Searching for Phytoplankton Biotic Interactions in the SRN Dataset with LFIT

Maxime Folschette

Univ. Lille, CNRS, Centrale Lille, UMR 9189 CRIStAL, F-59000 Lille, France

2024-12-09

Journée ADÉM: « Apprentissage à partir de Données d'Écologie Marine »

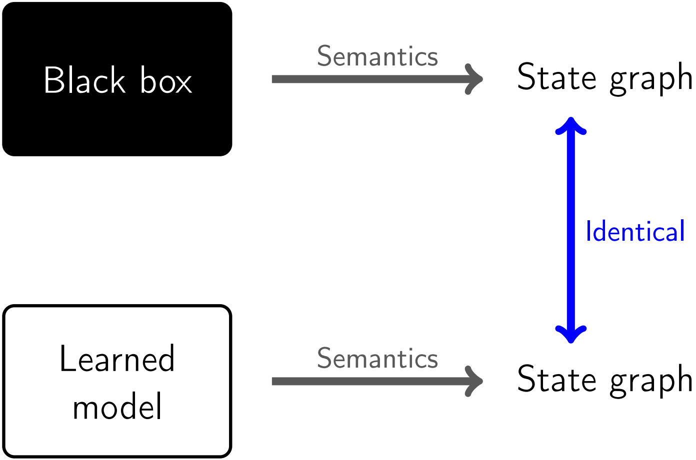
Outline

Introduction

Black box $\xrightarrow{\text{Semantics}}$ State graph

Learned model $\xrightarrow{\text{Semantics}}$ State graph

Introduction



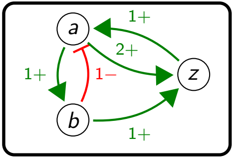
Introduction

Black box

Semantics

State graph

Identical



Semantics

State graph

Introduction

Black box

Semantics

State graph

Identical

$$
\begin{aligned}
&z_1 \leftarrow a_2.\\
&z_1 \leftarrow a_1 \wedge b_0.\\
&b_1 \leftarrow a_2.\\
&b_0 \leftarrow a_0.\\
&\dots
\end{aligned}
$$

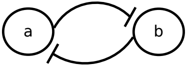
Semantics

State graph

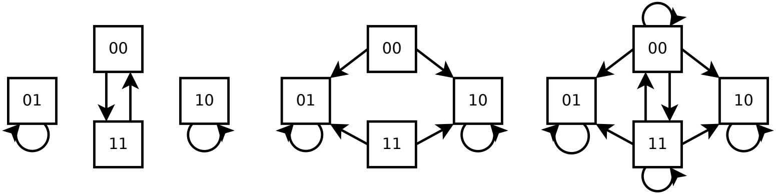
General Definitions

Dynamical Semantics

A Boolean network is a (syntactical) structure.
It must be interpreted with a semantics to run.



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

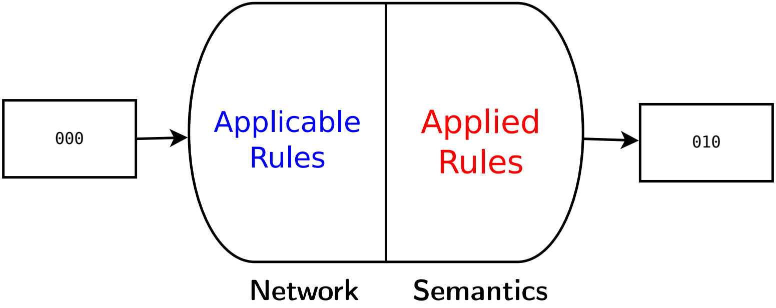


Synchronous | Asynchronous | General

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

Definition of Semantics

In a given state, among the possible changes permited by the network (structure), the semantics select which ones to apply and how to combine them.



