1} & = & b_t \wedge \neg c_t \\ f_{\mathsf{C}} : c_{t+1} & = & \neg c_t \end{cases}$$



Prior Knowledge Network (PKN)

Super-set of influences allowed

"A activates C"
"B interacts with itself"
"C activates A"
"C interacts with B"
"C inhibits itself"

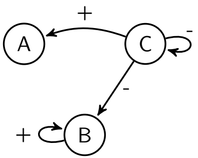


Multivariate Time Series (TS)

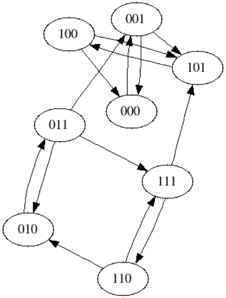
Concentrations of the components over time

t	1	2	3	4	5	6	7	8	9	10	11	12	13	...
A	0	3	7	13	20	30	49	61	100	63	36	25	2	...
B	100	86	64	57	54	53	51	49	45	37	33	28	22	...
C	0	27	36	42	60	75	54	44	38	48	60	72	88	...

Boolean Networks – Synthesis from Knowledge and Data



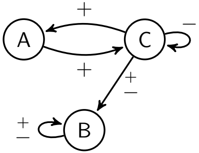
$$\mathscr{B} = \begin{cases} f_{\mathsf{A}} : a_{t+1} & = & c_t \\ f_{\mathsf{B}} : b_{t+1} & = & b_t \wedge \neg c_t \\ f_{\mathsf{C}} : c_{t+1} & = & \neg c_t \end{cases}$$



Prior Knowledge Network (PKN)

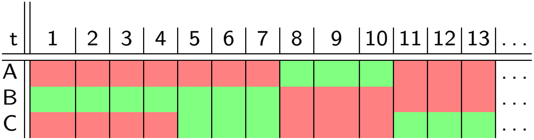
Super-set of influences allowed

"A activates C"
"B interacts with itself"
"C activates A"
"C interacts with B"
"C inhibits itself"

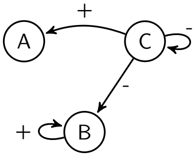


Multivariate Time Series (TS)

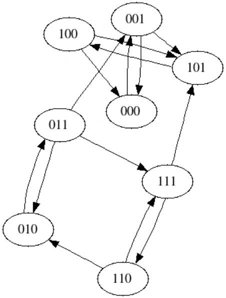
Concentrations of the components over time



Boolean Networks – Synthesis from Knowledge and Data



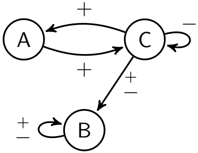
$$\mathscr{B} = \begin{cases} f_{\mathsf{A}} : a_{t+1} & = & c_t \\ f_{\mathsf{B}} : b_{t+1} & = & b_t \wedge \neg c_t \\ f_{\mathsf{C}} : c_{t+1} & = & \neg c_t \end{cases}$$



Prior Knowledge Network (PKN)

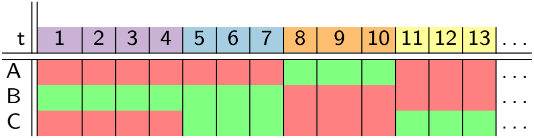
Super-set of influences allowed

"A activates C"
"B interacts with itself"
"C activates A"
"C interacts with B"
"C inhibits itself"

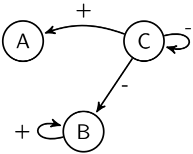


Multivariate Time Series (TS)

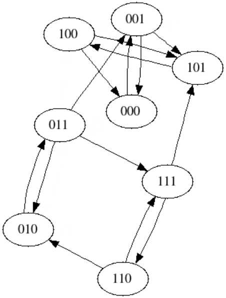
Concentrations of the components over time



Boolean Networks – Synthesis from Knowledge and Data



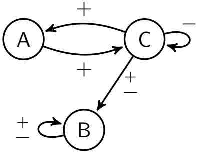
$$\mathscr{B} = \begin{cases} f_{\mathsf{A}} : a_{t+1} & = & c_t \\ f_{\mathsf{B}} : b_{t+1} & = & b_t \wedge \neg c_t \\ f_{\mathsf{C}} : c_{t+1} & = & \neg c_t \end{cases}$$



Prior Knowledge Network (PKN)

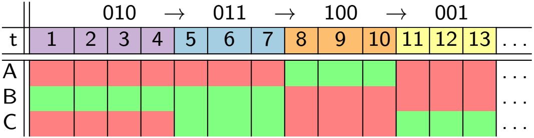
Super-set of influences allowed

"A activates C"
"B interacts with itself"
"C activates A"
"C interacts with B"
"C inhibits itself"

