Conservation Analysis and Discrete Probabilistic Approximations for Parameter Estimation of Biochemical Networks

QEST + FORMATS 2025

Olivier Bouët-Willaumez[1], Adrien Le Coënt[1], Benoît Barbot[1], Nihal Pekergin[1]

[1] LACL, Université Paris-Est Créteil, France

Modelisation and ODE formalism

To analyze the dynamics of biochemical networks, a common modeling approach is to use Ordinary Differential Equations (ODE) systems.

Example: Michaelis-Menten / Enzyme-catalyzed kinetics.

- $S + E \underset{k_2}{\overset{k_1}{\rightleftarrows}} ES \xrightarrow{k_3} P + E$
- $$\begin{cases} d[S]/dt = -k_1[S][E] + k_2[ES] \\ d[E]/dt = -k_1[S][E] + (k_2 + k_3)[ES] \\ d[ES]/dt = k_1[S][E] - (k_2 + k_3)[ES] \\ d[P]/dt = k_3[ES] \end{cases}$$

Modelisation and ODE formalism

To analyze the dynamics of biochemical networks, a common modeling approach is to use Ordinary Differential Equations (ODE) systems.

Example: Michaelis-Menten / Enzyme-catalyzed kinetics.

- $\mathrm{S} + \mathrm{E} \underset{k_2}{\overset{k_1}{\rightleftharpoons}} \mathrm{ES} \xrightarrow{k_3} \mathrm{P} + \mathrm{E}$
- $$\begin{cases} d[S]/dt = -k_1[S][E] + k_2[ES] \\ d[E]/dt = -k_1[S][E] + (k_2 + k_3)[ES] \\ d[ES]/dt = k_1[S][E] - (k_2 + k_3)[ES] \\ d[P]/dt = k_3[ES] \end{cases}$$

Modelisation and ODE formalism

To analyze the dynamics of biochemical networks, a common modeling approach is to use Ordinary Differential Equations (ODE) systems.

Example: Michaelis-Menten / Enzyme-catalyzed kinetics.

- $\mathrm{S} + \mathrm{E} \underset{k_2}{\overset{k_1}{\rightleftharpoons}} \mathrm{ES} \xrightarrow{k_3} \mathrm{P} + \mathrm{E}$
- $$\begin{cases} \dot{x_1} = -k_1 x_1 x_2 + k_2 x_3 \\ \dot{x_2} = -k_1 x_1 x_2 + (k_2 + k_3) x_3 \\ \dot{x_3} = k_1 x_1 x_2 - (k_2 + k_3) x_3 \\ \dot{x_4} = k_3 x_3 \end{cases}$$

Parameter Estimation and classical approaches

In such systems, estimating parameters from experimental data is a central challenge (system biology).

ODE-based approaches:

- Repeatedly simulating ODEs and minimizing a cost function.
- Markov Chain Monte Carlo (MCMC), to tackle uncertainty and improve inference robustness.

$\Longrightarrow$ High computational costs when the parameter space is large or when model stiffness is significant.

Parameter Estimation and classical approaches

In such systems, estimating parameters from experimental data is a central challenge (system biology).

ODE-based approaches:

- Repeatedly simulating ODEs and minimizing a cost function.
- Markov Chain Monte Carlo (MCMC), to tackle uncertainty and improve inference robustness.

⟹ High computational costs when the parameter space is large or when model stiffness is significant.

Parameter Estimation and classical approaches

In such systems, estimating parameters from experimental data is a central challenge (system biology).

ODE-based approaches:

- Repeatedly simulating ODEs and minimizing a cost function.
- Markov Chain Monte Carlo (MCMC), to tackle uncertainty and improve inference robustness.

$\Longrightarrow$ High computational costs when the parameter space is large or when model stiffness is significant.

Probabilistic Approximations of ODEs

Transform ODE models into discrete probabilistic models → Dynamic Bayesian Networks (DBNs):

- Faster cost function evaluation and avoid the need for repeated ODE simulations.
- Outperform classical ODE-based optimization for moderately sized system[1].

$\Longrightarrow$ Scalability limitations for larger models:

- Exponential growth of the state space.
- Complexity of computing and storing transition probabilities.

[1] Liu, B. et al. Probabilistic Approximations of ODEs-Based Bio-Pathway Dynamics (2011).

Probabilistic Approximations of ODEs

Transform ODE models into discrete probabilistic models → Dynamic Bayesian Networks (DBNs):

- Faster cost function evaluation and avoid the need for repeated ODE simulations.
- Outperform classical ODE-based optimization for moderately sized system[1].

$\Longrightarrow$ Scalability limitations for larger models:

- Exponential growth of the state space.
- Complexity of computing and storing transition probabilities.

[1] Liu, B. et al. Probabilistic Approximations of ODEs-Based Bio-Pathway Dynamics (2011).

Model reduction

Simplify the system while preserving key dynamical properties.

These approaches can generally be divided into two categories:

- **Structural simplification** methods → Conservation Analysis[2].
- **Component reduction** methods → Classical lumping techniques[3], Equivalence-based reductions using backward bisimulation principles[4].

