# Efficient stochastic simulation of systems with multiple time scales via statistical abstraction

Luca Bortolussi[123] Dimitrios Milios[4] Guido Sanguinetti[45]

Modelling and Simulation Group, University of Saarland, Germany

Department of Mathematics and Geosciences, University of Trieste

CNR/ISTI, Pisa, Italy

School of Informatics, University of Edinburgh

SynthSys, Centre for Synthetic and Systems Biology, University of Edinburgh

16th of September 2015
Computational Methods in Systems Biology

## Multiple Time-Scales in Biological Systems

**The problem – Stiffness**

- Existence of fast and slow time-scales
- Challenge to mathematical and computational treatment of systems

**In the literature – Abstraction techniques**

- Simplify some scales of the model
- Abstractions are non-trivial and model-specific

**We propose**:

- Model abstraction based on statistical methodologies
- Learned abstractions automatically from (few) exploratory runs of the models

## Stochastic Simulation of Stiff Systems

The *Gillespie algorithm* is exact

- simulates every single reaction event
- High computational costs in presence of stiffness, where a small number of reactions dominate computations

## Stochastic Simulation of Stiff Systems

The *Gillespie algorithm* is exact

- simulates every single reaction event
- High computational costs in presence of stiffness, where a small number of reactions dominate computations

**Enzyme-substrate example:**

$$\begin{aligned} E + S &\xrightarrow{f_1(\vec{X})} ES, & f_1(\vec{X}) &= c_1 X_E X_S \\ ES &\xrightarrow{f_2(\vec{X})} E + S, & f_2(\vec{X}) &= c_2 X_{ES} \\ ES &\xrightarrow{f_3(\vec{X})} E + P, & f_3(\vec{X}) &= c_3 X_{ES} \end{aligned}$$

Assuming $c_1, c_2 \gg c_3$:

- too many reaction events for $R_1$ and $R_2$,
- while $R_3$ progresses very slowly

## Model Reduction

Reaction partitioning into $\mathcal{R}_{fast}$ and $\mathcal{R}_{slow}$:

- based on their kinetic constants

System Variables: $\vec{X} = (\vec{Y}, \vec{Z})$

**Fast Variables:** $\vec{Y} = Y_1, \ldots, Y_m$

- Affected by either fast or slow reactions

**Slow Variables:** $\vec{Z} = Z_1, \ldots, Z_s$

- Affected by slow reactions only

## Model Reduction

Reaction partitioning into $\mathcal{R}_{fast}$ and $\mathcal{R}_{slow}$:

- based on their kinetic constants

System Variables: $\vec{X} = (\vec{Y}, \vec{Z})$

**Fast Variables:** $\vec{Y} = Y_1, \ldots, Y_m$

- Affected by either fast or slow reactions

**Slow Variables:** $\vec{Z} = Z_1, \ldots, Z_s$

- Affected by slow reactions only

**Enzyme-substrate example:**
We assume that $c_1, c_2 \gg c_3$

- fast and slow reactions: $\mathcal{R}_{fast} = \{R_1, R_2\}$ and $\mathcal{R}_{slow} = \{R_3\}$
- fast variables $\vec{Y} = (X_E, X_S, X_{ES})$ and slow variables $\vec{Z} = (X_P)$

## The Fast Subsystem

System State $\vec{Y}$

- Affected by either $\mathcal{R}_{fast}$ or $\mathcal{R}_{slow}$
- Slow reactions rarely occur — can be ignored
- Fast rates may depend on the slow variables

## The Fast Subsystem

System State $\vec{Y}$

- Affected by either $\mathcal{R}_{fast}$ or $\mathcal{R}_{slow}$
- Slow reactions rarely occur — can be ignored
- Fast rates may depend on the slow variables

**Conditional Fast subsystem:**

- Parametrised by the concentration $\vec{z}$ of slow variables
    - $\vec{z} = \vec{Z}/V$ in a volume $V$

## The Fast Subsystem

System State $\vec{Y}$

- Affected by either $\mathcal{R}_{fast}$ or $\mathcal{R}_{slow}$
- Slow reactions rarely occur — can be ignored
- Fast rates may depend on the slow variables

**Conditional Fast subsystem:**

- Parametrised by the concentration $\vec{z}$ of slow variables
  - $\vec{z} = \vec{Z}/V$ in a volume $V$

$$\begin{aligned} E + S &\xrightarrow{f_1(\vec{Y},\vec{z})} ES, &\quad f_1(\vec{Y},\vec{z}) = c_1 X_E (N - X_{ES} - X_P) \\ ES &\xrightarrow{f_2(\vec{Y},\vec{z})} E + S, &\quad f_2(\vec{Y},\vec{z}) = c_2 X_{ES} \end{aligned}$$

