Step-by-step Robustness for Biochemical Networks

Ruggero Lanotte, Desiree Manicardi, and Simone Tini

University of Insubria, Como, Italy

Abstract. We propose a notion of robustness for biochemical networks that, intuitively, measures the ability of the network to exhibit step-by-step limited variations on the concentration of a species of interest at varying of the initial concentration of other species. We provide a statistical technique allowing for estimating robustness and showcase it on the EnvZ/OmpR Osmoregulatory Signal System of E. coli.

1 Introduction

Living cells are complex systems whose morphological and functional organization have been thoroughly investigated in recent years, in the context of system biology [23]. The complex behavior of cells arises from the interactions of their huge amount of components, that interact with each other, through chemical reaction networks. Malfunctioning or corruption of these interactions may originate in severe diseases, such as cancer or diabetes. Therefore, it is relevant to study how the components of the cells interact with each other as a system in order to be able to predict how *perturbations* can modify their behavior. In particular, in some cases, it is interesting to predict in which case the nominal behavior of the cell can be maintained in presence of those perturbations, which leads to the notion of *robustness*. Notably, the ability to maintain the nominal behavior is not a qualitative property, but can be quantified, to formalize to which extent the original behavior can be maintained in function of the amount of perturbation introduced in the system.

The notion of *robustness* has been widely used in several contexts, from control theory [25] to security [9] to biology [12], and is commonly meant as the ability of a system to maintain its functionalities against external and internal perturbations. Among the notions of robustness introduced in biology, *α-robustness* [15] verifies how by varying the *initial* concentration of some species, called conventionally the *input species*, the concentration of other species of interest, called the *output species*, varies at *steady state*.

In the present paper, we study a notion of robustness that is inspired by that of α-robustness, but with two main differences: (i) we work within the stochastic model [10], whereas α-robustness has been proposed for the deterministic model [2]; (ii) coherently with the choice of the model, instead of evaluating the concentration level of output species at steady state, we evaluate it step-by-step, up to a finite horizon. Being the deterministic and the stochastic approach complementary [2], we can argue that also our results and those in [15] are. Clearly, robustness in our sense captures random effects and temporary effects that are typical of the stochastic model.

1.1 Our contribution

In order to study the robustness of system behavior in the stochastic model, we need: (i) a language allowing us to specify systems of interest; (ii) a semantic model, which must capture the probabilistic behavior that characterizes the stochastic approach; (iii) a formal notion of robustness, defined on the semantic model; (iv) an algorithm allowing us to estimate the robustness of a system starting from its specification.

As regards specification, we rely on *process calculi* (similar to those in [14,13]), adopting in particular the *species as processes* [3] approach. Essentially, a solution with n species is modeled by the parallel composition of n processes, where each process represents one species and its concentration level. In this paper, the concentration level is expressed in terms of the number of molecules, but this can be generalized.

In order to model the behavior, we follow Gillespie's approach [10], where computation steps represent single chemical reactions. Here, at each step there is a competition between all available reactions, giving rise to a probabilistic behavior. In this context, we believe that it is convenient to adopt the semantic model of *evolution sequences* proposed in [4,5]: essentially, an evolution sequence is a sequence of probability measures over the attainable configurations.

By adopting evolution sequences, one can exploit the whole theory developed in [4,5] to measure the *differences* between system behaviors, which supports the study of robustness properties. In [4,5] the focus is on cyber-physical systems and the starting idea is to provide a notion of *distance* between computation states that quantifies to which extent they perform differently with respect to some targets that are fixed initially. In other words, to each computation state one assigns a *penalty* quantifying how bad the system is working with respect to the targets, and the distance between two computation states is given by the difference of the penalties that are assigned to them. Then, the notion of distance between computation states is lifted first to probability measures over computation states and, then, to evolution sequences. In the present paper, we customize this approach to support the study of a robustness property for biochemical networks inspired by the α-robustness of [15]. Essentially, if we are interested in studying the difference between the behaviour of two systems by focusing on the quantity of a given set of species, we can assign to each system state a *rank* that depends on the available quantity of these species. Then, the distance between two states coincides with the difference between their ranks, and this can be lifted to probability measures and to evolution sequences. Since α-robustness bases on how much the output species vary depending on the variation of the initial concentration of input species, we can consider two different ranks, which capture the input and the output species, respectively, and that allow us to define the *input distance* and the *output distance* between the nominal system and the perturbed one. Then we formalize a notion of robustness whose intuition is that small variations on input distance should give rise to smooth and limited variations on output distance.