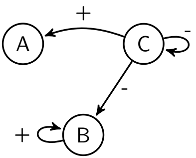


Multivariate Time Series (TS)

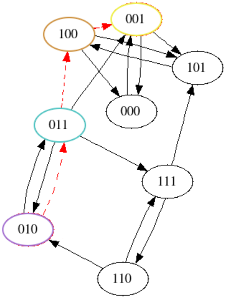
Concentrations of the components over time



Boolean Networks – Synthesis from Knowledge and Data



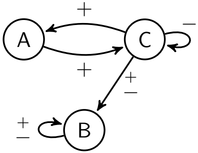
$$\mathscr{B} = \begin{cases} f_{\mathsf{A}} : a_{t+1} & = & c_t \\ f_{\mathsf{B}} : b_{t+1} & = & b_t \wedge \neg c_t \\ f_{\mathsf{C}} : c_{t+1} & = & \neg c_t \end{cases}$$



Prior Knowledge Network (PKN)

Super-set of influences allowed

"A activates C"
"B interacts with itself"
"C activates A"
"C interacts with B"
"C inhibits itself"

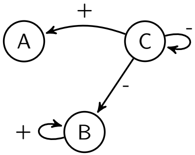


Multivariate Time Series (TS)

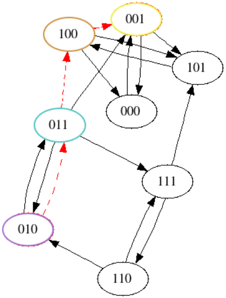
Concentrations of the components over time

	010				→ 011			→ 100			→ 001			
t	1	2	3	4	5	6	7	8	9	10	11	12	13	...
A														...
B														...
C														...

Boolean Networks – Synthesis from Knowledge and Data



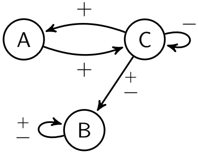
$$\mathscr{B} = \begin{cases} f_{\mathsf{A}} : a_{t+1} & = & c_t \\ f_{\mathsf{B}} : b_{t+1} & = & b_t \wedge \neg c_t \\ f_{\mathsf{C}} : c_{t+1} & = & \neg c_t \end{cases}$$



Prior Knowledge Network (PKN)

Super-set of influences allowed

"A activates C"
"B interacts with itself"
"C activates A"
"C interacts with B"
"C inhibits itself"



Multivariate Time Series (TS)

Concentrations of the components over time

	010				→	011		→	100		→	001		
t	1	2	3	4	5	6	7	8	9	10	11	12	13	...
A														...
B														...
C														...

Get the best coverage possible

Existing tools

Existing tools

Automatic synthesis of BNs from a Prior Knowledge Network (PKN)
and a multivariate Time Series (TS)
= hard problem (combinatorial explosion)

`REVEAL`
`Best-Fit`
`caspo-TS`

Existing tools

Automatic synthesis of BNs from a Prior Knowledge Network (PKN)
and a multivariate Time Series (TS)
= hard problem (combinatorial explosion)

	PKN	TS	assumptions
`REVEAL`	unsigned	binarized	time unit = 1
`Best-Fit`	unsigned	binarized	time unit = 1
`caspo-TS`	signed	numeric	local partial-monotony

Existing tools

Automatic synthesis of BNs from a Prior Knowledge Network (PKN)
and a multivariate Time Series (TS)
= hard problem (combinatorial explosion)

	PKN	TS	assumptions
REVEAL	unsigned	binarized	time unit = 1
Best-Fit	unsigned	binarized	time unit = 1
caspo-TS	signed	numeric	local partial-monotony
ASKeD-BN	signed	both	

Existing tools

Automatic synthesis of BNs from a Prior Knowledge Network (PKN)
and a multivariate Time Series (TS)
= hard problem (combinatorial explosion)

	PKN	TS	assumptions
`REVEAL`	unsigned	binarized	time unit = 1
`Best-Fit`	unsigned	binarized	time unit = 1
`caspo-TS`	signed	numeric	local partial-monotony
`ASKeD-BN`	signed	both	

Similar principles:

1. delimitation of the search space using the PKN as constraint
2. optimization of a criteria which measure the adequacy of the synthesized BNs with the TS (coverage)

Existing tools

Automatic synthesis of BNs from a Prior Knowledge Network (PKN)
and a multivariate Time Series (TS)
= hard problem (combinatorial explosion)

	PKN	TS	assumptions
`REVEAL`	unsigned	binarized	time unit = 1
`Best-Fit`	unsigned	binarized	time unit = 1
`caspo-TS`	signed	numeric	local partial-monotony
`ASKeD-BN`	signed	both	

Similar principles:

1. delimitation of the search space using the PKN as constraint
2. optimization of a criteria which measure the adequacy of the synthesized BNs with the TS (coverage)

Existing tools

Automatic synthesis of BNs from a Prior Knowledge Network (PKN)
and a multivariate Time Series (TS)
= hard problem (combinatorial explosion)

	PKN	TS	assumptions
`REVEAL`	unsigned	binarized	time unit = 1
`Best-Fit`	unsigned	binarized	time unit = 1
`caspo-TS`	signed	numeric	local partial-monotony
`ASKeD-BN`	signed	both	

Similar principles:

1. delimitation of the search space using the PKN as constraint
2. optimization of a criteria which measure the adequacy of the synthesized BNs with the TS (coverage)

Multiple optimal solutions are all returned

ASKeD-BN

ASKeD-BN

Formulation the BN synthesis problem as a logic program with the Answer Set Programming (ASP) framework

ASKeD-BN

Formulation the BN synthesis problem as a logic program with the Answer Set Programming (ASP) framework

Main parts of the logic program:

1. generates all the possible candidate functions
2. removes the ones that do not respect the PKN
3. acts like an exhaustive evaluation of all the candidates and returns the *parsimonious* candidates which explain best the *binarized* observations from the given time series

ASKeD-BN

Formulation the BN synthesis problem as a logic program with the Answer Set Programming (ASP) framework

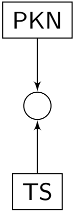
Main parts of the logic program:

1. generates **all the possible** candidate functions
2. removes the ones that do not respect the PKN
3. **acts like** an exhaustive evaluation of all the candidates and returns the *parsimonious* candidates which explain best the *binarized* observations from the given time series

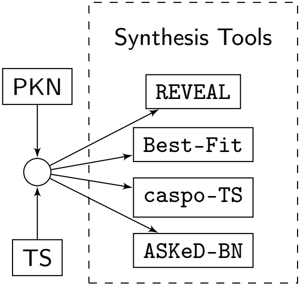
ASP solvers are designed to solve hard combinatorial satisfaction problem. They prune the search space *efficiently* (heuristic from SAT solvers).