**Assumption**: Quickly reaches equilibrium for any $\vec{z}$

## The Slow Subsystem

System State $\vec{Z}$

- Affected by $\mathcal{R}_{slow}$
- Slow rates may depend on the fast variables
  - Senses the fast system only via its steady state distribution

## The Slow Subsystem

System State $\vec{Z}$

- Affected by $\mathcal{R}_{slow}$
- Slow rates may depend on the fast variables
  - Senses the fast system only via its steady state distribution

All $R_j$ in $\mathcal{R}_{slow}$ are modified by:

1. removing the fast variables
2. replacing the rate function $f_j(\vec{Y}, \vec{z})$ by:

$$\hat{f}_j(\vec{z}) = \mathbb{E}_{|\vec{z}}[f_j(\vec{Y}, \vec{z})]$$

Average out fast variables wrt their steady state distribution

## The Slow Subsystem

System State $\vec{Z}$

- Affected by $\mathcal{R}_{slow}$
- Slow rates may depend on the fast variables
  - Senses the fast system only via its steady state distribution

All $R_j$ in $\mathcal{R}_{slow}$ are modified by:

1. removing the fast variables
2. replacing the rate function $f_j(\vec{Y}, \vec{z})$ by:

$$\hat{f}_j(\vec{z}) = \mathbb{E}_{|\vec{z}}[f_j(\vec{Y}, \vec{z})]$$

Average out fast variables wrt their steady state distribution

$$\emptyset \quad \xrightarrow{\hat{f}_3(\vec{z})} \quad P, \qquad \hat{f}_3(\vec{z}) = \mathbb{E}_{|\vec{z}}[f_3(\vec{Y}, \vec{z})]$$

## Slow-scale Simulation

Simulation of the slow subsystem:

- Derive expectations $\hat{f}_j(\vec{z}), \quad \forall R_j \in \mathcal{R}_{slow}$
- Fast reactions are ignored

## Slow-scale Simulation

Simulation of the slow subsystem:

- Derive expectations $\hat{f}_j(\vec{z}), \quad \forall R_j \in \mathcal{R}_{slow}$
- Fast reactions are ignored

**In the literature**:

- $\hat{f}_j(\vec{z})$ is given by model-dependent expressions
- Applicability is limited
- Required expertise on the modeller side

## Slow-scale Simulation

Simulation of the slow subsystem:

- Derive expectations $\hat{f}_j(\vec{z}), \quad \forall R_j \in \mathcal{R}_{slow}$
- Fast reactions are ignored

**In the literature**:

- $\hat{f}_j(\vec{z})$ is given by model-dependent expressions
- Applicability is limited
- Required expertise on the modeller side

**A more generic approach**:

- Construct a lookup table for the rate expectations
  - Explore the state-space of $\vec{Z}$
  - Estimate $\hat{f}_j(\vec{z})$ statistically
- **Problem**: The number of states for $\vec{Z}$ could be too large

## Approximation of Rate Expectations

### Theorem

*The equilibrium statistics of the fast variables are a continuous function of the slow variables (rescaled to concentrations)*

**Our approach**:

- Statistical estimate of the continuous function $\hat{f}_j(\vec{z})$
- Use a few samples from the slow state-space
- Interpolate via Gaussian Processes Regression
- Exhaustive state-space exploration is avoided

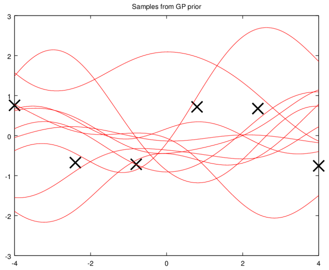
## Gaussian Process Regression

- Place a GP prior over $f$

$$p(\mathbf{f}) = \mathcal{N}(0, K)$$

- Assume noisy observations $\mathbf{y} = \mathbf{f} + \epsilon$

$$p(\mathbf{y} \mid \mathbf{f}) = \mathcal{N}(\mathbf{f}, \sigma^2 I)$$



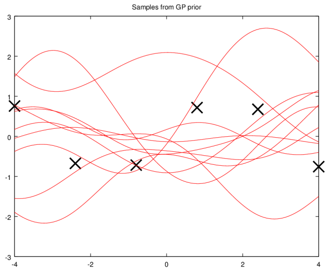
## Gaussian Process Regression

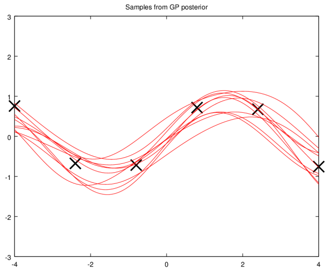
- Place a GP prior over $f$

$$p(\mathbf{f}) = \mathcal{N}(0, K)$$

- Assume noisy observations $\mathbf{y} = \mathbf{f} + \epsilon$

$$p(\mathbf{y} \mid \mathbf{f}) = \mathcal{N}(\mathbf{f}, \sigma^2 I)$$





$$p(\mathbf{f} \mid \mathbf{y}) = \frac{1}{Z} \underbrace{p(\mathbf{f})}_{\text{Gaussian Prior}} \underbrace{p(\mathbf{y} \mid \mathbf{f})}_{\text{Gaussian Noise}}$$

