Learning Dynamical Interactions Within Marine Ecosystems with LFIT

Tony Ribeiro[1,2,3], Maxime Folschette[4], Omar Iken[4]

1. Independant Researcher
2. Université de Nantes, Centrale Nantes, CNRS, LS2N, F-44000 Nantes, France
3. National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan
4. Univ. Lille, CNRS, Centrale Lille, UMR 9189 CRIStAL, F-59000 Lille, France

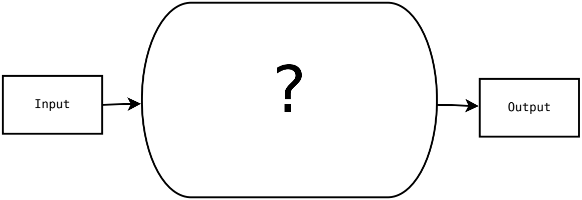
4th February 2022, Bioss, Online

Outline

Outline

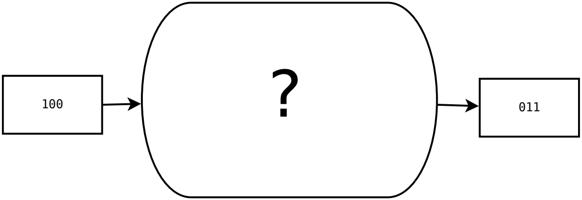
Research area

Idea: given a set of input/output states of a black-box system, learn its internal mechanics.



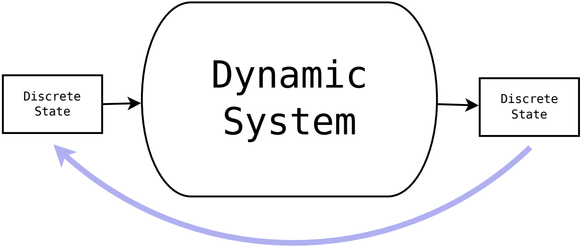
Research area

Discrete system: input/output are vectors of same size which contain discrete values.



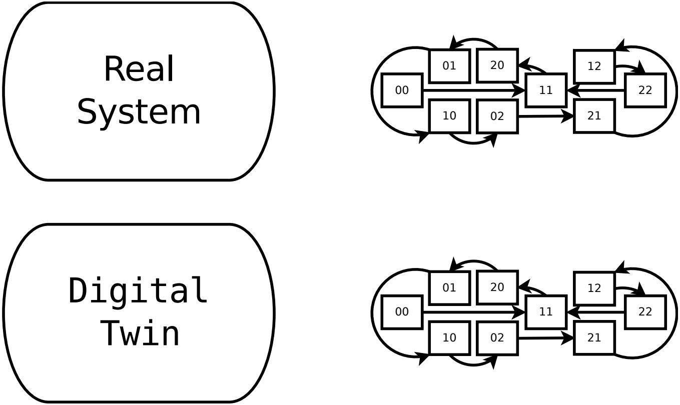
Research area

Dynamic system: input/output are states of the system and output becomes the next input.



Research area

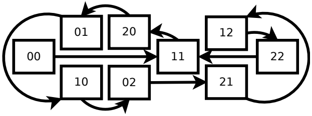
Goal: produce an artificial system with the same behavior as the one observed, i.e., a digital twin.



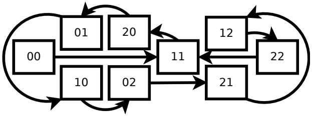
Research area

Representation: propositional logic programs with annotated atoms encoding multi-valued variables.

Real System



Logic Program

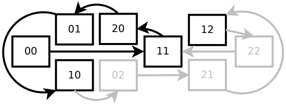


Research area

Method: learn the dynamics of systems from the observations of some of its state transitions.

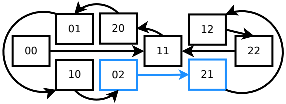
DATA

?