[2] Vallabhajosyula, R.R. et al.. Conservation Analysis of Large Biochemical Networks (2006).

[3] Dokoumetzidis, A. et al.. Proper lumping in systems biology models (2009).

[4] Cardelli, L. et al.. Symbolic computation of differential equivalences (2016).

Model reduction

Simplify the system while preserving key dynamical properties.

These approaches can generally be divided into two categories:

- **Structural simplification** methods → Conservation Analysis[2].
- **Component reduction** methods → Classical lumping techniques[3], Equivalence-based reductions using backward bisimulation principles[4].

[2] Vallabhajosyula, R.R. et al.. Conservation Analysis of Large Biochemical Networks (2006).

[3] Dokoumetzidis, A. et al.. Proper lumping in systems biology models (2009).

[4] Cardelli, L. et al.. Symbolic computation of differential equivalences (2016).

Model reduction

Simplify the system while preserving key dynamical properties.

These approaches can generally be divided into two categories:

- **Structural simplification** methods → Conservation Analysis[2].
- **Component reduction** methods → Classical lumping techniques[3], Equivalence-based reductions using backward bisimulation principles[4].

[2] Vallabhajosyula, R.R. et al.. Conservation Analysis of Large Biochemical Networks (2006).

[3] Dokoumetzidis, A. et al.. Proper lumping in systems biology models (2009).

[4] Cardelli, L. et al.. Symbolic computation of differential equivalences (2016).

Model reduction

Simplify the system while preserving key dynamical properties.

These approaches can generally be divided into two categories:

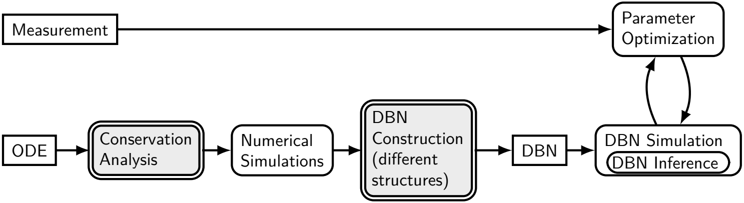
- **Structural simplification** methods → Conservation Analysis[2].
- **Component reduction** methods → Classical lumping techniques[3], Equivalence-based reductions using backward bisimulation principles[4].

[2] Vallabhajosyula, R.R. et al.. Conservation Analysis of Large Biochemical Networks (2006).

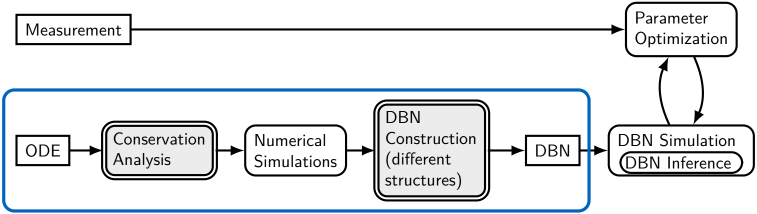
[3] Dokoumetzidis, A. et al.. Proper lumping in systems biology models (2009).

[4] Cardelli, L. et al.. Symbolic computation of differential equivalences (2016).

Parameter Estimation pipeline



Parameter Estimation pipeline



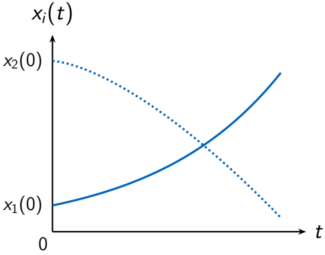
Performed once and for all

Probabilistic Approximations

Main Idea

Objective: learn the parameter $\boldsymbol{k}$ of the ODE system:

$$\dot{x_1} = \boldsymbol{k} x_1$$
$$\dot{x_2} = -0.9 \boldsymbol{k} x_1$$

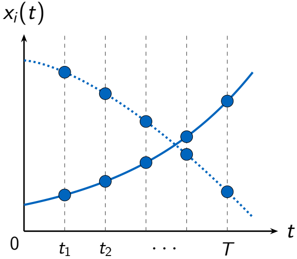


- ODE solution trajectory for an unknown parameter $\boldsymbol{k}$ value.

Main Idea

Objective: learn the parameter $\boldsymbol{k}$ of the ODE system:

$$\dot{x_1} = \boldsymbol{k} x_1$$
$$\dot{x_2} = -0.9 \boldsymbol{k} x_1$$

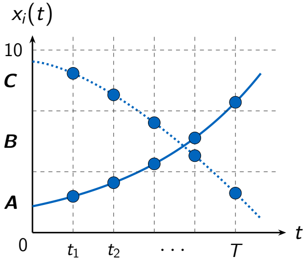


- Observe only at discrete time points $0, t_1, t_2, \ldots, T$.

Main Idea

Objective: learn the parameter $\boldsymbol{k}$ of the ODE system:

$$\dot{x}_1 = \boldsymbol{k} x_1$$
$$\dot{x}_2 = -0.9 \boldsymbol{k} x_1$$

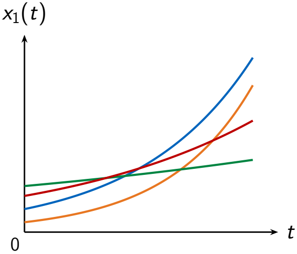


- Each species has bounds: $x_1, x_2 \in [0, 10]^2$.
- Discretize the trajectories with a given number of subintervals.
- Discretize the parameters.