Then, following [4,5], we provide a randomized algorithm that permits estimating the evolution sequences of systems and thus for the evaluation of the distances between them. This allows us to estimate robustness of a nominal system, by sampling perturbed systems at a fixed maximal input distance from it and by estimating their output distance. In order to validate our proposal, we have provided a Python implementation, available at `https://github.com/dmanicardi/spebnr`, for the *EnvZ/OmpR Osmoregulatory Signaling System* of E. coli.

1.2 Related work

Robustness in biology has been extensively studied in recent years. A very general approach has been proposed in [17,18], where the nominal behaviour of a system subject to perturbations is expressed in terms of a formula specified with a linear temporal logic equipped with a quantitative semantics, then the robustness of the system is quantified as the average satisfaction degree of that property over all admissible perturbations, possibly weighted by their probabilities. Several notions of robustness proposed in the literature are less general and focus on steady state behaviour. As an example, the notion of adaptability in [1] captures the idea that the behaviour at steady state of a system is insensitive with respect to the initial concentration of some species. Absolute concentration robustness [19,20] with respect to a given species requires that the system admits at least one positive steady state and that the concentration of that species is the same in all of the positive steady states that the system might admit. The notion of α-robustness has been proposed in [15] on the Continuous Petri Nets model. Intuitively, this notion captures the idea that by varying the initial concentration of input species under suitable constraints, namely by remaining within a so called interval marking of the net, then at steady state one observes a bounded variation of the concentration of output species, namely that concentration is in a ball of radius α.

1.3 Organization of contents.

We devote Section 2 to the presentation of our model. In Section 3 we discuss the behavioral distances and our notion of robustness. Then, in Section 4 we provide the algorithm, based on statistical inference, for estimating the distances and the robustness. The study of EnvZ/OmpR Osmoregulatory Signaling System is described in Section 5. Finally, Section 6 concludes the paper with a discussion on future research directions.

2 The model

We assume a set $\mathcal{N}$ of *names* for species and a set $\mathcal{R}$ of chemical *reactions*. Then, we use a set of *actions* $\mathcal{A}$ describing how a species can take part to a reaction. In particular, if a is a reaction in $\mathcal{R}$, then the following actions are in $\mathcal{A}$:

– $a?^r$, denoting that the species participates in reaction a as a reactant, consuming r molecules,
– $a!^p$, denoting that the species participates in reaction a as a product, producing p molecules,
– $a?^r!^p$, denoting that the species participates in reaction a as both a reactant and a product, like in the case of enzymatic reactions. Here, r molecules are consumed and p molecules are produced.

Elements in $\mathcal{A}$ will be denoted with α, α', α_1, ... For an action $\alpha \in \mathcal{A}$, the associated reaction $\mathsf{r}(\alpha)$ is defined by $\mathsf{r}(a?^r) = \mathsf{r}(a!^p) = \mathsf{r}(a?^r!^p) = a$. For a set of actions $A \subseteq \mathcal{A}$, we let $\mathsf{r}(A)$ denote $\mathsf{r}(A) = \{\mathsf{r}(\alpha) \mid \alpha \in A\}$.

Definition 1 (System). *The set $\mathcal{S}$ of* systems *over $\mathcal{N}$, $\mathcal{R}$ and $\mathcal{A}$ is defined by*

$$S ::= n[A]_L \ \mid \ S \parallel S$$

where: (i) $n \in \mathcal{N}$, (ii) $A \subseteq \mathcal{A}$, and (iii) $L \in \mathbb{N}$.

Intuitively, the system $n[A]_L$ represents that L molecules of species n are in the mixture and have a potential behaviour described by A, and $\parallel$ is the (commutative and associative) parallel composition operator, which allows us to model that several species coexist in the same mixture.

Given systems $n_i[A_i]_{L_i}$, for $i = 1, \ldots, k$, we will say that the system $n_1[A_1]_{L_1} \parallel \cdots \parallel n_k[A_k]_{L_k}$ is *well formed* if for all $1 \leq i < j \leq k$ it holds that $n_i \neq n_j$. We will always assume to work with well formed systems. Intuitively, well formedness ensures that each species is represented by precisely one parallel component.

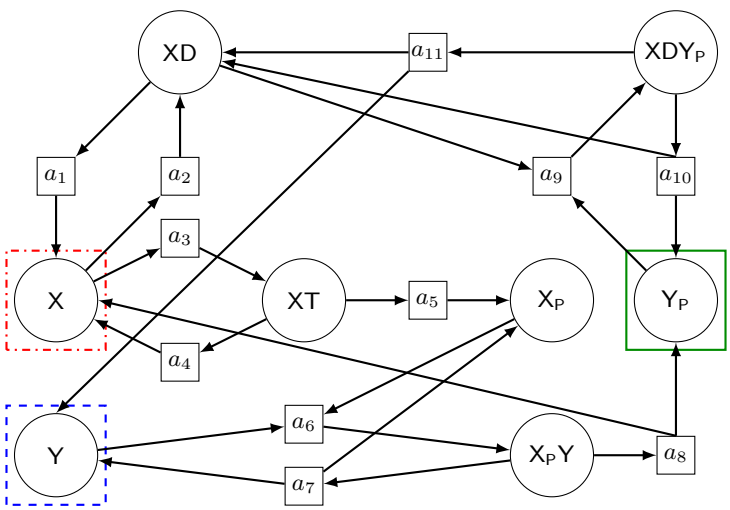


Fig. 1: The Continuous Petri Net model for EnvZ/OmpR [15]

