# CALIBRATION OF PROCESS ALGEBRA MODELS OF DISCRETELY OBSERVED STOCHASTIC BIOCHEMICAL SYSTEMS

**Paola Lecca**

The Microsoft Research – University of Trento
Centre for Computational and Systems Biology
Piazza Manci 17, 38123 Povo (Trento), Italy

lecca@cosbi,eu

**ABSTRACT**
We present a maximum likelihood method for inferring kinetics of stochastic systems of chemical reactions, given discrete time-course observations of the abundance of either some or all of the molecular species and a BlenX model of the system. BlenX is a process calculus providing a tool and algebraic laws for a high-level description of interactions, communications, and synchronizations between processes representing the biomolecules. BlenX offers an efficient alternative to differential equations, but it poses different challenges to the model calibration. The main difficulty is the sampling of the reaction pathways between two observed states. We define a maximum likelihood function in terms of reaction propensities and we estimate it by sampling the intermediate pathways from the transition system of a BlenX. The method of sampling the transition system is inspired to the elementary mode analysis. Our method is illustrated with the example of a BlenX model of chaperone-assisted protein folding.

Keywords: BlenX, parameter estimation, maximum likelihood estimation.

## 1. INTRODUCTION

Modelling the time evolution of biological systems requires the specification of the interactions among the biochemical species and the kinetic parameters of these interactions. The choice of a language able to express the main features of biological systems and the methods for estimating the model parameters (model calibration) are two interdependent. In this article, we introduce the BlenX language (Dematté, et al., 2008 (a) and (b)) for modeling biological processes and a way to calibrate a BlenX-specified model using discrete time-course observations of either some or all of the molecular species.

The majority of the models of inter- and intra-cellular dynamics are specified in ordinary differential equations. These equations are usually employed by physicist to formalize the natural laws and to describe the dynamics of the inert matter. However, the recent achievement of the systems biology paradigm highlights the need of a new systematic approach to modeling systems belonging to living matter. This need can be translated into the necessity to develop new mathematical methods and tools to model living systems, considering that these systems require mathematical and computational approaches substantially different from those used to model inert matter. Multifunctionality of biochemical complexes, parallelism and concurrency of their interactions and modular structure of the network of interactions are the main features characterizing a biological system. Process algebras (or process calculi) are currently proposing as suitable formalisms for the specification of a biological process (some examples are: stochastic π-calculus (Priami, 1995), BioAmbients (Regev, et al., 2004), Brane Calculi (Cardelli, 2005), CCS-R (Danos & Krivine, 2004), k-calculus (Danos & Laneve, 2004), PEPA (Gilmore & Hillstone, 1994)). Usually, these algebras are applied to the study of concurrent processes. The tools of the process algebras are algebraic languages for the specification of processes and the formulation of statements about them, together with calculi for the verification of these statements. Process calculi provide a tool for the high-level description of interactions, communications, and synchronizations between a collection of independent agents or processes. The use of these calculi in modelling biological system is based on a new abstraction of the physical concepts of interacting molecules, interactions and change of state. Interacting molecules are represented by processes. Interactions between molecules are represented by synchronized communication. The change of state consequent to an interaction is described by the modification that processes undergo after the realization of their communication.

In this article we focus on a new member of the family of process algebra: the BlenX language. It has been developed by our lab (CoSBi, 2011) to extend the expression capabilities of the stochastic π-calculus. As in stochastic π-calculus, also in BlenX models consist of agents (processes) which stochastically engage actions. However, with BlenX we can describe more easily spatial structures like membranes, compartments,

interaction domains, and the formation of biological complexes driven by chemical/physical affinities. Currently, BlenX can be used to specify continuous time stochastic systems. In fact, the stochastic simulation algorithm of the BlenX simulator is an efficient variant of the Gillespie algorithm (Gillespie, 1977). In Gillespie-like approaches every reaction is explicitly simulated. When simulated, a Gillespie realization represents a random walk that exactly provides the distribution of the Chemical Master Equation.

The communications representing the physico-chemical interactions between the biological entities are associated to a rate constant that quantifies the specific speed of the communication and reflects the kinetic rate constant (and/or the affinity) of the reaction. This rate constant is the parameter of an exponential probability distribution of the waiting time of reactions. Thus, the waiting time of a reaction is a realization of a random variable exponentially distributed with parameter equal to the rate constant of the reaction. The stochastic simulation algorithm generates random numbers to determine the next reaction to occur as well as the time at which the reaction occurs. The time evolution of the system proceeds by jumps from one state to another. The state of the system at time *t* is given by the number of molecules (or more generally of biological - biochemical entities) of each species included in the system at that time. Therefore, first of all, the calibration of a BlenX model is the calibration of a stochastic model.