Evaluation Procedure

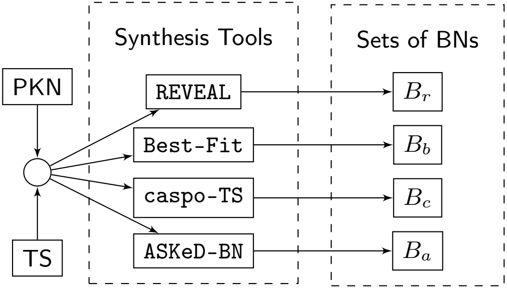
Evaluation Procedure



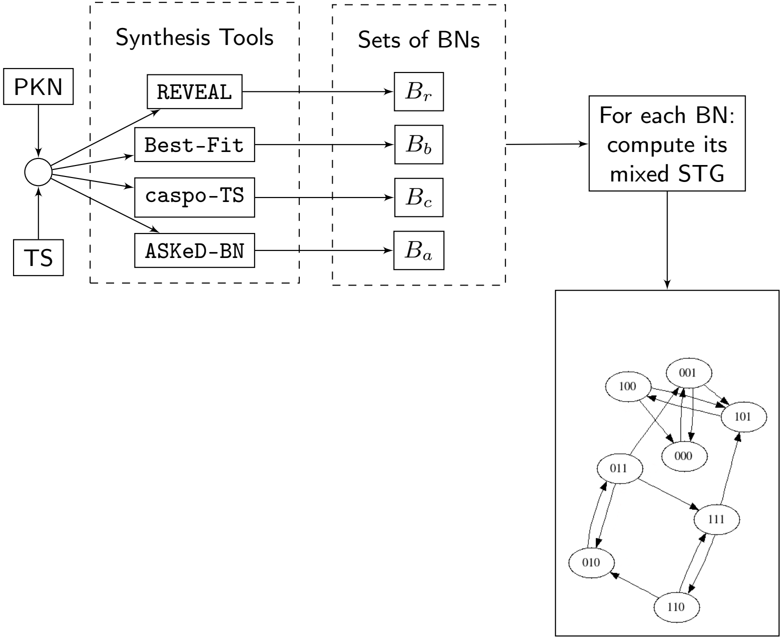
Evaluation Procedure



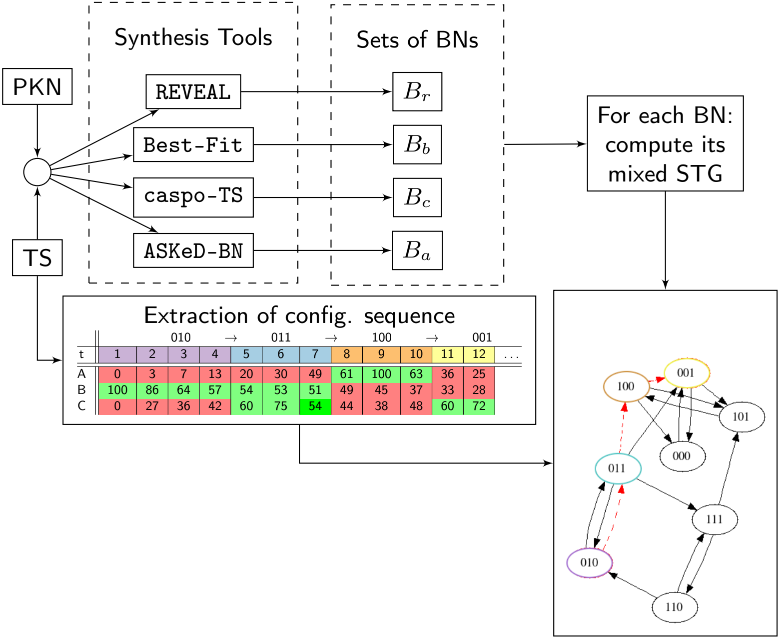
Evaluation Procedure



Evaluation Procedure

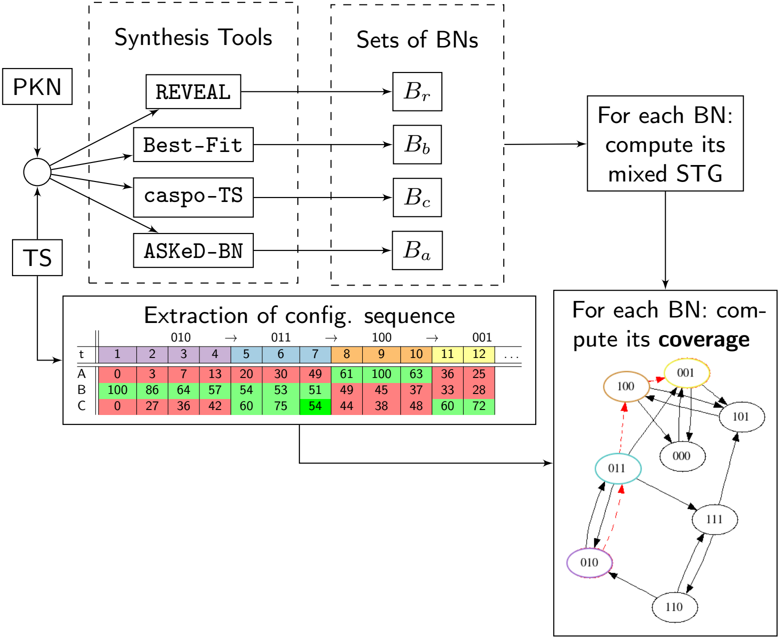


Evaluation Procedure



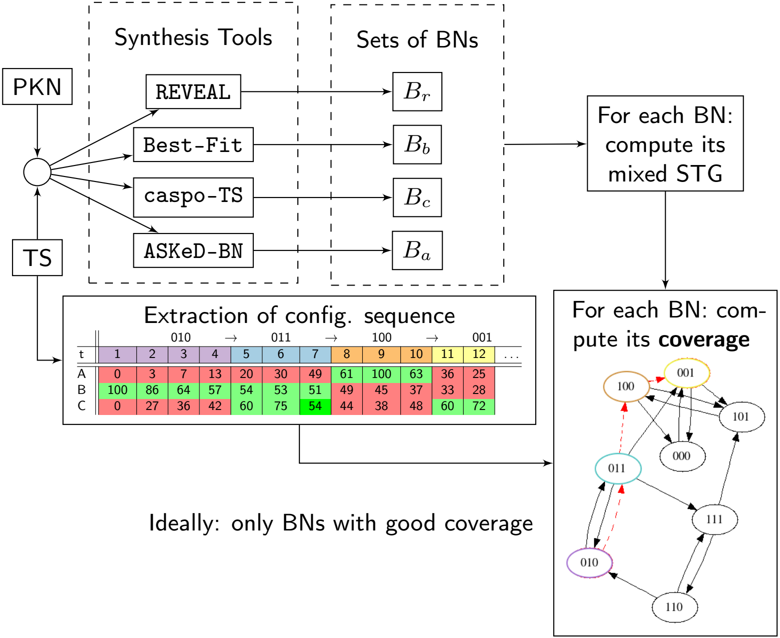
	010				→ 011			→ 100			→ 001		
t	1	2	3	4	5	6	7	8	9	10	11	12	...
A	0	3	7	13	20	30	49	61	100	63	36	25	
B	100	86	64	57	54	53	51	49	45	37	33	28	
C	0	27	36	42	60	75	54	44	38	48	60	72	