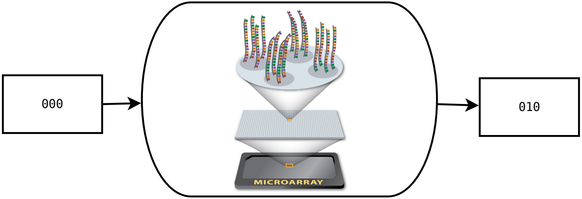
RESULTS

a(0,T) :- a(2,T-1)
a(1,T) :- a(0,T-1), b(0,T-1).
a(2,T) :- a(1,T-1)
a(2,T) :- a(0,T-1), b(2,T-1).
...
b(0,T) :- a(1,T-1).
b(1,T) :- b(2,T-1).
b(2,T):- b(2,T-1).
...



Motivation

Data: time series of gene expression levels in a organic cell.
Goal: model gene interactions to understand their influences.

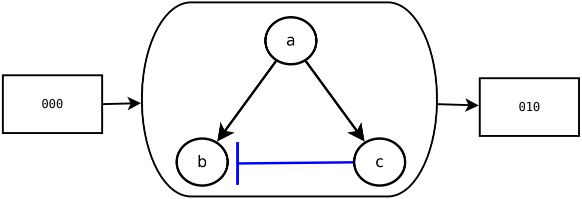


Example (Possible Applications)

- **Bioinformatics**: Construct gene regulatory networks.
- **Robotics**: Learn action models from robot observations.

Motivation

Data: time series of gene expression levels in a organic cell.
Goal: model gene interactions to understand their influences.

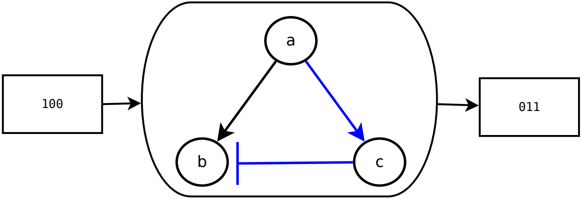


Example (Possible Applications)

- **Bioinformatics**: Construct gene regulatory networks.
- **Robotics**: Learn action models from robot observations.

Motivation

Data: time series of gene expression levels in a organic cell.
Goal: model gene interactions to understand their influences.

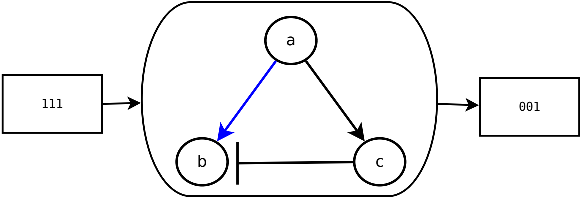


Example (Possible Applications)

- **Bioinformatics**: Construct gene regulatory networks.
- **Robotics**: Learn action models from robot observations.

Motivation

Data: time series of gene expression levels in a organic cell.
Goal: model gene interactions to understand their influences.

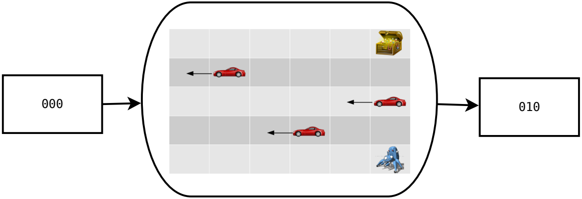


Example (Possible Applications)

- **Bioinformatics**: Construct gene regulatory networks.
- **Robotics**: Learn action models from robot observations.

Motivation

Data: observations of environment evolution according to a robot actions.
Goal: produce a predictive model of the environment for action planning.



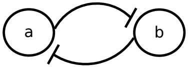
Example (Possible Applications)

- **Bioinformatics**: Construct gene regulatory networks.
- **Robotics**: Learn action models from robot observations.

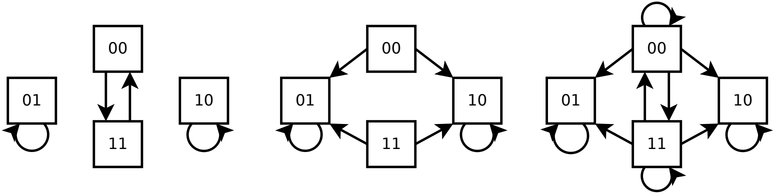
Outline

Dynamical Semantics

Boolean network transitions differ according to the update semantics used.



f(a) := not b.
f(b) := not a.