Main Idea

Objective: learn the parameter $\boldsymbol{k}$ of the ODE system:

$$\dot{x_1} = \boldsymbol{k} x_1$$
$$\dot{x_2} = -0.9 \boldsymbol{k} x_1$$



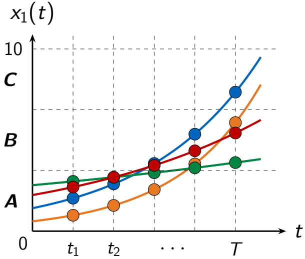
- Run a number of simulations, with randomized initial conditions and parameter values.

Main Idea

Objective: learn the parameter $\boldsymbol{k}$ of the ODE system:

$$\dot{x_1} = \boldsymbol{k} x_1$$

$$\dot{x_2} = -0.9 \boldsymbol{k} x_1$$

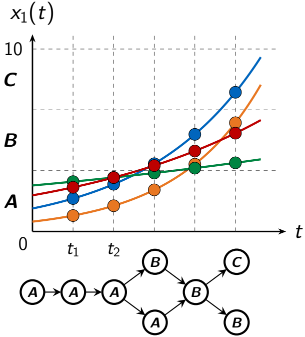


- Discretize the trajectories.

Main Idea

Objective: learn the parameter $\boldsymbol{k}$ of the ODE system:

$$\dot{x}_1 = \boldsymbol{k} x_1$$
$$\dot{x}_2 = -0.9\boldsymbol{k} x_1$$

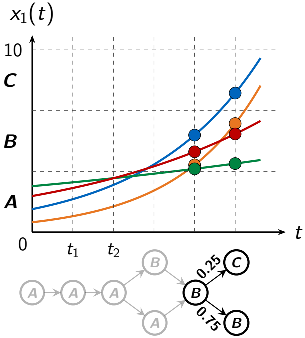


- Under the Markov property and time-homogeneity assumptions, the process forms a discrete-time Markov chain (DTMC).

Main Idea

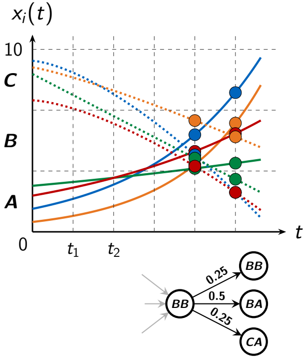
Objective: learn the parameter $\boldsymbol{k}$ of the ODE system:

$$\dot{x_1} = \boldsymbol{k} x_1$$
$$\dot{x_2} = -0.9\boldsymbol{k} x_1$$



- Compute the transition probabilities from the simulations.

Main Idea



- $\mathbb{P}((s, t_i) \rightarrow (s', t_{i+1}))$ is the fraction of trajectories in s at t_i that land in s' at t_{i+1}.

From Markov Chains to Bayesian Networks

Markov chain: joint distribution over trajectory:

$$p(\boldsymbol{X}^{0:T}) = p(\boldsymbol{X}^0)\prod_{t=1}^{T} p(\boldsymbol{X}^t \mid \boldsymbol{X}^{t-1})$$

Problem: The transition table $p(\boldsymbol{X}^t \mid \boldsymbol{X}^{t-1})$ grows exponentially with the number of variables and discrete states.

Bayesian Networks: exploit conditional independencies:

$$p(\boldsymbol{X}^t) = \prod_{i} p(X_i^t \mid \text{Parents}(X_i^t))$$

$\Longrightarrow$ Each X_i^t depends only on a subset of variables (its parents), enabling a compact representation.

From Markov Chains to Bayesian Networks

Markov chain: joint distribution over trajectory:

$$p(\boldsymbol{X}^{0:T}) = p(\boldsymbol{X}^0) \prod_{t=1}^{T} p(\boldsymbol{X}^t \mid \boldsymbol{X}^{t-1})$$

Problem: The transition table $p(\boldsymbol{X}^t \mid \boldsymbol{X}^{t-1})$ grows exponentially with the number of variables and discrete states.

Bayesian Networks: exploit conditional independencies:

$$p(\boldsymbol{X}^t) = \prod_{i} p(X_i^t \mid \text{Parents}(X_i^t))$$

$\Longrightarrow$ Each X_i^t depends only on a subset of variables (its parents), enabling a compact representation.

From Markov Chains to Bayesian Networks

Markov chain: joint distribution over trajectory:

$$p(\boldsymbol{X}^{0:T}) = p(\boldsymbol{X}^0) \prod_{t=1}^{T} p(\boldsymbol{X}^t \mid \boldsymbol{X}^{t-1})$$

Problem: The transition table $p(\boldsymbol{X}^t \mid \boldsymbol{X}^{t-1})$ grows exponentially with the number of variables and discrete states.