The existent method for parameter inference in stochastic system belongs to one of the following two categories: maximum-likelihood based approach and Bayesian inference approach. A comprehensive review of these methods can be found in the introduction of recent work of Y. Wang et al. (Wang, Christley, Mjolsness, & Xie, 2010), R. J. Boys et al. (Boys, Wilkinson, & Kirkwood, 2008) and in P. Lecca et al. (Lecca, Palmisano, Ihekwaba, & Priami, 2010). Here we focus on maximum likelihood (ML) appraoches, following our previous studies on ML inference models (Lecca, Palmisano, Ihekwaba, & Priami, 2010).

Most proposed methods for parameter inference in stochastic biochemical models consider how to calculate the maximum likelihood for the rate parameter values given a stochastic model and discrete experimental data of the amount of molecules of all or only of some species. Since for biological systems of realistic size and complexity, the likelihood function is computationally intractable, these methods either perform exact inference on an approximated model where the likelihood computation is tractable, or they approximate the likelihood with a more tractable function, or some combination of the two. In this paper we refer to a method of parameter estimation presented in (Wang, Christley, Mjolsness, & Xie, 2010) and we show how it can be adapted to estimate the kinetic parameters of a BlenX model. In particular, we show how the transition system of a BlenX model can be sampled to calculate the maximum likelihood function withoud any need of simulating the model. The paper proceeds as follows: in Section 2 we introduce the reader to the BlenX language and present the model on which we show the eprfomance of the infrence method; in Section 3 we present the method and the results, and finally in Section 4 we give some conclusions.

## 2. THE BLENX LANGUAGE: AN OVERVIEW

BlenX is a process algebra-based stochastic programming language that shares features with stochastic π-calculus \cite{priami95} and Beta-binders \cite{pqp}. BlenX, as these other members of the family of process algebra-based languages, has a strong focus on the interactions of entities. BlenX is explicitly designed to model the interactions of biological entities such as proteins and other biochemical species. It is a stochastic language in the sense that the probability and speed of the interactions and actions governing the time evolution of the system are specified in the body of the programs written in this language.

In BlenX, each species is given with an abstract entity that we call a *box*. Each box has a number of connectivity interfaces called *binders*, and it is equipped with an internal program. The sites of interaction are represented as binders on the box surface. For example in Figure 1, each box has only one binder. Binders are identified by their names, e.g., `x` and their types, e.g., `A`.

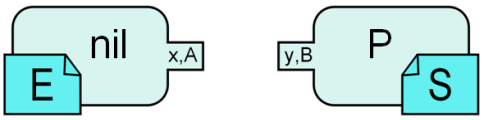


**Figure 1: an enzyme E and its substrate S can be represented by boxes equipped with interaction sites on the interfaces, i. e. the binders (x, A) and (y, B), and an internal processes, e.g. the deadlock process nil and a process P, respectively.**

A box can stochastically interact with another box, and change state as a result of this interaction with respect to the actions specified in its internal program. Alternatively, a box can autonomously change state by stochastically performing an action that is given in its internal program. For instance, the complexation of an enzyme E and a substrate S can be described in a BlenX model with the boxes depicted in Figure 1, where these boxes interact and bind with their binders. Then the interaction rate, specified in the BlenX code, determines the rate of the association. The internal program, which can be `nil` as it is the case for E here, determines the *actions* the box can undertake after this interaction.

The nil process does nothing (it is the dadlock process). Other stochastic actions that a BlenX box can perform are summarized as follows: a box can

- communicate with another box that is bound to it (or with itself) by performing

- an input action, e.g., `x?(message)` that is complementary to the output action, e.g., `x!(message)`, of the other box, or vice versa,; and this way send or receive a message;
- perform a stochastic `delay` action;
- change (`ch`) the type of one of its interfaces;
- eliminate itself by performing a `die` action;
- `expose` a new binder;
- `hide` one of its binders;
- `unhide` a binder which is hidden.

In addition to these actions, there are also other programming constructs available such as if-then statements and state-checks. For example, let us consider the box S in Figure 1. We can program this box by defining program `P` so that it will change its type from `B` to `C` if it is bound:

```
if (y,B) and (y,bound) then ch(y,C) endif
```

In BlenX, following the process algebra tradition, we can compose actions by using algebraic composition operators to define increasingly complex behaviors. We can sequentially compose actions by resorting to the prefix-operator, which is written as an infix dot. For instance,

```
ch(y,C).hide(y).nil
```

denotes a program that first performs change action and then hides the changed binder.