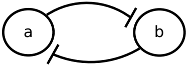


Synchronous   Asynchronous   General

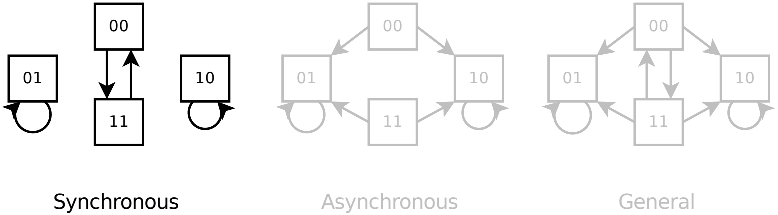
- Synchronous: all variables are updated
- Asynchronous: only one variable is updated
- General: any number of variables can be updated

Dynamical Semantics

Boolean network transitions differ according to the update semantics used.



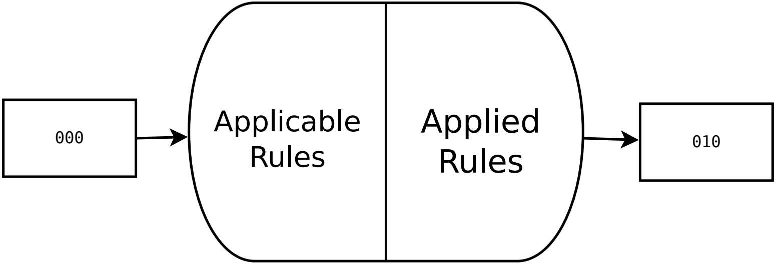
f(a) := not b.
f(b) := not a.



- Synchronous: all variables are updated
- Asynchronous: only one variable is updated
- General: any number of variables can be updated

What is a semantics?

For those three semantics at least, it is about computing the next state by selecting among applicable local rules the ones that will be applied.

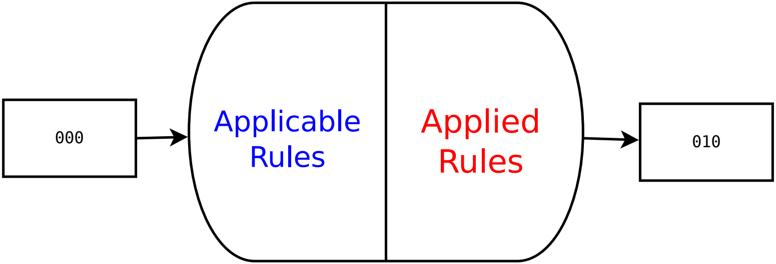


Semantics: what is an applicable rule and what is a valid set of applied rule.

The three semantics that are considered here differ on the selection but share the same definition of what is an applicable rule.

What is a semantics?

For those three semantics at least, it is about computing the next state by selecting among applicable local rules the ones that will be applied.



Semantics: what is an applicable rule and what is a valid set of applied rule.

The three semantics that are considered here differ on the selection but share the same definition of what is an applicable rule.

Learning algorithm intuition: classification problem

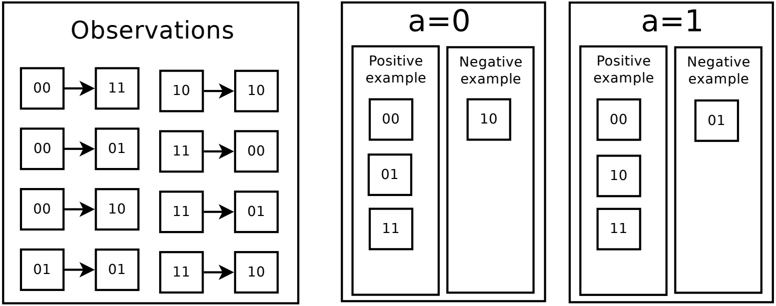
What is an applicable rule?

Learning algorithm intuition: classification problem

What is an applicable rule? The conditions so that a variable can take a certain value in next state.

Learning algorithm intuition: classification problem

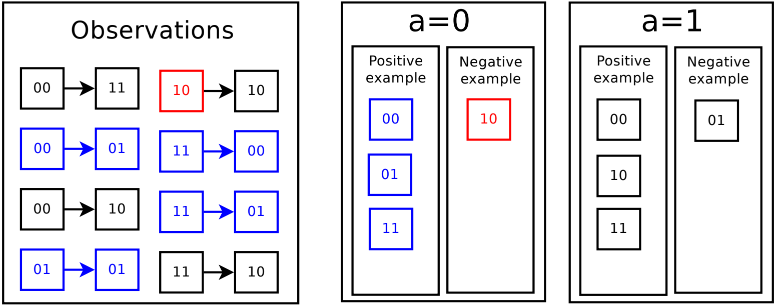
What is an applicable rule? The conditions so that a variable can take a certain value in next state.