Logic Rules

LFIT learns a logic program, which is a set of logic rules.
It is an alternative representation of biological networks.

$$a_1 \leftarrow a_0, b_0, c_2.$$

The network states that if a and b are at level 0 and c is at level 2, then a can change its value to 1.

$$a_1 \leftarrow c_2.$$

Whenever c is at level 2, a can change its value to 1.

$$a_1 \leftarrow .$$

a can change its value to 1 anytime.

Logic Rules

LFIT learns a logic program, which is a set of logic rules.
It is an alternative representation of biological networks.

$$a_1 \leftarrow a_0, b_0, c_2.$$

The network states that if a and b are at level 0 and c is at level 2, then a **can** change its value to 1.

$$a_1 \leftarrow c_2.$$

Whenever c is at level 2, a **can** change its value to 1.

$$a_1 \leftarrow .$$

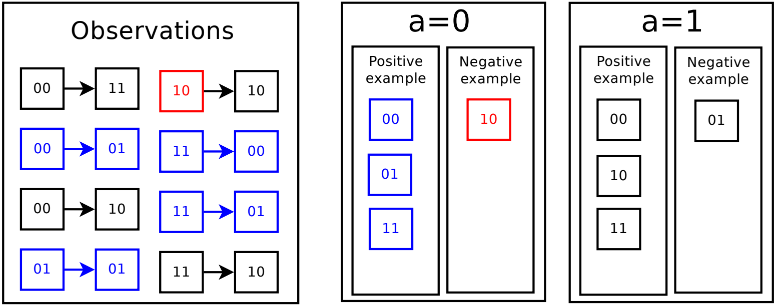
a **can** change its value to 1 anytime.

When **will** a take value 1? This depends on the semantics

Learning From Interpretation Transition (LFIT)

Learning Algorithm Intuition: Classification Problem

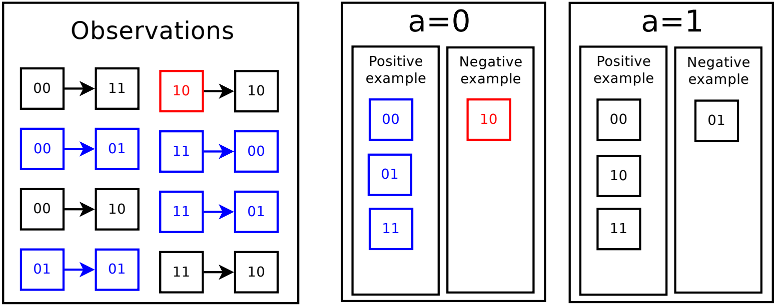
Learn applicable rules: conditions so that a variable **can** take a certain value in next state.



Equivalent to a **classification problem**: What is a typical state where a can take value 0 in the next state ? Here: when a_0 or b_1 is present.

Learning Algorithm Intuition: Classification Problem

Learn applicable rules: conditions so that a variable **can** take a certain value in next state.



Equivalent to a **classification problem**: What is a typical state where a can take value 0 in the next state ? Here: when a_0 or b_1 is present.

$$a_0 \leftarrow a_0. \qquad a_0 \leftarrow b_1.$$

Presentation of GULA

GULA = General Usage LFIT Algorithm

Input: a set of transitions ($s_1 \rightarrow s_2$)

Output: a logic program that respects:

- **Consistency**: the program allows no negative examples
- **Realization**: the program covers all positive examples
- **Completeness**: the program covers all the state space
- **Minimality** of the rules (most general conditions)

Method: start from most general rules and **specialize** iteratively.

Minimal refinements

Suppose: $\text{dom}(a) = \text{dom}(b) = \{0, 1\}$ and $\text{dom}(c) = \{0, 1, 2\}$
and the current program contains the following rules regarding a_1:

$$a_1 \leftarrow c_2. \qquad\qquad a_1 \leftarrow b_1.$$

From state $\langle a_1, b_0, c_2 \rangle$, a_1 is never observed in the next states.