Programs can be composed in parallel. Parallel composition (denoted by the infix operator "|", for instance `P|Q`, allows the description of programs, which may run independently in parallel and also synchronize on *complementary actions* (i.e., *input* and *output* over the same channel).

The `rep` operator replicates copies of the process passed as argument. Only guarded replication is used, i. e. the process argument of this operator must be prefixed by an action that forbids any other action of the process until the first action has been executed.

Programs can also be composed by *stochastic choice*, denoted with the summation operator "+". The sum of processes `P` and `Q`, `P + Q` behaves either as `P` or as `Q`, determined by specific speed (i. e. rate constants) defined for `P` and `Q`. The selection of one discards the other forever.

In BlenX, we use `events`, which are programming constructs for expressing actions that are enabled by global conditions. For example, in the model presented here, we use the `new` construct to introduce new molecules of a species if their amount reaches a minimum threshold. For instance in:

```
when (protein : |protein| < 10000 : r)
new(500);
```

the amount of species protein is increased by 500 units when it becomes less 10,000 units. The increasing rate is `r`.

### 2.1. The case study: chaperone-assisted protein folding

In this section we briefly describe the main mechanisms of chaperone-assisted protein folding and present the code of a BlenX model of this process. A more detailed descriptions of the biological processes and of the BlenX specification can be found in (Lecca P., 2011).

The ability of the cell to handle misfolded proteins is expressed by some complexes of macromolecules, called *chaperones*. Molecular chaperones interact with unfolded or partially folded protein subunits, e.g. nascent chains emerging from the ribosome, or extended chains being translocated across sub-cellular membranes. They prevent inappropriate association or aggregation of exposed hydrophobic surfaces and direct their substrates into productive folding, transport or degradation pathways. In the healthy cells, if a protein does not assume the correct 3D shape, or a cellular stress induces a right-folded protein to assume a wrong folding, the chaperones act to re-shape it correctly.

$$\mathrm{R}_1 : \mathrm{protein} + \mathrm{chaperone} \xrightarrow{k_1} \mathrm{healthy\_protein} + \mathrm{chaperone}$$

$$\mathrm{R}_2 : \mathrm{protein} + \mathrm{chaperone} \xrightarrow{k_2} \mathrm{faulty\_protein} + \mathrm{chaperone}$$

$$\mathrm{R}_3 : \mathrm{parkin} + \mathrm{ubiquitin} \xrightarrow{k_3} \mathrm{parkin} \cdot \mathrm{ubiquitin}$$

$$\mathrm{R}_4 : \mathrm{faulty\_protein} + \mathrm{parkin} \cdot \mathrm{ubiquitin} \xrightarrow{k_4} \mathrm{faulty\_protein} \cdot \mathrm{ubiquitin} + \mathrm{parkin}$$

$$\mathrm{R}_5 : \mathrm{faulty\_protein} \cdot \mathrm{ubiquitin} + \mathrm{proteasome} \xrightarrow{k_5} \mathrm{protesome}$$

**Figure 2: interactions between protein and chaperone, and between faulty protein and ubiquitin-proteasome system.**

In the case in which the protein is still not correctly refolded, the cellular ubiquitin-proteasome targets and degrades it before the faulty protein can cause damages. The protein parkin mediates the targeting of misfolded proteins for degradation by moving the molecules of ubiquitin on these proteins. The proteasome machinery recognizes the ubiquitinated proteins and degrades

them. The set of reactions driving the dynamics of the model is reported in Figure 2.

In the following, we report the BlenX code describing the main interaction among the component of the systems: the nascent protein, the chaperone, the ubiquitin, the parkin and the proteasome, all represented by boxes. The parameters of the model are specified in the body of the code and are bold-faced and grey highlighted. The rate parameters used in this model and the initial amounts of molecular species are expressed in arbitrary units, and qualitatively reproduce the real dynamic observed dynamics.