Evaluation Procedure



	010			→	011			→	100		→	001	
t	1	2	3	4	5	6	7	8	9	10	11	12	...
A	0	3	7	13	20	30	49	61	100	63	36	25	
B	100	86	64	57	54	53	51	49	45	37	33	28	
C	0	27	36	42	60	75	54	44	38	48	60	72	

Evaluation Procedure



	010			→	011		→	100		→	001	
t	1	2	3	4	5	6	7	8	9	10	11	12	...
A	0	3	7	13	20	30	49	61	100	63	36	25
B	100	86	64	57	54	53	51	49	45	37	33	28
C	0	27	36	42	60	75	54	44	38	48	60	72

Evaluation on Real Datasets

Evaluation on Real Datasets

yeast

A. thaliana

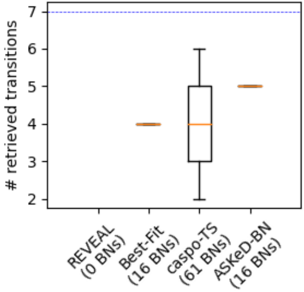
Evaluation on Real Datasets

yeast
4 components, 7 transitions

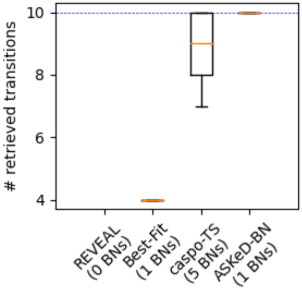
A. thaliana
5 components, 10 transitions

Evaluation on Real Datasets

yeast
4 components, 7 transitions

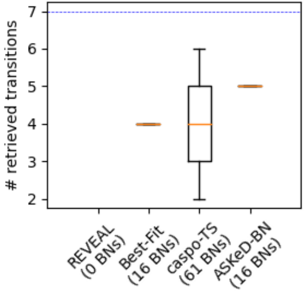


A. thaliana
5 components, 10 transitions



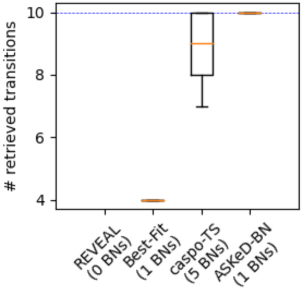
Evaluation on Real Datasets

yeast
4 components, 7 transitions



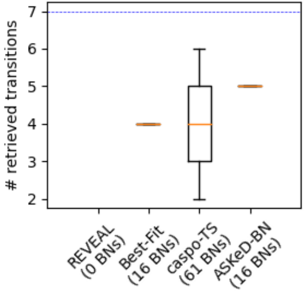
- `REVEAL` fails

A. thaliana
5 components, 10 transitions

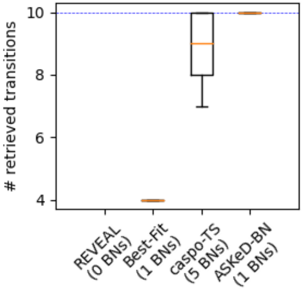


Evaluation on Real Datasets

yeast
4 components, 7 transitions



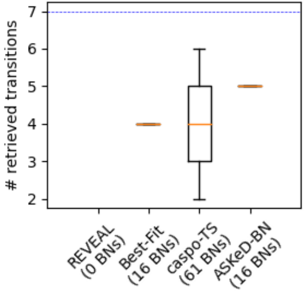
A. thaliana
5 components, 10 transitions



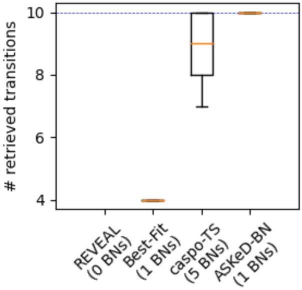
- `REVEAL` fails
- `Best-Fit` lacks consistency

Evaluation on Real Datasets

yeast
4 components, 7 transitions



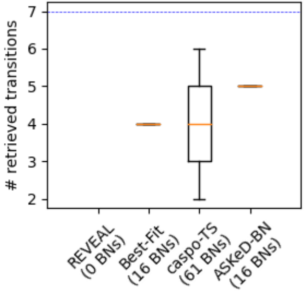
A. thaliana
5 components, 10 transitions



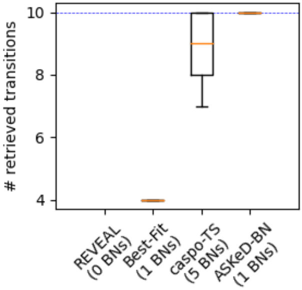
- `REVEAL` fails
- `Best-Fit` lacks consistency
- `caspo-TS` and `ASKeD-BN` find good BNs