Example 1. Escherichia Coli is a bacterium whose cell contains a few million proteins of different types [23]. The *EnvZ/OmpR Osmoregulatory Signaling System* regulates two of those proteins. In order to support our exposition, in Figure 1 we report the graphical representation of this regulatory network as given in [15] by exploiting the formalism of Continuous Petri Nets. The main components of this chemical network are EnvZ (histine kinase) and OmpR (response regulator), denoted, respectively, as X and Y. EnvZ phosphorylates OmpR ($\mathsf{Y_P}$ in the figure) and itself ($\mathsf{X_P}$), by binding and breaking down ATP. In the picture we can see that this system is characterized by eight species, represented by places in the net, and eleven reactions, represented by transitions. More in detail, the reactants and the products of a reaction are those that are represented by the places that are the sources and the targets, respectively, of the transition represented by that reaction. For instance, the species XT is a reactant for the reactions a_4 and a_5, and is a product for a_3. In this example, each reaction uses precisely one molecule for each of its reactants and produces precisely one molecule for each of its products. Below we give the system S that represents the regulatory network with

species X, Y, XD and $\mathsf{Y_P}$ having a number of molecules corresponding to 25, 150, 50 and 10, respectively, and the other species having no molecule.

$$S = \mathsf{X}[\{a_2?^1, a_3?^1, a_1!^1, a_4!^1, a_8!^1\}]_{25} \;\parallel\; \mathsf{Y}[\{a_6?^1, a_7!^1, a_{11}!^1\}]_{150} \;\parallel\; \mathsf{XD}[\{a_1?^1, a_9?^1, a_2!^1, a_{10}!^1, a_{11}!^1\}]_{50}$$
$$\mathsf{XT}[\{a_3!^1\}]_0 \;\parallel\; \mathsf{X_P}[\{a_5!^1, a_7!^1\}]_0 \;\parallel\; \mathsf{X_P Y}[\{a_6!^1\}]_0 \;\parallel\; \mathsf{XDY_P}[\{a_9!^1\}]_0 \;\parallel\; \mathsf{Y_P}[\{a_9?^1, a_8!^1, a_{10}!^1\}]_{10} \tag{1}$$

2.1 Behavioral model

We define the system's behaviour by means of two types of transitions. Those of the first type represent the potential behaviour of systems. They are of the form $S \xrightarrow{(a,w)} S'$, which models that the reaction a can take S to S'. Here, w is the *weight* of the transition, and is a real number that will allow us to calculate the rate of the reaction a, which is needed in order to calculate the probability of all enabled reactions. The transitions of the second type describe how all enabled reactions for a system S compete and take it to a probability distribution over systems. They are of the form $S \Longrightarrow \pi$, where π is a discrete distribution over systems, namely a mapping $\pi : \mathcal{S} \to [0,1]$ with $\sum_{S \in \mathcal{S}} \pi(S) = 1$. From the transitions of the form $S \Longrightarrow \pi$ we will derive transitions of the form $\pi \Longrightarrow \pi'$ that model the evolution of distributions of system.

$$\frac{a?^r \in A \quad L \geq r}{n[A]_L \xrightarrow{(a,\binom{L}{r})} n[A]_{L-r}} \qquad \frac{a!^p \in A}{n[A]_L \xrightarrow{(a,1)} n[A]_{L+p}} \qquad \frac{a?^r!^p \in A \quad L \geq r}{n[A]_L \xrightarrow{(a,\binom{L}{r})} n[A]_{L-r+p}}$$

$$\frac{a \notin \mathsf{r}(A)}{n[A]_L \xrightarrow{(a,\#)} n[A]_L} \qquad \frac{S_1 \xrightarrow{(a,w_1)} S_1' \quad S_2 \xrightarrow{(a,w_2)} S_2'}{S_1 \parallel S_2 \xrightarrow{(a,w_1 \cdot w_2)} S_1' \parallel S_2'}$$