The Figure 3 depicts the boxes representing each entity involved in the systems and the interactions on dedicated binders.

```
[time=200] // absolute simulation time

<<BASERATE: 100>> // basal rate

// A nascent protein interacts with a
// molecular chaperone

let protein : bproc =
 #(y:100, P), #(ubi:100, U), 10
#(prot:100000,PTSP)
[
  y?().ch(1000,y, DR).hide(1,ubi).nil
  + y?().ch(1000,y,DW).ubi?().
    prot?().die(1).nil
 ];

// Definition of chaperone

let chaperone : bproc = #(x:100, C)
[rep x!().nil];

// Definition of parkin bioprocess

let parkin : bproc =
#(to_ubiquitin:0.5, T_UB)
[to_ubiquitin!().nil];

let ubiquitin : bproc =
#h(u:1, UB), #(actp:1, UB2),
#(from_parkin:0.5, F_PARK)
[
 from_parkin?().unhide(0.5,u).u!().actp!()
];

let proteasome : bproc =
#h(pt:50000, PTS), #(actv:500000, ACT),
[
 delay(0.1).(actv?().unhide(600000,pt).
 pt!().nil)
];

// Production of ubiquitin, parkin and nascent
// proteins

when (ubiquitin: |ubiquitin| = 0: inf) new(10000);
when (parkin: |parkin| = 0 : inf) new(10000);
when (protein :: protein_prod) new(1);

// Initial amount of components
run 5000 protein || 10000 parkin ||
10000 chaperone || 10000 ubiquitin ||
10000 proteasome
```

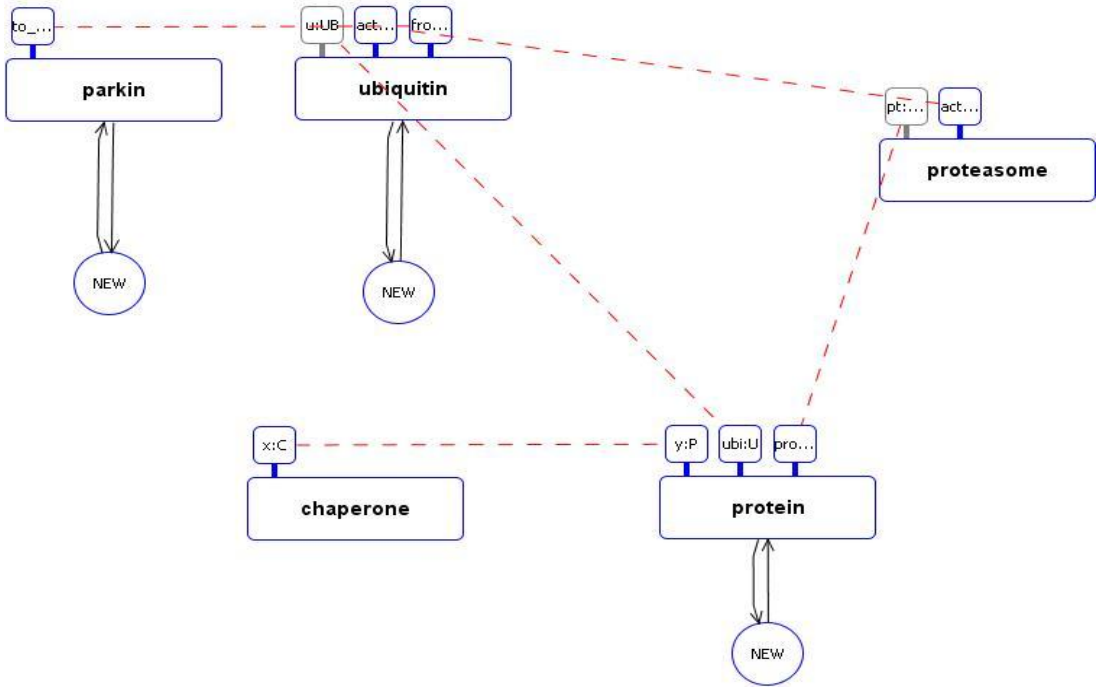


**Figure 3: picture representing the boxes of protein, chaperone, parkin, ubiquitin and proteasome. A dotted ted line connect the interacting boxes through the communication binders. The model also consider the production of protein, parkin and ubiquiting molecules represented with the circle "NEW".**

The structured operational (interleaving) semantics of the language is used to generate a labelled transition system. A state transition system is an abstract machine consisting of a set of states and transitions between states. The state transition system of the model presented in this paper is shown in Figure 4.