Evaluation on Real Datasets

yeast
4 components, 7 transitions



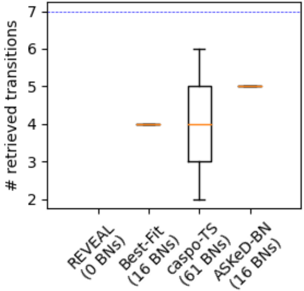
A. thaliana
5 components, 10 transitions



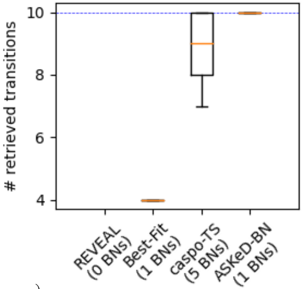
- `REVEAL` fails
- `Best-Fit` lacks consistency
- `caspo-TS` and `ASKeD-BN` find good BNs
- `caspo-TS` returns more BNs,
  some of them with poor coverage.

Evaluation on Real Datasets

yeast
4 components, 7 transitions



A. thaliana
5 components, 10 transitions

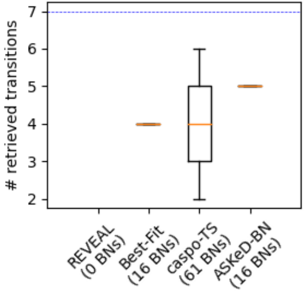


- `REVEAL` fails
- `Best-Fit` lacks consistency
- `caspo-TS` and `ASKeD-BN` find good BNs
- `caspo-TS` returns more BNs, some of them with poor coverage.

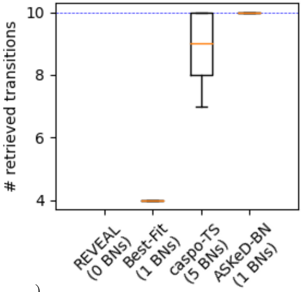
$\Big\}$ confirmed on synthetic datasets

Evaluation on Real Datasets

yeast
4 components, 7 transitions



A. thaliana
5 components, 10 transitions



- `REVEAL` fails
- `Best-Fit` lacks consistency
- `caspo-TS` and `ASKeD-BN` find good BNs
- `caspo-TS` returns more BNs, some of them with poor coverage.

$\Big\}$ confirmed on synthetic datasets

$\rightarrow$ `ASKeD-BN` returns a small number of BN, with good coverage and low variance ✓

Conclusion

Contribution:

- `ASKeD-BN`: Automatic Synthesis of Boolean Networks constrained in their structure (PKN knowledge) and their dynamics (TS data)
- Approach free of strong / restraining assumptions
- Formulation as a logic program (Answer-Set Programming)
- `ASKeD-BN` gives good results

All data + code available at:
`https://gitlab.inria.fr/avaginay/OLA2021`

Work in progress:

- Apply `ASKeD-BN` on PKN and TS directly extracted from existing biological models (ODE-like)

The end. Any questions ?

Annexe

Datasets for Evaluation

2 real datasets:

System	PKN		TS	
	# nodes	# edges	# time steps	# transitions
yeast (cell cycle)	4	28	14	6
A. thaliana (circadian clock)	5	8	50	11

6 synthetic datasets:

Various complexity: from 3 to 10 nodes.
Various conditions: synch. or async. update scheme, with or without repetition, with or without noise

336 experiments at total including 42 with the ARN setting.

Datasets for Evaluation

2 real datasets:

System	PKN		TS	
	# nodes	# edges	# time steps	# transitions
yeast (cell cycle)	4	28	14	6
A. thaliana (circadian clock)	5	8	50	11

6 synthetic datasets:

Various complexity: from 3 to 10 nodes.
Various conditions: synch. or **async.** update scheme, **with** or without **repetition**, **with** or without **noise** (“**ARN**”)

336 experiments at total including 42 with the ARN setting.

Comparison time and RAM

yest

method	running time (s)	cputime (s)	max_rss (MB)
REVEAL	1.0095	0.55	72.97
Best-Fit	1.4069	1.10	92.77
caspo-TS	24.6545	12.91	183.08
ASKeD-BN	5.4209	4.90	186.80

A. thaliana

method	running time (s)	cputime (s)	max_rss (MB)
caspo-TS	7.0394	1.85	139.93
ASKeD-BN	8.5820	8.19	163.38

observed in general:

- caspo-TS faster and less RAM usage than ASKeD-BN
- but not on some datasets, where ASKeD-BN terminated in less than 100 hours while caspo-TS is still running after more than 300 hours. ← to investigate...