Minimal refinement to make the rules inapplicable in this state:

Minimal refinements

Suppose: $\mathrm{dom}(a) = \mathrm{dom}(b) = \{0, 1\}$ and $\mathrm{dom}(c) = \{0, 1, 2\}$
and the current program contains the following rules regarding a_1:

$$a_1 \leftarrow c_2. \qquad\qquad a_1 \leftarrow b_1.$$

From state $\langle a_1, b_0, c_2 \rangle$, a_1 is never observed in the next states.

Minimal refinement to make the rules inapplicable in this state:

$$a_1 \leftarrow a_0, c_2.$$
$$a_1 \leftarrow b_1, c_2.$$
$$a_1 \leftarrow c_2, c_0.$$
$$a_1 \leftarrow c_2, c_1.$$

$$a_1 \leftarrow b_1.$$
(No change)

Minimal refinements

Suppose: $\text{dom}(a) = \text{dom}(b) = \{0, 1\}$ and $\text{dom}(c) = \{0, 1, 2\}$
and the current program contains the following rules regarding a_1:

$$a_1 \leftarrow c_2. \qquad\qquad a_1 \leftarrow b_1.$$

From state $\langle a_1, b_0, c_2 \rangle$, a_1 is never observed in the next states.

Minimal refinement to make the rules inapplicable in this state:

$$a_1 \leftarrow a_0, c_2. \qquad\qquad a_1 \leftarrow b_1.$$
$$a_1 \leftarrow b_1, c_2.$$
$$a_1 \leftarrow c_2, c_0.$$
$$a_1 \leftarrow c_2, c_1.$$

Minimal refinements

Suppose: $\mathrm{dom}(a) = \mathrm{dom}(b) = \{0, 1\}$ and $\mathrm{dom}(c) = \{0, 1, 2\}$
and the current program contains the following rules regarding a_1:

$a_1 \leftarrow c_2.$ $\quad$ $a_1 \leftarrow b_1.$

From state $\langle a_1, b_0, c_2 \rangle$, a_1 is never observed in the next states.

Minimal refinement to make the rules inapplicable in this state:

$a_1 \leftarrow a_0, c_2.$
$a_1 \leftarrow b_1, c_2.$

$a_1 \leftarrow b_1.$
(More general)

Minimal refinements

Suppose: $\mathrm{dom}(a) = \mathrm{dom}(b) = \{0, 1\}$ and $\mathrm{dom}(c) = \{0, 1, 2\}$
and the current program contains the following rules regarding a_1:

$$a_1 \leftarrow c_2. \qquad\qquad a_1 \leftarrow b_1.$$

From state $\langle a_1, b_0, c_2 \rangle$, a_1 is never observed in the next states.

Minimal refinement to make the rules inapplicable in this state:

$$a_1 \leftarrow a_0, c_2. \qquad\qquad a_1 \leftarrow b_1.$$

Results

Tony Ribeiro, Maxime Folschette, Morgan Magnin and Katsumi Inoue. **Learning any memory-less discrete semantics for dynamical systems represented by logic programs**. *Machine Learning* 111, Springer. November 2021. `https://doi.org/10.1007/s10994-021-06105-4`

- **Allows to learn the network** (structure of the model)
- **Independent of the semantics**
  (characterization of applicable memoryless semantics)

Nice in theory, but in practice?

- **Exponential complexity** $\rightarrow$ How to handle big datasets?
  (many transitions, many variables)
- **Exact learning** $\rightarrow$ How to handle noise?

Two Heuristic on LFIT

Weighted Likeliness/Unlikeliness Rules

- Use the algorithm twice to learn two logic programs:
  - likeliness rules: what is possible
  - unlikeliness rules: what is impossible
- Weight each rule by the number of observations it matches

Statistical overlay $\Rightarrow$ usable on **noisy datasets**

Likeliness rules	**Unlikeliness rules**
$(3, a_0 \leftarrow b_1)$	$(30, a_0 \leftarrow c_1)$
$(15, a_1 \leftarrow b_0)$	$(5, a_1 \leftarrow c_0)$
$\vdots$	$\vdots$

Using Weighted Likeliness/Unlikeliness Rules

Explainable predictions:

- Compare weights of applicable likeliness/unlikeliness rules
- Ratio of highest weights ⇒ **probability** P
- Rules with highest weights ⇒ **explanation** E

$$\text{predict} : (atom, state) \mapsto (P, E)$$

Likeliness rules

$(3, a_0 \leftarrow b_1)$

$(15, a_1 \leftarrow b_0)$

Unlikeliness rules

$(30, a_0 \leftarrow c_1)$

$(5, a_1 \leftarrow c_0)$

Using Weighted Likeliness/Unlikeliness Rules

Explainable predictions:

- Compare weights of applicable likeliness/unlikeliness rules
- Ratio of highest weights ⇒ **probability** P
- Rules with highest weights ⇒ **explanation** E

$$\text{predict} : (atom, state) \mapsto (P, E)$$

Likeliness rules

$(3, a_0 \leftarrow b_1)$

$(15, a_1 \leftarrow b_0)$

Unlikeliness rules

$(30, a_0 \leftarrow c_1)$

$(5, a_1 \leftarrow c_0)$

$\text{predict}(a_1, \langle a_1, b_1, c_0 \rangle) = (0.75, ((15, a_1 \leftarrow b_0), (5, a_1 \leftarrow c_0))) \Rightarrow$ Likely

Using Weighted Likeliness/Unlikeliness Rules

Explainable predictions:

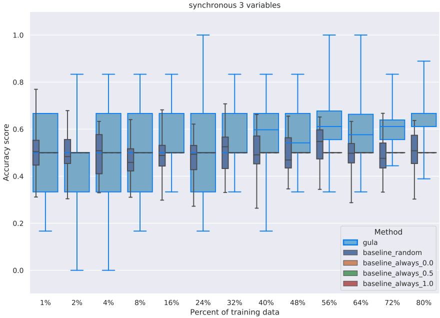
- Compare weights of applicable likeliness/unlikeliness rules
- Ratio of highest weights ⇒ **probability** P
- Rules with highest weights ⇒ **explanation** E

$$\text{predict} : (atom, state) \mapsto (P, E)$$

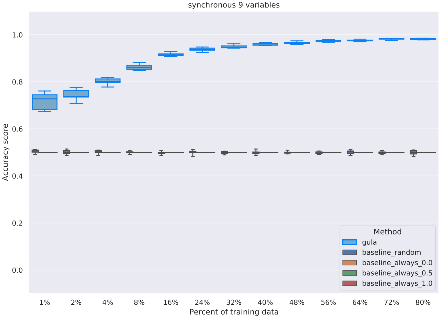
Likeliness rules	**Unlikeliness rules**
$(3, a_0 \leftarrow b_1)$	$(30, a_0 \leftarrow c_1)$
$(15, a_1 \leftarrow b_0)$	$(5, a_1 \leftarrow c_0)$

$\text{predict}(a_1, \langle a_1, b_1, c_0 \rangle) = (0.75, ((15, a_1 \leftarrow b_0), (5, a_1 \leftarrow c_0))) \Rightarrow$ Likely
$\text{predict}(a_0, \langle a_1, b_1, c_0 \rangle) = (0.09, ((3, a_0 \leftarrow b_1), (30, a_0 \leftarrow c_1))) \Rightarrow$ Unlikely

Prediction power



3 variables



9 variables

Training data = X% of transitions

Tested against unseen states (not in the training data)

PRIDE: Polynomial Alternative to GULA

GULA: **Exponential complexity** in the number of variables

PRIDE: Greedy version of **GULA** that only keeps the first compatible minimal refinement ⇒ subset of rules

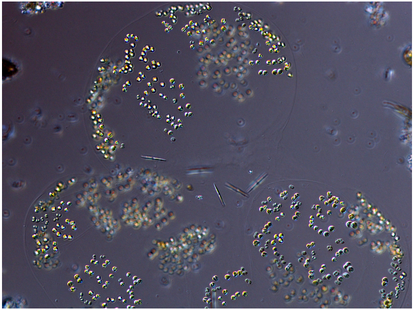
- **Consistency**: the program allows no negative examples
- **Realization**: the program covers all positive examples
- ~~**Completeness**: the program covers all the state space~~
- **Minimality** of the rules (most general conditions)