Bayesian Networks: exploit conditional independencies:

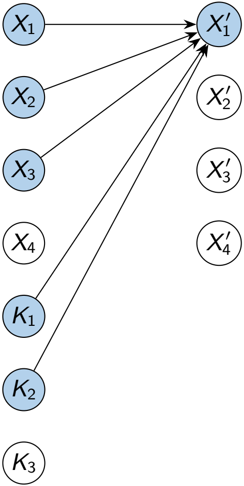
$$p(\boldsymbol{X}^t) = \prod_{i} p(X_i^t \mid \mathrm{Parents}(X_i^t))$$

$\Longrightarrow$ Each X_i^t depends only on a subset of variables (its parents), enabling a compact representation.

Bayesian Networks structure

$$\begin{cases} \dot{x_1} = -k_1 x_1 x_2 + k_2 x_3 \\ \\ \\ \end{cases}$$

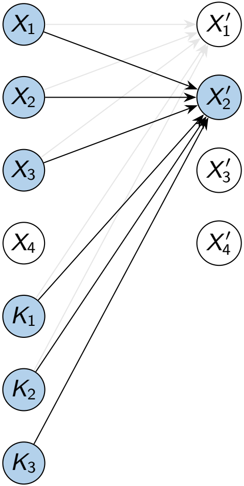
Parents$(X_1') = \{X_1, X_2, X_3, K_1, K_2\}$



Bayesian Networks structure

$$\begin{cases} \dot{x_1} = -k_1 x_1 x_2 + k_2 x_3 \\ \dot{x_2} = -k_1 x_1 x_2 + (k_2 + k_3) x_3 \\ \\ \end{cases}$$

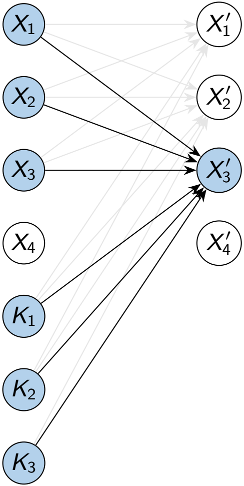
Parents$(X_2') = \{X_1, X_2, X_3, K_1, K_2, K_3\}$



Bayesian Networks structure

$$\begin{cases} \dot{x_1} = -k_1 x_1 x_2 + k_2 x_3 \\ \dot{x_2} = -k_1 x_1 x_2 + (k_2 + k_3) x_3 \\ \dot{x_3} = k_1 x_1 x_2 - (k_2 + k_3) x_3 \\ \end{cases}$$

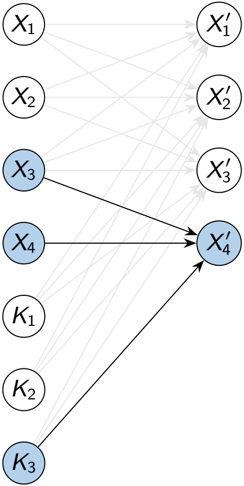
Parents$(X_3') = \{X_1, X_2, X_3, K_1, K_2, K_3\}$



Bayesian Networks structure

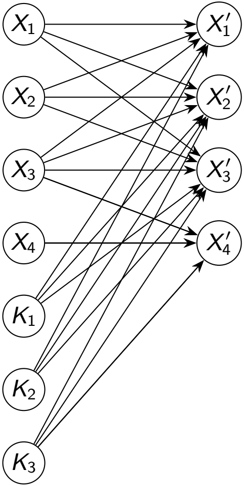
$$\begin{cases} \dot{x_1} = -k_1 x_1 x_2 + k_2 x_3 \\ \dot{x_2} = -k_1 x_1 x_2 + (k_2 + k_3) x_3 \\ \dot{x_3} = k_1 x_1 x_2 - (k_2 + k_3) x_3 \\ \dot{x_4} = k_3 x_3 \end{cases}$$

$\text{Parents}(X_4') = \{X_3, X_4, K_3\}$



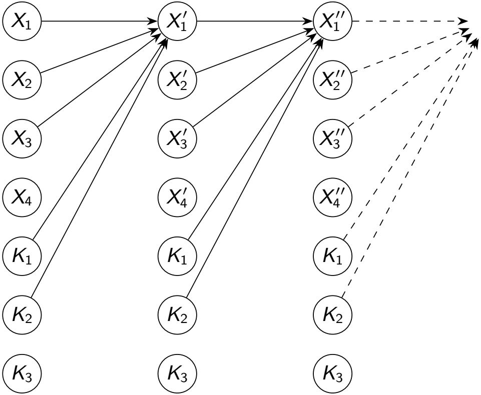
Bayesian Networks structure

$$
\begin{cases}
\dot{x_1} = -k_1 x_1 x_2 + k_2 x_3 \\
\dot{x_2} = -k_1 x_1 x_2 + (k_2 + k_3) x_3 \\
\dot{x_3} = k_1 x_1 x_2 - (k_2 + k_3) x_3 \\
\dot{x_4} = k_3 x_3
\end{cases}
$$

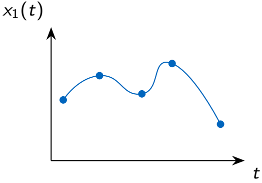


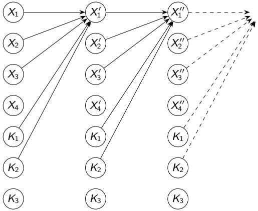
Dynamic Bayesian Networks

Extending Bayesian networks to sequences of variables over time.