Equivalent to a classification problem: for each value of a variable, what is a typical state where the variable can takes this value in the next state ?

Learning algorithm intuition: classification problem

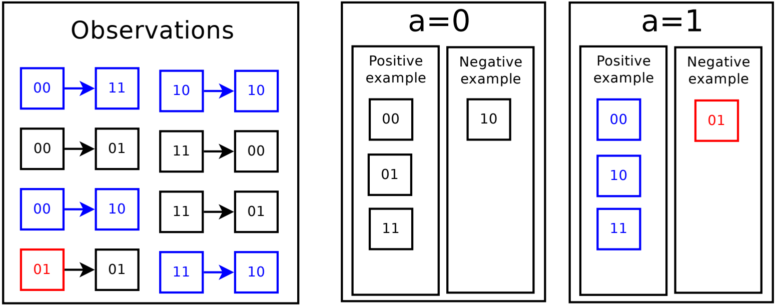
What is an applicable rule? The conditions so that a variable can take a certain value in next state.



Equivalent to a classification problem: for each value of a variable, what is a typical state where the variable can takes this value in the next state ?

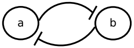
Learning algorithm intuition: classification problem

What is an applicable rule? The conditions so that a variable can take a certain value in next state.

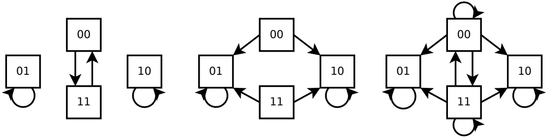


Equivalent to a classification problem: for each value of a variable, what is a typical state where the variable can takes this value in the next state ?

General Usage LFIT Algorithm (**GULA**) ouptut



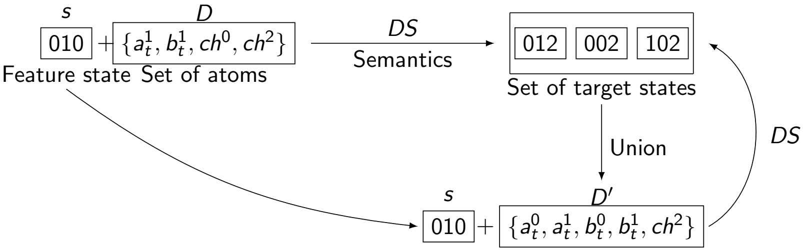
f(a) := not b.
f(b) := not a.



Synchronous	Asynchronous	General
// f(a) := not b	// f(a) := not b	// f(a) := not b
$a_t^0 \leftarrow b_{t-1}^1$	$a_t^0 \leftarrow b_{t-1}^1$	$a_t^0 \leftarrow b_{t-1}^1$
$a_t^1 \leftarrow b_{t-1}^0$	$a_t^1 \leftarrow b_{t-1}^0$	$a_t^1 \leftarrow b_{t-1}^0$
// f(b) := not a	// f(b) := not a	// f(b) := not a
$b_t^0 \leftarrow a_{t-1}^1$	$b_t^0 \leftarrow a_{t-1}^1$	$b_t^0 \leftarrow a_{t-1}^1$
$b_t^1 \leftarrow a_{t-1}^0$	$b_t^1 \leftarrow a_{t-1}^0$	$b_t^1 \leftarrow a_{t-1}^0$
	// Default rules	// Default rules
	$a_t^0 \leftarrow a_{t-1}^0$	$a_t^0 \leftarrow a_{t-1}^0$
	$a_t^1 \leftarrow a_{t-1}^1$	$a_t^1 \leftarrow a_{t-1}^1$
	$b_t^0 \leftarrow b_{t-1}^0$	$b_t^0 \leftarrow b_{t-1}^0$
	$b_t^1 \leftarrow b_{t-1}^1$	$b_t^1 \leftarrow b_{t-1}^1$

Pseudo-idempotent semantics

GULA can model observations from any pseudo-idempotent semantics.



$$\longrightarrow DS(s, D) = DS\Big(s, \bigcup_{s' \in DS(s,D)} s'\Big)$$

where DS is the dynamical semantics, and D is the head of rules of a multi-valued logic program that match the sate s.