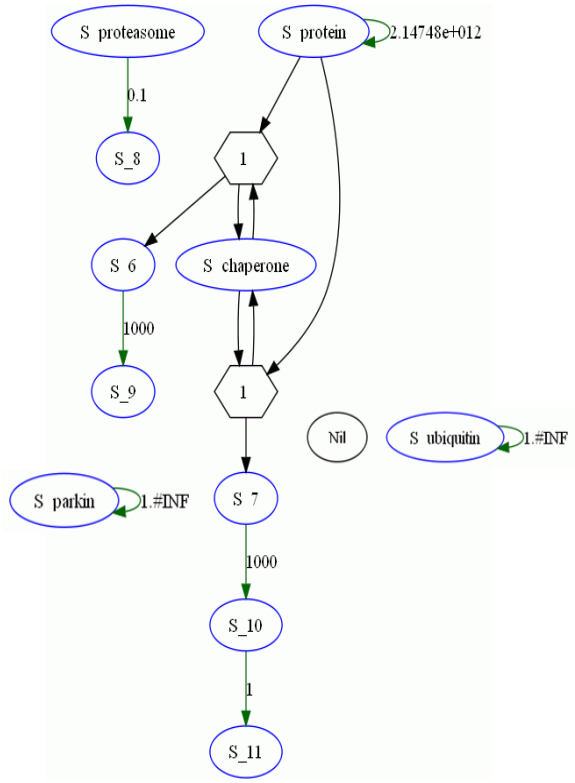


**Figure 4: graph of the state transition system of the BlenX model of chaperone-assisted protein folding and protein ubiquitination. "S" stays for species. All the intermediate species are showed and numbered. The labels on the arrows indicate the rate constant values. Where no label is specified, it is assumed to be equal to the basal rate constant defined in the BlenX code.**

## 3. ML-BASED PARAMETER INFERENCE

In this section we first review the key-points of method developed by Y. Wang et al. for the estimation of parameters of stochastic systems. Then, we show how these ideas can be adapted and used for the estimation of the rate constant of a BlenX model from experimental measurements of the amount of molecules at discrete time points.

### 3.1. The likelihood function

Our goal is to estimate the rate parameters of a stochastic Markov process algebra based on the observations at a set of discrete time points. In this section we review the methods of Y. Wang et al. (Wang, Christley, Mjolsness, & Xie, 2010).

Suppose we have the vector of the observations

$$\mathbf{X}(t) \equiv \{\hat{X}_\Gamma(t_1), \hat{X}_\Gamma(t_2), \ldots, \hat{X}_\Gamma(t_m)\} \qquad (3.1.1)$$

of the system at $m$ discrete time points $\{t_1, t_2, ..., t_m\}$ for a subset of species $\Gamma \subseteq \{1, ..., N\}$. Denoting the likelihood of the observations for a given set of rate parameters by

$$L(\hat{X}_\Gamma(t_1), \hat{X}_\Gamma(t_2), \ldots, \hat{X}_\Gamma(t_m); \boldsymbol{\Theta}),$$

we estimate the rate parameters by maximizing the likelihood function with respect to the parameters.

Suppose that the reaction system involves $M$ reactions, $R_1, \ldots, R_M$. Denote with $X = (x_1, \ldots, x_n)$ the state vector of the system. Each reaction has an associated propensity, also called *hazard function*, $h_i(X, \boldsymbol{\Theta})$. $\boldsymbol{\Theta}$ is the set of rate parameters associated with the reactions. The hazard function determines the rate of the transition probability out of state $X$ due to the reaction of type $i$. For convenience, as in (Wang, Christley, Mjolsness, & Xie, 2010), we adopt the following compact representation of a reaction system:

$$\mathcal{U}S \longrightarrow \mathcal{V}S$$

where $\mathcal{U} = [u_{ij}]$ are $M \times N$stoichiometry matrices. It is useful to introduce also the *net effect reaction matrix*

$$\mathcal{A} = \mathcal{U} - \mathcal{V} \qquad (3.1.2)$$

which reports the net change of species numbers associated with a reaction.

Denote $P(X, t)$ the probability of the system in state $X$at time $t$. For a time increment $\Delta t$, $P(X, t + \Delta t)$ can be written as the sum of probabilities of the number of ways in which the system can reach or leave the current state:

$$P(X, t + \Delta t) = \sum_{i=1}^{M} \left[ h_i(X - A_i, \boldsymbol{\Theta}) P(X - A_i, t) \Delta t + \left( 1 - \sum_{i=1}^{M} h_i(X, \boldsymbol{\Theta}) \Delta t \right) \right] P(X, t) \qquad (3.1.3)$$

where $A_j$ is the i-th row of the matrix $\mathcal{A}$. In the limit of $\Delta t \to 0$, Eq. (3.1.3) becomes

$$\frac{d}{dt}P(X,t) = \sum_{X'}\sum_{i=1}^{M}\left[h_i(X',\boldsymbol{\Theta})\delta_{X',X-A_i} - h_i(X,\boldsymbol{\Theta})\delta_{X',X}\right] \quad (3.1.4)$$

where $\delta_{X',X}$ is the Kronecker delta function. For our convenience we introduce $H_{X',X}$ as follows:

$$H_{X',X} = \sum_{i=1}^{M}\left[h_i(X',\boldsymbol{\Theta})\delta_{X',X-A_i} - h_i(X,\boldsymbol{\Theta})\delta_{X',X}\right] \quad (3.1.5)$$