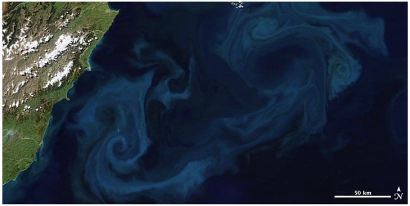
...And the results depends on the ordering of variables

Polynomial complexity ⇒ usable on **large datasets**

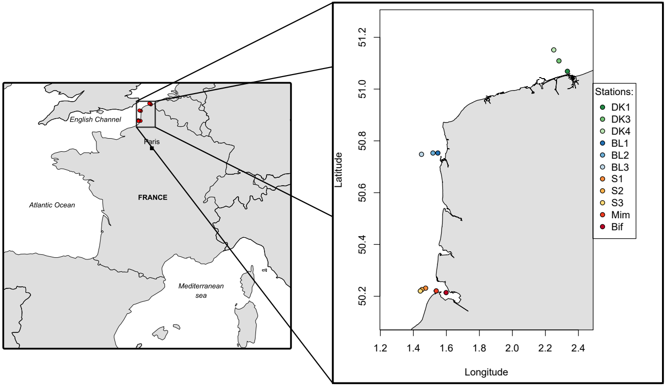
Application: Dynamics of Marine Phytoplankton

Phytoplankton Blooms





SRN Dataset



Karasiewicz Stephane, Lefebvre Alain (2022). Environmental Impact on Harmful Species Pseudo-nitzschia spp. and Phaeocystis globosa Phenology and Niche. *Journal Of Marine Science And Engineering*. 10 (2). 174 (31p.). https://doi.org/10.3390/jmse10020174

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

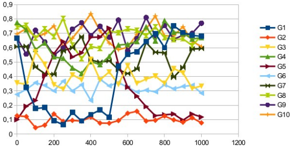
Contents of the SRN Dataset

1992–2023
$\simeq$ 3000 data points

Sampling location	Sampling date	Taxon/Parameter	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)
⋮	⋮	⋮	⋮	⋮

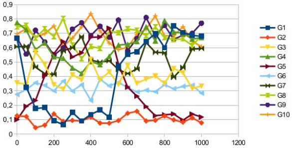
Environmental variables (7) Phytoplankton species (12)

Learning from Noisy Time Series



Time series

Learning from Noisy Time Series

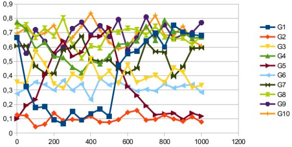


Time series

$$
\begin{aligned}
Led_{\mathbf{1}} &\leftarrow SIOH_{\mathbf{0}} \wedge TEMP_{\mathbf{1}} \wedge TURB_{\mathbf{0}}.\\
Led_{\mathbf{1}} &\leftarrow SIOH_{\mathbf{0}} \wedge SALI_{\mathbf{1}}\\
Led_{\mathbf{0}} &\leftarrow PO4_{\mathbf{0}} \wedge TEMP_{\mathbf{1}} \wedge TURB_{\mathbf{1}}.\\
Led_{\mathbf{0}} &\leftarrow CHLOROA_{\mathbf{0}} \wedge PO4_{\mathbf{1}}.\\
Led_{\mathbf{1}} &\leftarrow TEMP_{\mathbf{0}} \wedge SALI_{\mathbf{0}}\\
Led_{\mathbf{1}} &\leftarrow PO4_{\mathbf{1}} \wedge TEMP_{\mathbf{0}} \wedge TURB_{\mathbf{1}}.
\end{aligned}
$$