## Stochastic Simulation via Statistical Abstraction

The SA-SSA Approach

**Initialisation Phase:** For a grid of *n* states of the slow process:

- Calculated rate expectations:

$$\hat{f}_j(\vec{z}) = 1/t_f \int_{t_0}^{t_0+t_f} f_j(\vec{Y}, \vec{z}) dt$$

- $t_0$: time required to reach equilibrium (estimated by heuristic)
- Train a GP regression model

## Stochastic Simulation via Statistical Abstraction

**Initialisation Phase:** For a grid of $n$ states of the slow process:

- Calculated rate expectations:

$$\hat{f}_j(\vec{z}) = 1/t_f \int_{t_0}^{t_0+t_f} f_j(\vec{Y}, \vec{z})dt$$

- $t_0$: time required to reach equilibrium (estimated by heuristic)
- Train a GP regression model

**Simulation Phase:**

- Simulate the slow system (ignoring the fast variables/reactions)
- Using the rate expectations as given by the GP regression model

## Cost of SA-SSA

**Pre-simulation Cost** (only during initialisation)

- Few samples of the slow system state-space
- Excessive simulation of the fast system is avoided

**Regression Cost** (only during initialisation)

- Dominated by the solution of a linear system — $O(n^2)$

**Cost of using the Analytical Approximation** (during simulation)

- Produce estimation from $n$ training points — $O(n)$
- For higher-dimensional slow state-spaces, sparse schemes are necessary

Note: Can learn rate expectations as functions of the system parameters

- approximate an entire family of stiff systems

## Enzyme-substrate system — Parameter exploration

Let $c_1$ vary in the range $[0.01, 1]$

- The system remains stiff
- Sampled a grid of 1000 values for $X_P \in [0, 3000]$ and $c_1 \in [0.01, 1]$

Table: Relative mean error values for approximating the mean value of $X_P$, for $10^3$ simulation runs.

| | $P$ (RME) | | | |
|---|---|---|---|---|
| Time | $c_1 = 0.01$ | $c_1 = 0.1$ | $c_1 = 0.5$ | $c_1 = 1$ |
| $5 \times 10^4$ | $1.83 \times 10^{-3}$ | $9.08 \times 10^{-4}$ | $2.35 \times 10^{-3}$ | $2.17 \times 10^{-3}$ |
| $10 \times 10^4$ | $1.20 \times 10^{-3}$ | $1.49 \times 10^{-3}$ | $1.94 \times 10^{-3}$ | $2.87 \times 10^{-3}$ |
| $18 \times 10^4$ | $8.04 \times 10^{-4}$ | $3.73 \times 10^{-5}$ | $4.49 \times 10^{-4}$ | $3.05 \times 10^{-4}$ |
| $20 \times 10^4$ | $9.13 \times 10^{-4}$ | $4.56 \times 10^{-5}$ | $6.06 \times 10^{-5}$ | $3.26 \times 10^{-5}$ |

**Gillespie algorithm:** 1911 sec
**SA-SSA:** 32 sec + 3.562 sec for initialisation

## Weinan et al 2005

Weinan E, Di Liu, and Eric Vanden-Eijnden. Nested stochastic simulation algorithm for chemical kinetic systems with disparate rates. *The Journal of Chemical Physics*, 123(19), 2005.

The *Nested Stochastic Simulation Algorithm* (Nested-SSA) is proposed to approximate the steady-state of the fast subsystem

- The fast subsystem is only simulated up to a given step
    - .. assuming that steady-state is reached by then
- Completely transparent wrt the slow process

We have implemented Nested-SSA, to produce comparative results

- The step parameter for Nested-SSA has been explored experimentally such that the efficiency of both simulation approaches has been roughly the same

## Enzyme-substrate system — Accuracy results

Initial state: $\vec{X}_0 = (X_E, X_S, X_{ES}, X_P) = (220, 3000, 0, 0)$.