For simplicity, consider a single time interval $[t_k, t_{k+1}]$ between two measurements of the abundance of the species $j$ ($j = 1,\ldots,N$)), $\hat{X}_j(t_k)$ and $\hat{X}_j(t_{k+1})$. We discretize this time interval in $K$subintervals and denote the system state by $\{X^\nu|\nu = 1,2,\ldots,K\}$. Therefore $\hat{X}^0 \equiv \hat{X}(t_k)$ and $\hat{X}^K \equiv \hat{X}(t_{k+1})$ are the full observations available at the start and at the end of this time interval. All $X^\nu$ are the intermediate states not directly observable. Using the Markov property of the stochastic process, the likelihood of observing $\hat{X}_j^0$ and $\hat{X}_j^K$ under a model with parameters $\boldsymbol{\Theta}$ is

$$L(\hat{X}_j^0, \hat{X}_j^K;\boldsymbol{\Theta}) = \sum_{\hat{X}_j^1,\ldots,\hat{X}_j^{K-1}} P(\hat{X}_j^0)\prod_{\nu=0}^{K-1} P(\hat{X}^{\nu+1}|\hat{X}^\nu;\boldsymbol{\Theta}) \quad (3.1.6)$$

If $K \gg 1$, then

$$P(\hat{X}^{\nu+1}|\hat{X}^\nu;\boldsymbol{\Theta}) \approx \delta_{\hat{X}^\nu,\hat{X}^{\nu+1}} + \frac{1}{K}H_{\hat{X}^\nu,\hat{X}^{\nu+1}}(t_{s+1}-t_s) \quad (3.1.7)$$

If we choose $K$ equal to the number of reactions occurring in the time interval $[t_s, t_{s+1}]$, $H_{\hat{X}^\nu,\hat{X}^{\nu+1}}$ can be expressed as follows

$$H_{X^\nu,X^{\nu+1}} = h_{R_\nu}(X^\nu,\boldsymbol{\Theta})\delta_{X^\nu,X^\nu-A_\nu} - h_{R_\nu}(X^{\nu+1},\boldsymbol{\Theta})\delta_{X^\nu,X^{\nu+1}} \quad (3.1.8)$$

where $R_\nu$ is the reaction transforming $X^\nu$ into $X^{\nu+1}$ ($\nu = 0,\ldots,K-1$). For a biological system of realistic size, the number of reactions occurring between two measured state is usually much greater than 1, so that the condition $K \gg 1$ is usually satisfied.

### 3.2. Sampling the BlenX state transition system

Since the system is stochastic $K$ is not constant, but it can change simulation by simulation. $\{R_\nu\}$ with $\nu = 0,\ldots,K-1$ can be considered a latent reaction pathway. To calculate the likelihood we have to find an efficient way to sample this latent reaction pathway conditioned to the observations. Namely, we have to sample the latent reaction processes that match the initial and the end state in the time interval. The parameter estimation is then formulated as maximization of the likelihood function. The parameter estimate $\boldsymbol{\Theta}^*$ is calculated as

$$\boldsymbol{\Theta}^* = \arg\max_{\boldsymbol{\Theta}} L(\hat{X}_j^0, \hat{X}_j^K;\boldsymbol{\Theta}) \quad (3.2.1)$$

and the likelihood function over the entire duration of the observation is the product of the likelihood of each subinterval.

$$L(\mathbf{X}(t);\boldsymbol{\Theta}) = \prod_{j\subseteq\Gamma}\prod_{s=1}^{m} L(X_j(t_s), X_j(t_{s+1});\boldsymbol{\Theta}) \quad (3.2.2)$$

To sample latent path that are consistent with the observations means to generate a Markov chain that match the initial and the end state of the system in the considered time interval. One commonly used sampling method is the stochastic simulation algorithm (SSA). However, SSA is computationally inefficient when the total number of possible state is high. Y. Wang et al (Wang, Christley, Mjolsness, & Xie, 2010) suggested a Markov chain sampler working as follows:

1. generate an initial path
2. generate a set of reaction, by adding or removing reactions front he initial set
3. estimating the acceptance probability of a new set
4. accept or reject a pathway

Note that both the initial path and the processed path have to match the observations at the start and the end of the interval, implying that only a subset of the reactions can be used for either initialization or addition/deletion. In this work, the first path is randomly generated.