Logic program

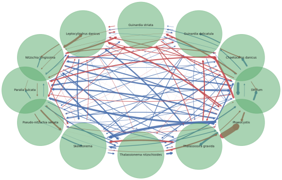
Learning from Noisy Time Series



Time series

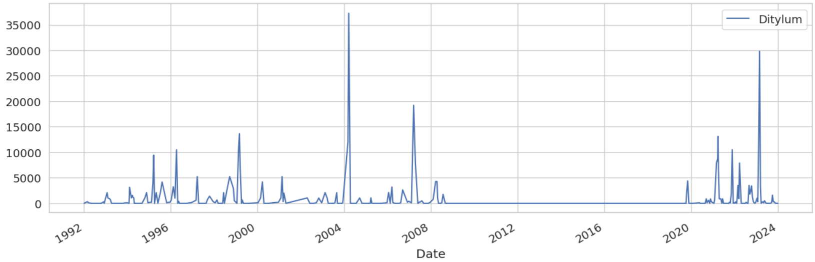
$$
\begin{aligned}
&Led_1 \leftarrow SIOH_0 \wedge TEMP_1 \wedge TURB_0 .\\
&Led_1 \leftarrow SIOH_0 \wedge SALI_1\\
&Led_0 \leftarrow PO4_0 \wedge TEMP_1 \wedge TURB_1 .\\
&Led_0 \leftarrow CHLOROA_0 \wedge PO4_1 .\\
&Led_1 \leftarrow TEMP_0 \wedge SALI_0\\
&Led_1 \leftarrow PO4_1 \wedge TEMP_0 \wedge TURB_1 .
\end{aligned}
$$