Heuristic Model for Real Data

Modeling:

- Use GULA to learn two logic programs: possibilities/impossibilities.
- Weight each rule by the number of observations it matches.

Prediction:

- Likeliness: combine weight of matching possibility/impossibility rules
- Explanation: the rules used for the prediction

Example

Likeliness rules
$(3, a^0 \leftarrow b^1)$
$(15, a^1 \leftarrow b^0)$
...

Unlikeliness rules
$(30, a^0 \leftarrow c^1)$
$(5, a^1 \leftarrow c^0)$
...

Heuristic Model for Real Data

Modeling:

- Use GULA to learn two logic programs: possibilities/impossibilities.
- Weight each rule by the number of observations it matches.

Prediction:

- Likeliness: combine weight of matching possibility/impossibility rules
- Explanation: the rules used for the prediction

Example

Likeliness rules	Unlikeliness rules
$(3, a^0 \leftarrow b^1)$	$(30, a^0 \leftarrow c^1)$
$(15, a^1 \leftarrow b^0)$	$(5, a^1 \leftarrow c^0)$
...	...

$predict(target, feature\_state) = (\text{proba, explanations})$

$predict(a^0, \{a^1, b^1, c^1\}) = (0.5, (0, None), (0, None))$ Unconclusive

Heuristic Model for Real Data

Modeling:

- Use GULA to learn two logic programs: possibilities/impossibilities.
- Weight each rule by the number of observations it matches.

Prediction:

- Likeliness: combine weight of matching possibility/impossibility rules
- Explanation: the rules used for the prediction

Example

Likeliness rules	Unlikeliness rules
$(3, a^0 \leftarrow b^1)$	$(30, a^0 \leftarrow c^1)$
$(15, a^1 \leftarrow b^0)$	$(5, a^1 \leftarrow c^0)$
...	...

$predict(a^0, \{a^1, b^1, c^1\}) = (1.0, (3, a^0 \leftarrow b^1), (0, None))$ Possible

$predict(a^1, \{a^0, b^1, c^0\} = (0.0, (0, None), (5, a^1 \leftarrow c^0)$ Impossible

Heuristic Model for Real Data

Modeling:

- Use GULA to learn two logic programs: possibilities/impossibilities.
- Weight each rule by the number of observations it matches.

Prediction:

- Likeliness: combine weight of matching possibility/impossibility rules
- Explanation: the rules used for the prediction

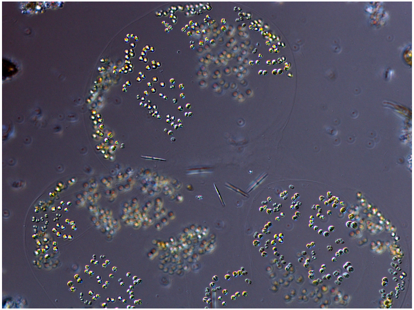
Example

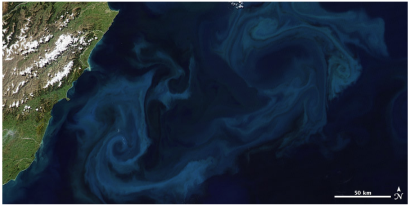
Likeliness rules	Unlikeliness rules
$(3, a^0 \leftarrow b^1)$	$(30, a^0 \leftarrow c^1)$
$(15, a^1 \leftarrow b^0)$	$(5, a^1 \leftarrow c^0)$
...	...