DBN parameter estimation





$$\varepsilon(\boldsymbol{K}) = \sum_i \sum_t \left[\overbrace{x_i^t} - \left(\sum_l \overbrace{p\left(X_i^t = l \mid \boldsymbol{K}, \boldsymbol{X}^0\right)} \bar{x}_i^l \right) \right]^2$$

$$\tilde{\boldsymbol{K}}$$

Conservation Analysis

ODE matrix form

Recall:

$$\begin{cases} \dot{x_1} = -k_1 x_1 x_2 + k_2 x_3 \\ \dot{x_2} = -k_1 x_1 x_2 + (k_2 + k_3) x_3 \\ \dot{x_3} = k_1 x_1 x_2 - (k_2 + k_3) x_3 \\ \dot{x_4} = k_3 x_3 \end{cases}$$

The system is written:

$$\dot{x}_i(t) = f_i(\boldsymbol{x}(t), \boldsymbol{k}),\ \ i \in 1, \ldots, n$$

with $\boldsymbol{x}(t) = (x_1(t), \ldots, x_n(t))$, $\boldsymbol{k} = (k_1, \ldots, k_m)$, and the $f_i,\ \ i \in 1, \ldots, n$ are rational functions.

ODE matrix form

Recall:

$$\begin{cases} \dot{x}_1 = -k_1 x_1 x_2 + k_2 x_3 \\ \dot{x}_2 = -k_1 x_1 x_2 + (k_2 + k_3) x_3 \\ \dot{x}_3 = k_1 x_1 x_2 - (k_2 + k_3) x_3 \\ \dot{x}_4 = k_3 x_3 \end{cases}$$

The system is written:

$$\dot{x}_i(t) = f_i(\boldsymbol{x}(t), \boldsymbol{k}),\ \ i \in 1, \dots, n$$

with $\boldsymbol{x}(t) = (x_1(t), \dots, x_n(t))$, $\boldsymbol{k} = (k_1, \dots, k_m)$, and the $f_i,\ \ i \in 1, \dots, n$ are rational functions.

ODE matrix form

The ODE can be written in matrix form as follows:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{N} \cdot \boldsymbol{v}(\boldsymbol{x}(t), \boldsymbol{k})$$

where:

- $\boldsymbol{v}(\boldsymbol{x}(t), \boldsymbol{k}) = (v_1(\boldsymbol{x}(t), \boldsymbol{k}), \ldots, v_r(\boldsymbol{x}(t), \boldsymbol{k}))^T$ is the rate vector of the r reactions in the network.
- The Stoichiometric matrix $\boldsymbol{N}$ is $n \times r$ dimensional.

ODE matrix form

Example:

$$\begin{cases} \dot{x_1} = -k_1x_1x_2 + k_2x_3 \\ \dot{x_2} = -k_1x_1x_2 + (k_2+k_3)x_3 \\ \dot{x_3} = k_1x_1x_2 - (k_2+k_3)x_3 \\ \dot{x_4} = k_3x_3 \end{cases} \iff \underbrace{\begin{pmatrix} -1 & 1 & 0 \\ -1 & 1 & 1 \\ 1 & -1 & -1 \\ 0 & 0 & 1 \end{pmatrix}}_{\boldsymbol{N}} \cdot \underbrace{\begin{pmatrix} k_1x_1x_2 \\ k_2x_3 \\ k_3x_3 \end{pmatrix}}_{\boldsymbol{v}}$$

Conservation Analysis

Conservation analysis[5] separates species into independent ($\boldsymbol{x}_i$) and dependent ($\boldsymbol{x}_d$) sets by analyzing the linear dependence of the Stoichiometric matrix $\boldsymbol{N}$ rows.

In practice, we perform a full pivoting QR decomposition of $\boldsymbol{N}$ using Householder reflections to extract a link matrix $\boldsymbol{L}_0$.

This enhance numerical stability, even for systems with large condition number.

[5] Vallabhajosyula, R.R. et al.. Conservation Analysis of Large Biochemical Networks (2006).

Conservation Analysis

Conservation analysis[5] separates species into independent ($\boldsymbol{x}_i$) and dependent ($\boldsymbol{x}_d$) sets by analyzing the linear dependence of the Stoichiometric matrix $\boldsymbol{N}$ rows.

In practice, we perform a full pivoting QR decomposition of $\boldsymbol{N}$ using Householder reflections to extract a link matrix $\boldsymbol{L}_0$.

This enhance numerical stability, even for systems with large condition number.

[5] Vallabhajosyula, R.R. et al.. Conservation Analysis of Large Biochemical Networks (2006).