Logic program

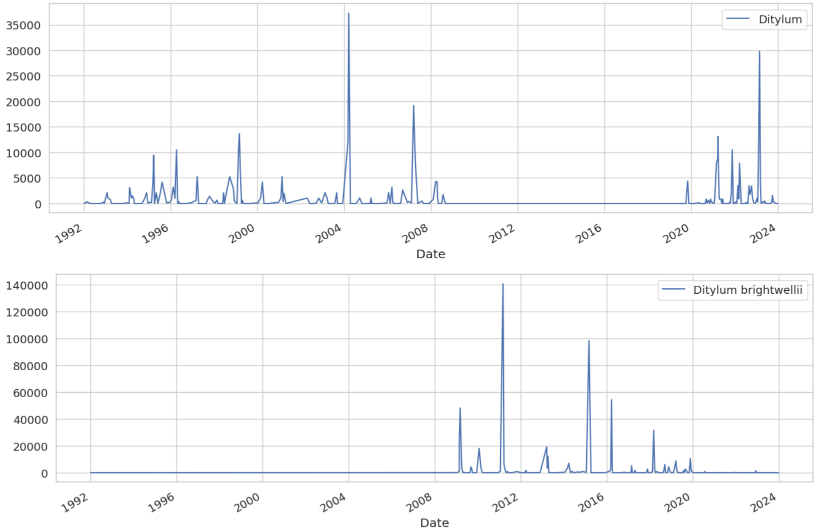


Influence graph

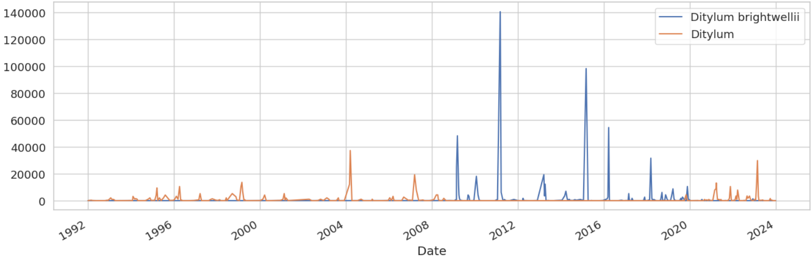
Step 1: Pre-Processing



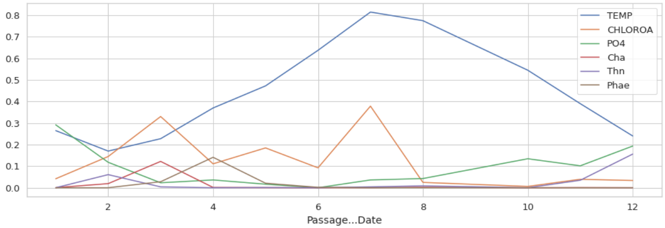
Step 1: Pre-Processing

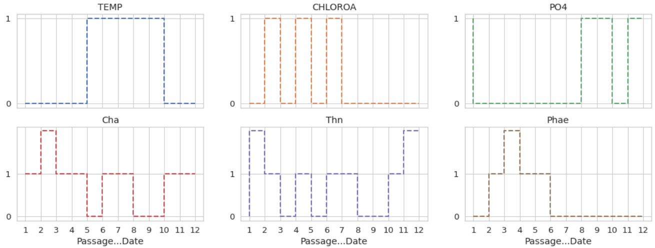


Step 1: Pre-Processing

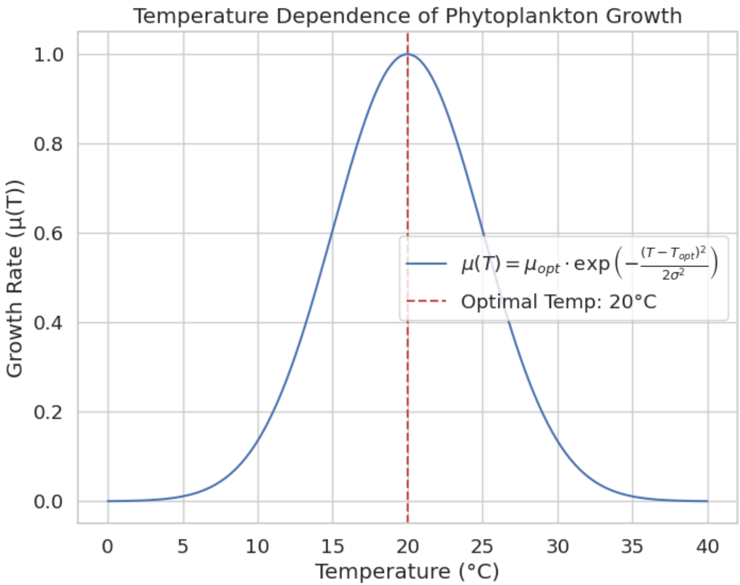


Step 2: Discretization

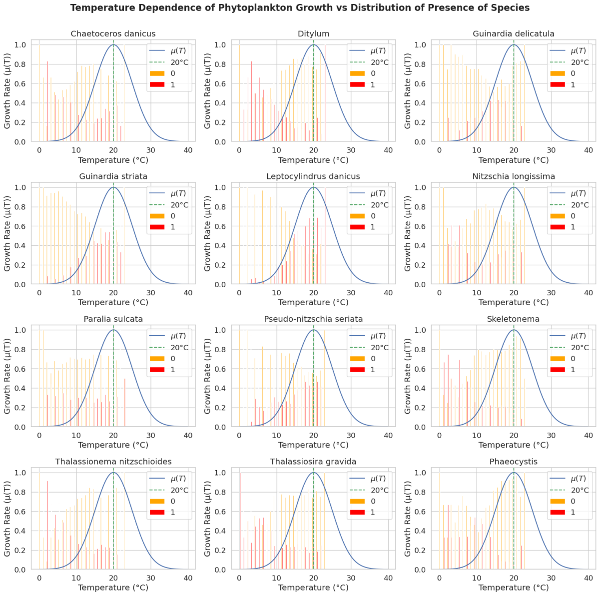




Step 2: Discretization



Step 2: Discretization



Step 3: Applying LFIT

$$\vdots$$
$$Led_1 \leftarrow SIOH_0 \wedge TEMP_1 \wedge TURB_0.$$
$$Led_1 \leftarrow SIOH_0 \wedge SALI_1$$
$$Led_0 \leftarrow PO4_0 \wedge TEMP_1 \wedge TURB_1.$$
$$Led_0 \leftarrow CHLOROA_0 \wedge PO4_1.$$
$$Led_1 \leftarrow TEMP_0 \wedge SALI_0$$
$$Led_1 \leftarrow PO4_1 \wedge TEMP_0 \wedge TURB_1.$$
$$\vdots$$