After an initial path is generated we can use the *elementary mode analysis* to generate a new sample. In this study we select randomly the first path. An elementary mode of a biochemical network is a set of reactions that does not change the observed number of molecular species. Therefore, an elementary mode is a column vector $\mathbf{q}_k$ of non-negative integers that satisfy the following condition

$$A_\Gamma^T\mathbf{q}_k = 0 \quad (3.2.3)$$

where $A_\Gamma$ is the net effect matrix of the system $\Gamma$.
The set of all independent elementary mode $\{\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_J\}$ is called *null set* of the biochemical reaction system. Provided with a reaction path and a null set, after randomly choosing an elementary mode $\mathbf{q}_k$ from the null set, we can proceed as follows:

- with probability $p_a$ = 0.25 (Wang, Christley, Mjolsness, & Xie, 2010) , add the set of reactions in $\mathbf{q}_k$ with random waiting time of reaction from a uniform distribution within the considered time interval;

- with probability $p_r$= 0.25 (Wang, Christley, Mjolsness, & Xie, 2010), remove one set of randomly selected reactions in $\mathbf{q}_k$ from the current path within the considered time interval;

A new sample have to undergo two additional constraints:

- the number of any reaction type must be positive after the move
- the population numbers for all species have to remain positive throughout the whole process.

If either of the two conditions is violated we set the probability of the new sample path to be zero and reject the new path.

Each time a new pathway is sampled, we determine the acceptance probability of the proposed pathway according to the formula

$$p = \frac{p_{\text{current}}}{p_{\text{previous}}}$$

where $p_{\text{previous}}$ is the probability of the previous pathway, and $p_{\text{current}}$ is the probability of the current proposed pathway. The probability of a pathway is calculated as in the following formula.

$$p_{\text{pathways}} = \prod_{\{j|q_{k,j}\neq 0\}} \frac{k_j n \tau}{\alpha_j \pi}$$

where $n$ is the total number of components in the reaction system, $\tau$ is the time length of the subinterval, $k_j$ is the rate constant of the reaction of type $j$, $\alpha_j = 1$ if the reaction is monomolecular, and $\alpha_j = 2$ if the reaction is bimolecular. $\pi \in (0,1)$ is a uniform deviate, as in Colvin et al. (Colvin, Monine, Faeder, Hlavacek, Von Hoff, & Posner, 2009). A pathways is accepted if $p \geq 0.25$.

## 4. RESULTS

We generated the time series of the components of the system in Figure 3 synthetically by running the BlenX code with the values of parameters reported in the code in previous pages, and then we applied the procedure of parameter inference described in the previous section. In Figure 5, we show the time series taken as an input for the model calibration procedure. In Table 2 we report the results of the inference for the main reactions (i. e. the rate-limiting step reactions) listed in Figure 2.

Good agreement between inferred and expected values has been obtained within the parameter variance estimated around 1.

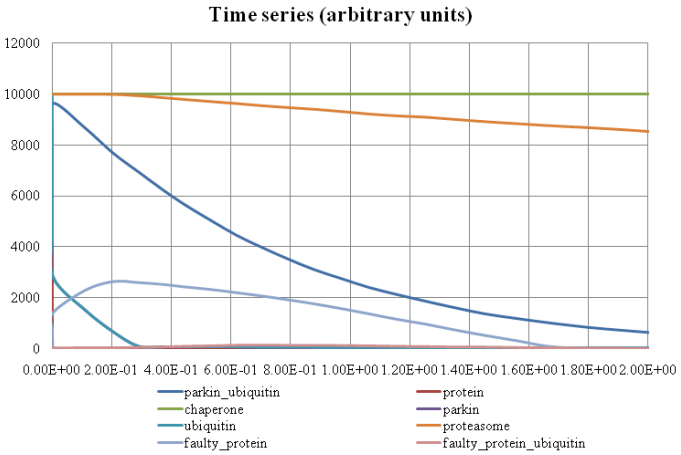


**Figure 5: time-series synthetically generated from the model with given parameters are used as an input to the parameter inference procedure.**