Conservation Analysis

By rearranging rows, $\boldsymbol{N}$ can be written:

$$\boldsymbol{N} = \begin{bmatrix} \boldsymbol{N}_I \\ \boldsymbol{N}_D \end{bmatrix}, \quad \text{with} \quad \boldsymbol{N}_D = \boldsymbol{L}_0 \cdot \boldsymbol{N}_I$$

Then, we can deduce the independent and dependent species subsystems such as:

$$\dot{\boldsymbol{x}}_i(t) = \boldsymbol{N}_I \cdot \boldsymbol{v}(t)$$

and

$$\dot{\boldsymbol{x}}_d(t) = \boldsymbol{L}_\mathbf{0} \cdot \dot{\boldsymbol{x}}_i(t)$$

Conservation Analysis

By rearranging rows, $\boldsymbol{N}$ can be written:

$$\boldsymbol{N} = \begin{bmatrix} \boldsymbol{N}_I \\ \boldsymbol{N}_D \end{bmatrix}, \quad \text{with} \quad \boldsymbol{N}_D = \boldsymbol{L}_0 \cdot \boldsymbol{N}_I$$

Then, we can deduce the independent and dependent species subsystems such as:

$$\dot{\boldsymbol{x}}_i(t) = \boldsymbol{N}_I \cdot \boldsymbol{v}(t)$$

and

$$\dot{\boldsymbol{x}}_d(t) = \boldsymbol{L_0} \cdot \dot{\boldsymbol{x}}_i(t)$$

Conservation Analysis

By rearranging rows, $\boldsymbol{N}$ can be written:

$$\boldsymbol{N} = \begin{bmatrix} \boldsymbol{N}_I \\ \boldsymbol{N}_D \end{bmatrix}, \quad \text{with} \quad \boldsymbol{N}_D = \boldsymbol{L}_0 \cdot \boldsymbol{N}_I$$

Then, we can deduce the independent and dependent species subsystems such as:

$$\dot{\boldsymbol{x}}_i(t) = \boldsymbol{N}_I \cdot \boldsymbol{v}(t)$$

and

$$\boldsymbol{x}_d(t) = \boldsymbol{x}_d(0) + \boldsymbol{L_0} \cdot [\boldsymbol{x}_i(t) - \boldsymbol{x}_i(0)]$$

Example

The original system:

$$\begin{cases} \dot{x}_1 = -k_1 x_1 x_2 + k_2 x_3 \\ \dot{x}_2 = -k_1 x_1 x_2 + (k_2 + k_3) x_3 \\ \dot{x}_3 = k_1 x_1 x_2 - (k_2 + k_3) x_3 \\ \dot{x}_4 = k_3 x_3 \end{cases}$$

can be reduced to a subsystem of only **2** independent species:

$$\begin{cases} \dot{x}_1 = -k_1 x_1 x_2 + k_2 x_3 \\ \dot{x}_2 = -k_1 x_1 x_2 + (k_2 + k_3) x_3 \end{cases}$$

with $x_3(t) = -x_2(t) + x_3(0) + x_2(0)$.

Example

The original system:

$$\begin{cases} \dot{x_1} = -k_1 x_1 x_2 + k_2 x_3 \\ \dot{x_2} = -k_1 x_1 x_2 + (k_2 + k_3) x_3 \\ \dot{x_3} = k_1 x_1 x_2 - (k_2 + k_3) x_3 \\ \dot{x_4} = k_3 x_3 \end{cases}$$

can be reduced to a subsystem of only **2** independent species:

$$\begin{cases} \dot{x_1} = -k_1 x_1 x_2 + k_2 x_3 \\ \dot{x_2} = -k_1 x_1 x_2 + (k_2 + k_3) x_3 \end{cases}$$

with $x_3(t) = -x_2(t) + x_3(0) + x_2(0)$.

Reduced DBN Structures

Different reduced DBN structures

How to represent the following reduced subsystem with a BN ?

$$\begin{cases} \dot{x_1} = -k_1 x_1 x_2 + k_2 x_3 \\ \dot{x_2} = -k_1 x_1 x_2 + (k_2 + k_3) x_3 \end{cases}$$

We propose 4 different reduced BN structures:

- **Merged model**
- **Stacked model, Unstacked model**
- **No-var model**

In this talk, we present the **Merged model**, which offers a good balance between reduction and inference accuracy.

Different reduced DBN structures

How to represent the following reduced subsystem with a BN ?

$$\begin{cases} \dot{x_1} = -k_1 x_1 x_2 + k_2 x_3 \\ \dot{x_2} = -k_1 x_1 x_2 + (k_2 + k_3) x_3 \end{cases}$$

We propose 4 different reduced BN structures:

- **Merged model**
- **Stacked model, Unstacked model**
- **No-var model**

In this talk, we present the **Merged model**, which offers a good balance between reduction and inference accuracy.

Different reduced DBN structures

How to represent the following reduced subsystem with a BN ?

$$\begin{cases} \dot{x_1} = -k_1 x_1 x_2 + k_2 \boldsymbol{x_3} \\ \dot{x_2} = -k_1 x_1 x_2 + (k_2 + k_3) \boldsymbol{x_3} \end{cases}$$

We propose 4 different reduced BN structures:

- **Merged model**
- **Stacked model, Unstacked model**
- **No-var model**

In this talk, we present the **Merged model**, which offers a good balance between reduction and inference accuracy.