- The rate expectation for $R_3$ has been approximated via GP regression
- Sampled 1000 states for the slow variable $P$ between 0 and 3000

Table: Enzyme-substrate model: histogram distances for $10^3$ simulation runs (estimated self-distance: 0.252).

| | $P$ | |
|---|---|---|
| Time | Nested-SSA | SA-SSA |
| $5 \times 10^4$ | 0.290 | 0.246 |
| $10 \times 10^4$ | 0.250 | 0.204 |
| $18 \times 10^4$ | 1.016 | 0.160 |
| $20 \times 10^4$ | 0.940 | 0.142 |

## Viral Infection model

Reactions: $\mathcal{R}_{fast} = \{R_3, R_5\}$ and $\mathcal{R}_{slow} = \{R_1, R_2, R_4, R_6\}$

Fast variables $\vec{Y} = (X_S)$, and slow variables $\vec{Z} = (X_G, X_T)$

$$\begin{array}{ccll} \emptyset & \xrightarrow{f_3(\vec{Y},\vec{z})} & S, & f_3(\vec{Y},\vec{z}) = k_3 X_T \\ S & \xrightarrow{f_5(\vec{Y},\vec{z})} & \emptyset, & f_5(\vec{Y},\vec{z}) = k_5 X_S \end{array}$$

$$\begin{array}{ccll} T & \xrightarrow{f_1(\vec{z})} & G + T, & f_1(\vec{z}) = k_1 X_T \\ G & \xrightarrow{f_2(\vec{z})} & T, & f_2(\vec{z}) = k_2 X_G \\ T & \xrightarrow{f_4(\vec{z})} & \emptyset, & f_4(\vec{z}) = k_4 X_T \\ G & \xrightarrow{\hat{f}_6(\vec{z})} & V, & \hat{f}_6(\vec{z}) = \mathbb{E}_{|\vec{z}}[f_6(\vec{Y},\vec{z})] \end{array}$$

The rate $\hat{f}_6(\vec{z})$ depends on $X_G$ directly, and on $X_T$ indirectly

- $T$ affects the steady-state of the fast process

## Viral Infection model — Accuracy results

Random grid of 256 uniformly distributed population values for $G$ and $T$,

- given upper bounds of 500 and 100 molecules correspondingly

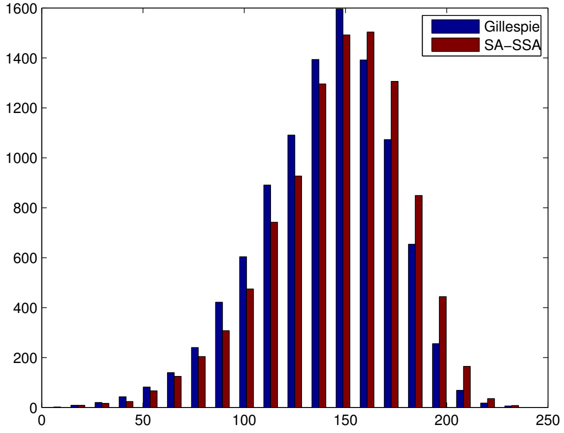
Naïve exploration of the rate expectation would require 50000 evaluations

Table: Viral infection model: histogram distances for $10^3$ simulation runs (estimated self-distance: 0.252).

| | $G$ | | $T$ | |
|---|---|---|---|---|
| Time | Nested-SSA | SA-SSA | Nested-SSA | SA-SSA |
| 50 | 0.988 | 0.308 | 0.548 | 0.242 |
| 100 | 0.244 | 0.414 | 0.154 | 0.226 |
| 200 | 0.388 | 0.406 | 0.156 | 0.204 |
| 500 | 0.346 | 0.432 | 0.198 | 0.238 |

## Viral Infection model — Accuracy results

Distribution of $X_G$ at $t = 50$



## Efficiency results

Table: Execution times in seconds for $10^3$ simulation runs.

| Method | | Enzyme-substrate | Viral model |
|---|---|---|---|
| SA-SSA | Pre-simulation | 0.291 | 26.11 |
| | Hyperparam. opt. | 1.484 | 1.68 |
| | Training | 0.080 | 0.05 |
| | Total initialisation | 1.855 | 27.84 |
| | Simulation | 153 | 316 |
| | Exact SSA | 6947 | 2410 |

## Conclusions

**Time-scale separation**

- In the literature: exploit structure to produce estimations for the rate expectations for the slow process
- We proposed SA-SSA:
  rate expectations are approximated via machine learning
- Learn the rate expectations as functions of the parameters as well
- Similar or better accuracy than Nested-SSA

**Future Work**

- Efficient simulation in presence of multiple spatio-temporal scales
- Abstraction of intra-cellular dynamics for cell population models

## Acknowledgements...