**Table 1: comparison between inferred and expected values of the rate –limiting step reaction in the system of chaperon-assisted protein folding.**

| Parameter | Inferred | Expected |
|---|---|---|
| $k_1$ | 0.03203 | 0.01 |
| $k_2$ | 0.03249 | 0.01 |
| $k_3$ | 0.42681 | 0.5 |
| $k_4$ | 0.03793 | 0.5 |
| $k_5$ | 0.00032 | 0.0002 |

## 5. CONCLUSIONS

We presented a maximum-likelihood method for inferring rate parameters of reactions of a stochastic biochemical systems from discrete time observations. The core of the method has been proposed in 2010 by Wang et al. (Wang, Christley, Mjolsness, & Xie, 2010). In this work we illustrated how it can be adapted to calibrate a process-lagebra model of a biochemical system. We showed that the mathematical approach of the method is suitable to the identification of parameters in language- reaction-based model. In this paper we reported a simple example to give to the reader the flavor both of the BlenX process albegra language and of thr capabilities of the infernce method. From the resslts obtained from synthetic and real case data (not described in this paper) we conclude that this procedure is trustable.

## REFERENCES


*CoSBi*. (2011). Retrieved from http://www.cosbi.eu.

Boys, R. J., Wilkinson, D. J., & Kirkwood, T. L. (2008). Bayesian inference for a discretely observed stochastic kinetic model. *Stat. Comput., 18*, 125-135.

Cardelli, L. (2005). Brane Calculi - Interactions of B iological Membranes. *Workshop on Computatinal Methods in sSystems Biology (CMSB'04) Lecure Notes in Computer Science. 3082*, pp. 257-278. Springer.

Colvin, J., Monine, M. I., Faeder, J. R., Hlavacek, W. S., Von Hoff, D. D., & POsner, R. G. (2009). Simulation of large-scalle rule-nased models. *Bioinfromatics, 25*(7), 910-917.

Danos, V., & Krivine, J. (2004). Reservible communicating systems. *CONCUR 2004. 3170 of LNCS*, pp. 292-307. Springer-Verlag.

Danos, V., & Laneve, C. (2004). Formal molecular biology. *TCS*.

Dematté, L., Priami, C., & Romanel, A. (. (2008). The BetaWorkbench: a computational tool to study the dynamics of biological systems. *9*(5), 437-448.

Dematté, L., Priami, C., & Romanel, A. (. (2008). *The BlenX language: a tutorial*. Trento, Italy: The Microsoft Research - University of Trento centre fro Computational and Systems Biology.

Gillespie, D. T. (1977). Exact stochastic simulation of coupled chemical reactions. *The Journal of Chemical Physics, 81*, 2340–2361.

Gilmore, S., & Hillstone, J. (1994). The PEPA Workbench: A Tool to Support a Process Algebra-based Approach to Performance Modelling. *Lecture Notes in Computer Science, 794*, 353-368.

Lecca, P. (2011). BlenX models of alpha-synuclein and parkin kinetics in neuropathology of Parkinson's disease. *Journal of Biological Systems, 19*(2).

Lecca, P., Palmisano, A., Ihekwaba, A. E., & Priami, C. (2010). Calibration of dynamic models of biological systems with KInfer. *European Journal of Biophysics, 39*(6), 1019.

Priami, C. (1995). Stochastic p-calculus. *The Computer Journal, 38*, 578-589.

Regev, A., Panina, E. M., Silverman, L., Cardelli, L., & Shapiro, E. (2004). Bioambients: an abstraction for biological compartments. *Theor. Comput. Sci., 325*(1), 141-167.

Wang, Y., Christley, S., Mjolsness, E., & Xie, X. (2010). Parameter infernece for discretely observed stochastic kinetic models using gradient descendent. *BMC Systems Biology, 4*(99).


## AUTHORS BIOGRAPHY

Dr. **Paola Lecca** received a Master Degree in Theoretical Physics from the University of Trento (Italy) and a PhD in Computer Science from the International Doctorate School in Information and Communication Technologies at the University of Trento (Italy). Currently Paola Lecca is the Principal Investigator of the Inference and Data manipulation research group at The Microsoft Research – University of Trento Centre for Computational and Systems Biology (Trento,, Italy). Dr. Paola Lecca's research interests include stochastic biochemical kinetic, biological networks inference, optimal experimental design in biochemistry, and computational cell biology. She designed prototypes for biological model calibration and for the simulation of diffusion pathways in cells and tissues. She has published articles in leading medical, biological and bioinformatics Journals and Conferences (http://www.cosbi.eu/).