Different reduced DBN structures

How to represent the following reduced subsystem with a BN ?

$$\begin{cases} \dot{x_1} = -k_1 x_1 x_2 + k_2\big(-\boldsymbol{x_2} + \boldsymbol{x_3(0)} + \boldsymbol{x_2(0)}\big) \\ \dot{x_2} = -k_1 x_1 x_2 + (k_2 + k_3)\big(-\boldsymbol{x_2} + \boldsymbol{x_3(0)} + \boldsymbol{x_2(0)}\big) \end{cases}$$

We propose 4 different reduced BN structures:

- **Merged model**
- **Stacked model, Unstacked model**
- **No-var model**

In this talk, we present the **Merged model**, which offers a good balance between reduction and inference accuracy.

Different reduced DBN structures

How to represent the following reduced subsystem with a BN ?

$$\begin{cases} \dot{x_1} = -k_1 x_1 x_2 + k_2 \big(-\boldsymbol{x_2} + \cancel{\boldsymbol{x_3(0)} + \boldsymbol{x_2(0)}} \big) \\ \dot{x_2} = -k_1 x_1 x_2 + (k_2 + k_3) \big(-\boldsymbol{x_2} + \cancel{\boldsymbol{x_3(0)} + \boldsymbol{x_2(0)}} \big) \end{cases}$$

We propose 4 different reduced BN structures:

- **Merged model**
- **Stacked model, Unstacked model**
- **No-var model**

In this talk, we present the **Merged model**, which offers a good balance between reduction and inference accuracy.

Different reduced DBN structures

How to represent the following reduced subsystem with a BN ?

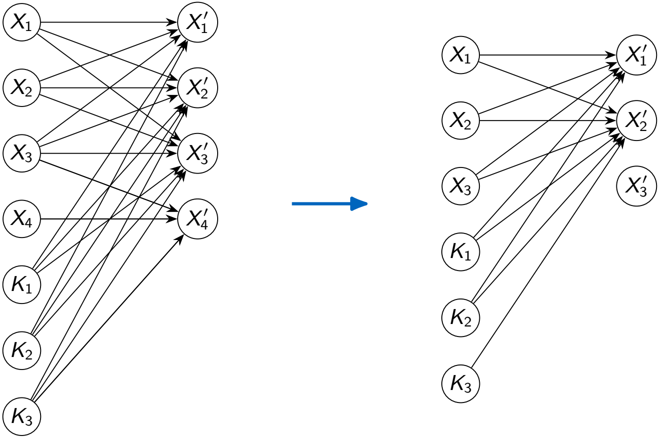
$$\begin{cases} \dot{x_1} = -k_1 x_1 x_2 + k_2 \boldsymbol{x_3} \\ \dot{x_2} = -k_1 x_1 x_2 + (k_2 + k_3) \boldsymbol{x_3} \end{cases}$$

We propose 4 different reduced BN structures:

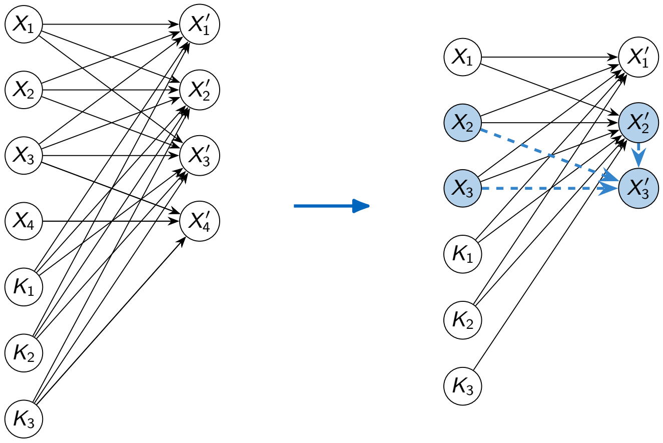
- **Merged model**
- **Stacked model, Unstacked model**
- **No-var model**

In this talk, we present the **Merged model**, which offers a good balance between reduction and inference accuracy.

Merged model



Merged model

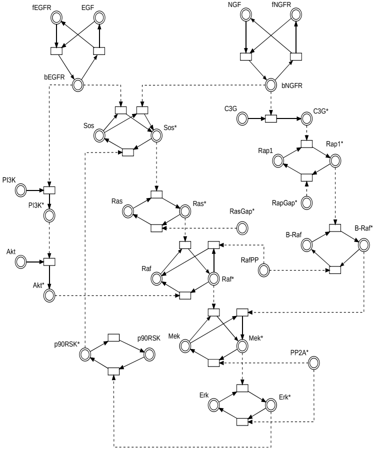


$$x_3(t) = -x_2(t) + x_3(0) + x_2(0)$$

Applications and implementation

Case study: EGF-NGF signaling pathway:

- 32 species,
- 48 parameters,
- 26 reactions.



Implementation: The approach was implemented in Python and is available online as a prototype called BayeSBML[6].