- Run time = 2.35s (**PRIDE**)
- $\simeq$ 3500 rules
- Model accuracy: depends on the discretization choices! between **67%** and **77%**

Step 4: Compute Global Influences

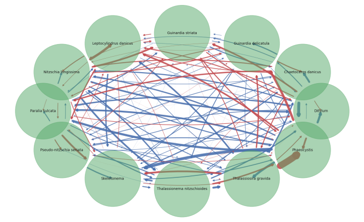
Process: Search and count patterns in rules that characterize an activation/inhibition

Hypotheses: Monotonous influences & same threshold for all variables

Result: Score $[-1; +1]$ between each pair of variables (no threshold)

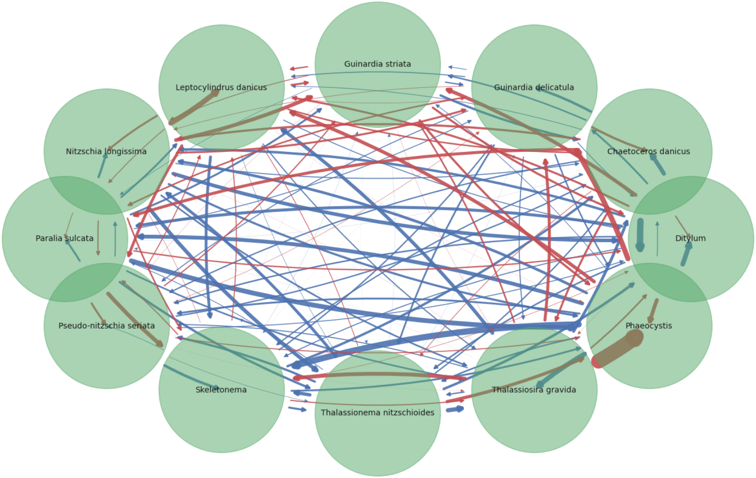
Influences on phytoplankton specie `Led`:

Variable	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



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

Influence Graph



Conclusion

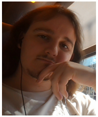
Conclusion

- **Learn** the network with LFIT (theory)
- **Heuristics** to tackle real data (practice)
- **Application** to phytoplankton

Outlooks:

- Quatify how many rules are “missed” by PRIDE
- Integrate biological knowledge to improve learning
- Improve the Biological network inference
- ...

Thanks



Tony
RIBEIRO

Omar
IKNE

Morgan
MAGNIN

Katsumi
INOUE

Cédric
LHOUSSAINE

Sébastien
LEFEBVRE

Madeleine
EYRAUD

Bibliography

- About GULA: Tony Ribeiro, Maxime Folschette, Morgan Magnin and Katsumi Inoue. **Learning any memory-less discrete semantics for dynamical systems represented by logic programs**. *Machine Learning* 111, Springer. November 2021. https://doi.org/10.1007/s10994-021-06105-4
- pyLFIT Python library: https://github.com/Tony-sama/pylfit
- About PRIDE: Tony Ribeiro, Maxime Folschette, Morgan Magnin and Katsumi Inoue. **Polynomial Algorithm For Learning From Interpretation Transition**. Poster at the *1st International Joint Conference on Learning & Reasoning*. October 2021, Online. https://hal.science/hal-03347026v1
- About the application: Omar Iken, Maxime Folschette and Tony Ribeiro. **Automatic Modeling of Dynamical Interactions Within Marine Ecosystems**. Poster in the *1st International Joint Conference on Learning & Reasoning*. October 2021, Online. https://hal.science/hal-03347033v1