$predict(a^1, \{a^1, b^1, c^0\} = (0.75, (15, a^1 \leftarrow b^0), (5, a^1 \leftarrow c^0)$ Likely
$predict(a^0, \{a^1, b^1, c^0\} = (0.09, (3, a^0 \leftarrow b^1), (30, a^0 \leftarrow c^1)$ Unlikely

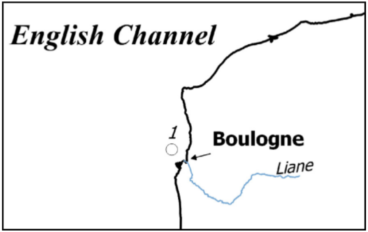
Outline

Phytoplankton Blooms





SRN Dataset



https://www.seanoe.org/data/00397/50832/

Sampling location	Sampling date	Taxon	Value	Sampling depth
001-P-015	1992-05-18	CHLOROA	6.0	Surface (0-1m)
006-P-001	2019-12-02	Chaetoceros	1000.0	Surface (0-1m)
002-P-007	1994-05-25	Pleurosigma	100.0	Surface (0-1m)
002-P-030	2005-10-19	SALI	34.83	Surface (0-1m)
006-P-007	2015-09-28	Guinardia delicatula	11400.0	Surface (0-1m)

Environmental variables (7) Phytoplankton (12)

Applying *LFIT*

Input

- 253 training transitions
- 53 testing transitions
- Features = Phytoplankton + Envir^t
- Targets = Phytoplankton

Output

- Run time = 2.35s (PRIDE)
- 1683 likeliness rules
- 1981 unlikeliness rules
- Model accuracy: **0.670**

Prediction example for phytoplankton specie `Cha`, in a given state:

- Level 0 with 5.3% probability (Non-deterministic dynamics $\Rightarrow$ Total $\neq$ 100%)
  - 1 match for $Cha(0) \leftarrow SALI(0), TEMP(0), TURB(1), Cha(0), Dit(0).$
  - 18 anti-matches for $Cha(0) \leftarrow TURB(1), Nit(2).$
- Level 1 with 87.5% probability
  - 7 matches for $Cha(1) \leftarrow SALI(0), TURB(1), Nit(2).$
  - 1 anti-match for $Cha(1) \leftarrow SALI(0), TEMP(0), TURB(1), Cha(0), Dit(0).$
- Level 2 with 2.6% probability
  - 1 match for $Cha(2) \leftarrow TEMP(0), Dit(0), Par(1), Pss(0), Thn(1).$
  - 37 anti-matches for $Cha(2) \leftarrow SALI(0), TURB(1).$

Model Improvement

Process: For each rule in the program, replace the body with a subset and weight this new rule on the dataset

Expectations: Noise removal, accuracy improvement, simpler rules

Likeliness rule: $Cha(0) \leftarrow CHLOROA(0), TEMP(1), Cha(0), Gud(2).$

Replace body with...	correct	wrong
$CHLOROA(0), TEMP(1), Cha(0), Gud(2)$	1	0
$CHLOROA(0), Cha(0), Gud(2)$	10	4
$TEMP(1), Cha(0)$	18	5
$Gud(2)$	25	2

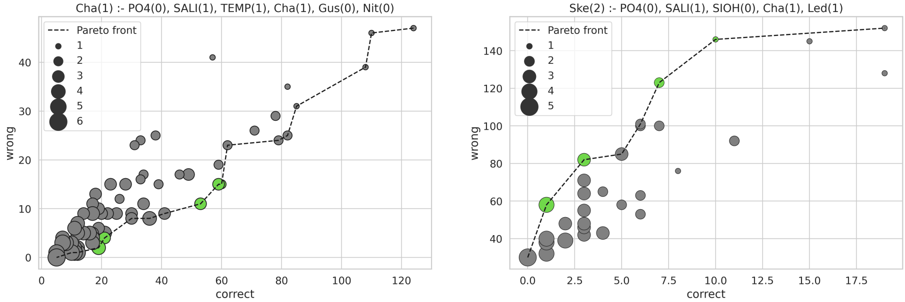
Unlikeliness rule: $Cha(0) \leftarrow Par(1), Dit(1), TEMP(0), Cha(1).$

Replace body with...	correct	wrong
$Par(1), Dit(1), TEMP(0), Cha(1)$	0	1
$Par(1), TEMP(0), Cha(1)$	3	12
$Dit(1), Cha(1)$	5	11
$Cha(1)$	11	66

Model Improvement

Pareto frontier

- For likeliness rules : maximize correct and minimize wrong weights
- For unlikeliness rules : maximize wrong and minimize correct weights