[6] https://git.lacl.fr/barbot/bayesbml

Applications and implementation

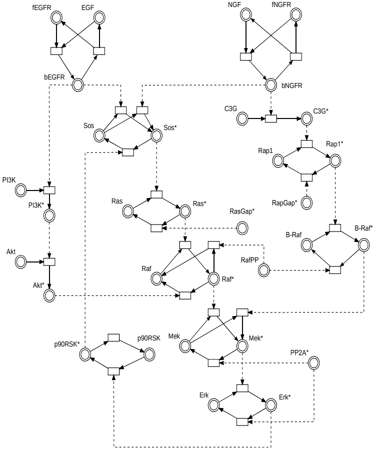
Case study: EGF-NGF signaling pathway:

- 32 species,
- 48 parameters,
- 26 reactions.

Implementation: The approach was implemented in Python and is available online as a prototype called `BayeSBML`[6].

QEST + FORMATS Artifacts Evaluation Functional

QEST + FORMATS Artifacts Evaluation Available



[6] https://git.lacl.fr/barbot/bayesbml

Fan-in analysis

Cumulative (Cum) Fan-in: The total number of arcs in the Bayesian Network (BN).
Maximum (Max) Fan-in: The highest number of incoming arcs to any derivative node.

Enzyme-catalyzed reaction:

- Original model: (Cum,Max) = (20,6) → 172500 total CPT rows.
- Merged model: (Cum,Max) = (**11**,6) → 93750 total CPT rows.

EGF-NGF:

- Original model: (Cum,Max) = (206,11) → 1.01e6 total CPT rows.
- Merged model: (Cum,Max) = (**94**,11) → 4.66e5 total CPT rows.

Fan-in analysis

Cumulative (Cum) Fan-in: The total number of arcs in the Bayesian Network (BN).
Maximum (Max) Fan-in: The highest number of incoming arcs to any derivative node.

Enzyme-catalyzed reaction:

- Original model: (Cum,Max) $=$ (20,6) $\rightarrow$ 172500 total CPT rows.
- Merged model: (Cum,Max) $=$ (**11**,6) $\rightarrow$ 93750 total CPT rows.

EGF-NGF:

- Original model: (Cum,Max) $=$ (206,11) $\rightarrow$ $1.01e6$ total CPT rows.
- Merged model: (Cum,Max) $=$ (**94**,11) $\rightarrow$ $4.66e5$ total CPT rows.

Fan-in analysis

Cumulative (Cum) Fan-in: The total number of arcs in the Bayesian Network (BN).
Maximum (Max) Fan-in: The highest number of incoming arcs to any derivative node.

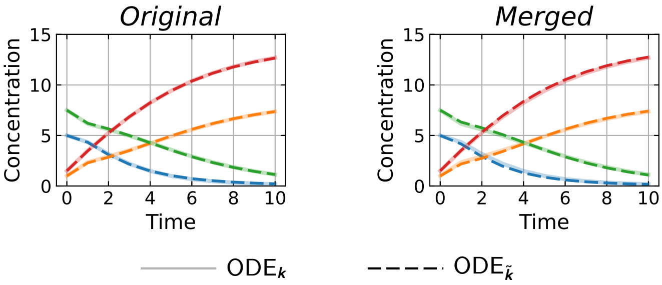
Enzyme-catalyzed reaction:

- Original model: (Cum,Max) = (20,6) → 172500 total CPT rows.
- Merged model: (Cum,Max) = (**11**,6) → 93750 total CPT rows.

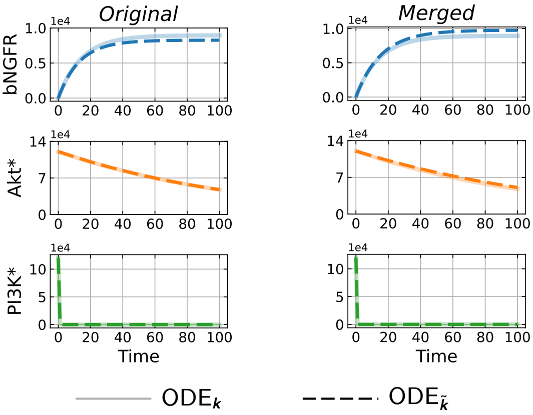
EGF-NGF:

- Original model: (Cum,Max) = (206,11) → $1.01e6$ total CPT rows.
- Merged model: (Cum,Max) = (**94**,11) → $4.66e5$ total CPT rows.

Numerical results: Enzyme-catalyzed reaction



Numerical results: EGF NGF case study



Original Model: ≈ 10 hours, Merged Model: ≈ 4 hours.

Conclusion

Conclusion

The proposed novel approach leverages a model simplification technique that directly impacts the structure of our probabilistic approximation.

- Marginal accuracy loss.
- Significant performance gains.
- Enhance the method's applicability to high-dimensional case studies.
- Implementation available online.

Future Work

- Collaborate with biologists to apply our methods to real-world data.
- Integrate efficient CPU parallelization and batching into our existing framework:
  $\Longrightarrow$ Accelerate numerical simulations (CPTs construction time).
  $\Longrightarrow$ Accelerate parameter optimization (exploration phase).

Thank You!

Any Questions?

Olivier Bouët-Willaumez
LACL, Université Paris-Est Créteil, France
https://olivbw.github.io/