Synthetic Data – Comparison of the Number of BN Returned

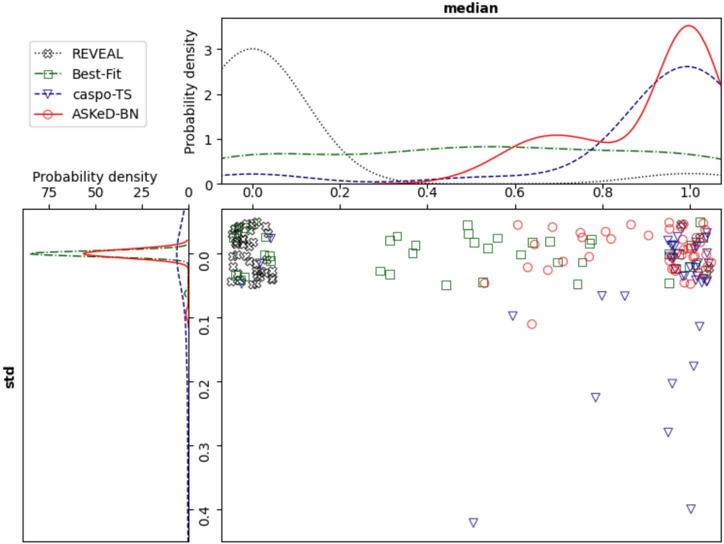
336 experiments at total including 42 with the ARN* setting.

	setting	REVEAL before filter	REVEAL after filter	Best-Fit before filter	Best-Fit after filter	caspo-TS	ASKeD-BN
# failing xp	all	230	240	0	64	20	0
# total returned BNs	all	100 677 500	406	100 678 198	724	8481	1210
# total returned BNs	ARN	3	3	51	35	720	85

- REVEAL often fails
- REVEAL and Best-Fit return a lots of BNs which are not respecting the PKN
- caspo-TS returns in average between 5 and 7 times more BNs than ASKeD-BN (depending on the setting)

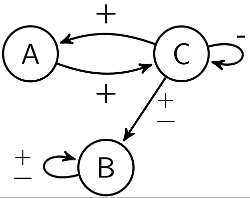
*: ARN = Asyn. update scheme, with repetition and noise

Synthetic data, ARN setting – Quality of the BNs



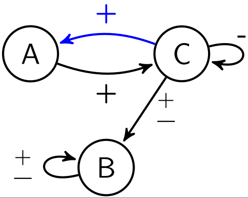
ASKeD-BN: Hard Constraints

"A activates C"
"B interacts with itself"
"C activates A"
"C interacts with B"
"C inhibits itself"



ASKeD-BN: Hard Constraints

"A activates C"
"B interacts with itself"
"C activates A"
"C interacts with B"
"C inhibits itself"



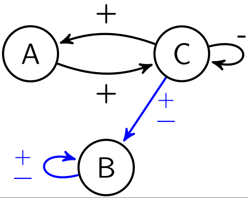
For A:

3 choices:
A := C
A := 0
A := 1

but not:
A := B
A := $\neg$C

ASKeD-BN: Hard Constraints

"A activates C"
"B interacts with itself"
"C activates A"
"C interacts with B"
"C inhibits itself"



For A:

3 choices:
A $:=$ C
A $:= 0$
A $:= 1$

but not:
A $:=$ B
A $:= \neg$C

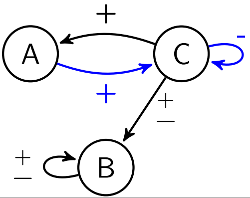
For B:

16 choices:
B $:=$ B $\wedge \neg$C
B $:=$ (B $\wedge \neg$C) $\vee$ ($\neg$B $\wedge$ C);
...
B $:= 0$
B $:= 1$

but not:
B $:=$ A

ASKeD-BN: Hard Constraints

"A activates C"
"B interacts with itself"
"C activates A"
"C interacts with B"
"C inhibits itself"



For A:

3 choices:
A $:=$ C
A $:= 0$
A $:= 1$

but not:
A $:=$ B
A $:= \neg$C

For B:

16 choices:
B $:=$ B $\wedge \neg$C
B $:= ($B $\wedge \neg$C$) \vee (\neg$B $\wedge$ C$)$;
...
B $:= 0$
B $:= 1$

but not:
B $:=$ A

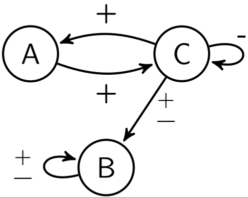
For C:

6 choices:
C $:= \neg$C
C $:=$ A
...
C $:= 0$
C $:= 1$

but not:
C $:=$ A $\wedge$ B

ASKeD-BN: Hard Constraints

"A activates C"
"B interacts with itself"
"C activates A"
"C interacts with B"
"C inhibits itself"



For A:

3 choices:
A := C
A := 0
A := 1

but not:
A := B
A := ¬C

For B:

16 choices:
B := B ∧ ¬C
B := (B ∧ ¬C) ∨ (¬B ∧ C);
...
B := 0
B := 1

but not:
B := A

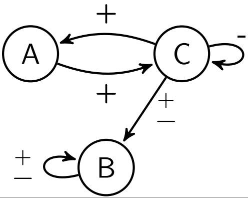
For C:

6 choices:
C := ¬C
C := A
...
C := 0
C := 1

but not:
C := A ∧ B

ASKeD-BN: Hard Constraints

"A activates C"
"B interacts with itself"
"C activates A"
"C interacts with B"
"C inhibits itself"



For A:

3 choices:
A $:=$ C
A $:= 0$
A $:= 1$

but not:
A $:=$ B
A $:= \neg$C

For B:

16 choices:
$\boxed{\text{B} := \text{B} \wedge \neg\text{C}}$
$\boxed{\text{B} := (\text{B} \wedge \neg\text{C}) \vee (\neg\text{B} \wedge \text{C});}$
...
B $:= 0$
B $:= 1$

but not:
B $:=$ A

For C:

6 choices:
$\boxed{\text{C} := \neg\text{C}}$
$\boxed{\text{C} := \text{A}}$
...
C $:= 0$
C $:= 1$

but not:
C $:=$ A $\wedge$ B

ASKeD-BN: Soft constraints — Example 1

	010			→	011			→	100		→	001								
t	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A	0	3	7	13	20	30	49	61	100	63	36	25	2	3	1	1	3	0	0	0
B	100	86	64	57	54	53	51	49	45	37	33	28	22	19	14	12	9	5	2	0
C	0	27	36	42	60	75	54	44	38	48	60	72	88	90	100	100	100	100	100	100

$\mathscr{U}$ set of unexplained timesteps

Mean Absolute Error $\mathsf{MAE_X} = \frac{\sum_{t' \in \mathscr{U}} |\theta_{\mathsf{X}} - x_{t'}|}{T}$

	$a_{t+1} = c_t$	✓	$a_{t+1} = 0$
$\mathscr{U}$	$\emptyset$		$\{8\}$
MAE	0	✓	0.55

ASKeD-BN: Soft constraints — Example 2

	010			→	011		→	100		→	001									
t	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A	0	3	7	13	20	30	49	61	100	63	36	25	2	3	1	1	3	0	0	0
B	100	86	64	57	54	53	51	49	45	37	33	28	22	19	14	12	9	5	2	0
C	0	27	36	42	60	75	54	44	38	48	60	72	88	90	100	100	100	100	100	100

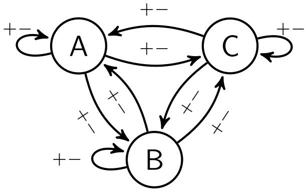
$\mathscr{U}$ set of unexplained timesteps

Mean Absolute Error $\mathsf{MAE}_{\mathsf{X}} = \frac{\sum_{t' \in \mathscr{U}} |\theta_{\mathsf{X}} - x_{t'}|}{T}$

	$b_{t+1} = b_t \wedge \neg \mathsf{c}_t$	✓	$b_{t+1} = (b_t \wedge \neg c_t) \vee (\neg b_t \wedge c_t)$	
$\mathscr{U}$	$\emptyset$		$\emptyset$	
MAE	0	✓	0	✓
# influences	2	✓	4	

How does the PKN help reducing the search space?

Without information



$2^{2^3} = 256$ candidates for each components

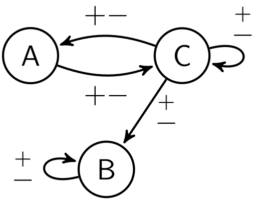
$\rightarrow 256 \times 256 \times 256 = 16777216$ candidate BNs

How does the PKN help reducing the search space?

With PKN

directions only

(`REVEAL` & `Best-Fit`)



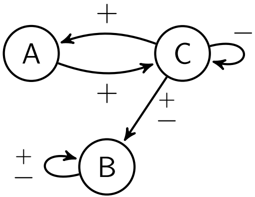
A	B	C
4	16	16

$\rightarrow 4 \times 16 \times 16 = 1024$

candidate BNs

direction + signs

(`caspo-TS` & `ASKeD-BN`)



	A	B	C
all	3	16	6
monotonous	3	14	6

$\rightarrow 3 \times 16 \times 6 = 288$

candidate BNs

including 252 locally

partial-monotonous.