Accuracy improvement: 0.670 → **0.716**

Likeliness rules: 1683 → **1609**

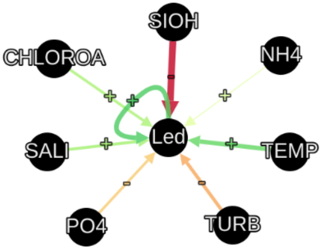
Unlikeliness rules: 1981 → **1405**

Global Influences

Process: Search and count patterns in rules that characterize an activation/inhibition
Hypotheses: Monotonous influences & same threshold for all variables
Result: Score $[-1; +1]$ between each pair of variables (no threshold)

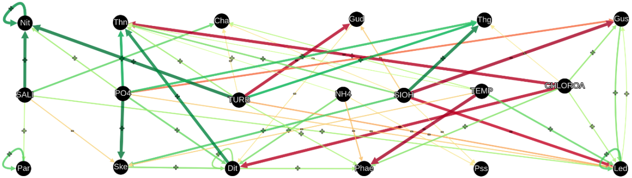
Example: **Influences** on phytoplankton specie `Led`:

Feature	positive	negative	global
PO4	$+0$	-58	-0.36
SALI	$+71$	-4	$+0.42$
CHLOROA	$+84$	-22	$+0.39$
SIOH	$+3$	-161	-0.98
NH4	$+25$	-5	$+0.12$
TEMP	$+106$	-5	$+0.63$
TURB	$+10$	-87	-0.48

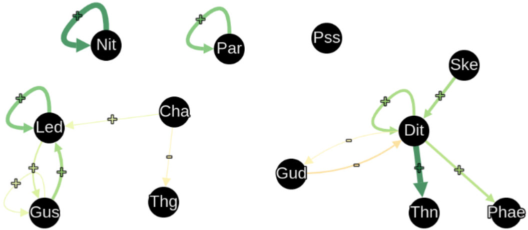


$$\text{global\_influence}(\texttt{PO4} \rightarrow \texttt{Led}) = \frac{+0+(-58)}{161} = -0.36$$

Result

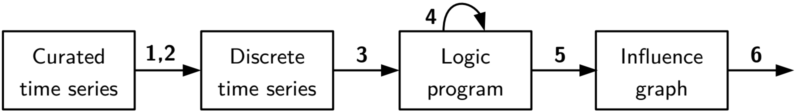


Global influence graph (biotic and abiotic interactions)



Biotic interactions (between phytoplankton only)

Conclusion



- **1** Data cleaning
- **2** Discretization ★
- **3 LFIT**

- **4** Rules improvement with Pareto front ★
- **5** Influence graph extraction ★
- **6** TODO: Validation & Exploitation ★

★ = Still work to do!

Special thanks:

- Sébastien Lefebvre — Professor of ecology, Université de Lille
  Laboratoire d'Océanologie et de Géosciences (Wimereux)
- Omar Iken — Master student in Data Science

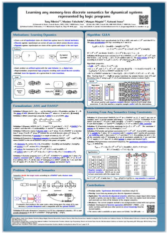
We are looking for PhD candidates!
Potential PhD thesis funding by Université de Lille
Please contact Cédric Lhoussaine or Maxime Folschette

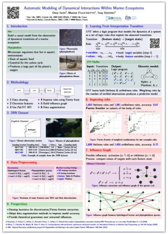
Outline

Conclusions

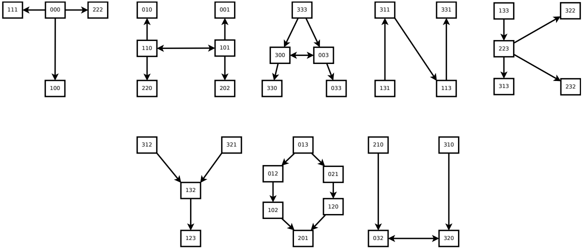
- **Previous works:** Synchronous deterministic transitions only.
- **Novelty:** Learn from any memory-less discrete dynamical semantics.
- **Application:** Explain observed dynamics as simple logic rules.
- **Collaboration:** Learn biotic influences from a marine ecosystem.
- **Outlook:** Heuristic to extract more knowledge from learned programs.
- Source code (Python) available as open source on Github.

Manuscript





Source Code


111
000
222
100
010
001
110
101
220
202
333
300
003
330
033
311
331
131
113
133
322
223
313
232
312
321
132
123
013
012
021
102
120
201
210
310
032
320