Table 1: Potential behaviour of systems

The transitions of the first type are derived by means of the inference rules in Table 1. The first rule represents that r molecules of species n are consumed by taking part to reaction a as reactant. The weight $\binom{L}{r}$ coincides with the number of ways r molecules of n can be taken out from the available L, namely it is the number of sets of molecules of n that can take part to the reaction a. The second rule represents that p molecules of n are produced by taking part to reaction a as product. The weight 1 reflects that the number of molecules of the species does not impact on the rate of the reaction. The third rule represents that r molecules of n are consumed and p molecules of n are produced by taking part to reaction a both as reactant and as product. The fourth rule is applied when n is not involved in reaction a. The inferred transition allows for composing the behaviour of $n[A]_L$ with that of other species. The last rule allows for combining the behaviour of systems running in parallel. Clearly, they have to agree upon the reaction to be taken and the resulting weight is the product of the weights of the composed transitions. Here, we extend classic product by $\# \cdot w = w \cdot \# = w$ for all reals w, and $\# \cdot \# = \#$.

For a system S we denote by $\mathsf{trgt}(S)$ the set of the triples $\{(a_i, w_i, S_i) \mid S \xrightarrow{(a_i,w_i)} S_i \text{ and } w_i \neq \#\}$. Notice that for $S = n_1[A_1]_{L_1} \parallel \cdots \parallel n_k[A_k]_{L_k}$, we have $(a_i, w_i, S_i) \in \mathsf{trgt}(S)$ with $w_i > 1$ if and only if S contains w_i different sets of reagents that can take part to reaction a_i

$$\frac{\mathsf{trgt}(S) = \{(a_i, w_i, S_i) \mid i \in I\}}{S \Longrightarrow \sum_{i \in I} \frac{(c_{a_i} \cdot w_i)}{\sum_{j \in I}(c_{a_j} \cdot w_j)} \delta(S_i)} \qquad \frac{S_i \Longrightarrow \pi_i}{\sum_{i \in I} p_i \delta(S_i) \Longrightarrow \sum_{i \in I} p_i \pi_i}$$

Table 2: Probabilistic behaviour of systems

In order to define the transitions of the form $S \Longrightarrow \pi$, we need some notation for distributions over systems. By $\delta(S)$ we denote the *point distribution* giving probability 1 to S and probability 0 to all S' different from S. Then, for a countable set of indexes I and distributions π_i with $i \in I$, the distribution $\sum_{i\in I} p_i\pi_i$ with all $p_i \geq 0$ and $\sum_{i\in I} p_i = 1$ is the convex distribution defined by $(\sum_{i\in I} p_i\pi_i)(S) = \sum_{i\in I} p_i \cdot \pi_i(S)$.

The first rule in Table 2 represents the competition between all enabled reactions of S. For each reaction a, we consider the constant reaction c_a associated with the reaction, where, intuitively, $c_a\, dt$ is the probability that a particular combination of reactants gives rise to reaction a in an infinitesimal time interval dt. The probability that S behaves as described by (a_i, w_i, S_i) is the ratio between the rate $c_{a_i} \cdot w_i$ of the reaction a_i and the sum of the rates of all reactions $\sum_{j\in I}(c_{a_j} \cdot w_j)$. To explain this point, we recall that $c_{a_i} \cdot w_i$ is the parameter of the exponential distribution modeling the time elapsing between two consecutive occurrences of reaction a_i and $\sum_{j\in I}(c_{a_j} \cdot w_j)$ is the parameter of the exponential distribution modeling the time elapsing between two consecutive occurrences of arbitrary reactions. Summarizing, S reaches the convex distribution of systems $\sum_{i\in I} p_i\pi_i$ with $p_i = \frac{(c_{a_i}\cdot w_i)}{\sum_{j\in I}(c_{a_j}\cdot w_j)}$ and $\pi_i = \delta(S_i)$.

The second rule in Table 2 lifts the behaviour of systems to that of distributions over systems. We note that if there is a sequence of transitions $\pi_1 \Longrightarrow \pi_2 \Longrightarrow \ldots \pi_i \ldots$, for π_1 the point distribution $\delta(n_1[A_1]_{L_1} \parallel \cdots \parallel n_m[A_m]_{L_m})$, then all systems S in the support of any π_i are of the form $S = n_1[A_1]_{L'_1} \parallel \cdots \parallel n_m[A_m]_{L'_m}$, for suitable $L'_1, \ldots, L'_m$. Following [4,5], the sequence $\pi_1 \Longrightarrow \pi_2 \Longrightarrow \ldots \pi_i \ldots$ is called an *evolution sequence*. In particular, it is the evolution sequence of S if $\pi_1 = \delta(S)$.

In a transition $S \Longrightarrow \pi$ the probability weight assigned to each element in the support of π depends on the rate of all reactions that are possible in S. This is no more true in a transition $\pi \Longrightarrow \pi'$ since the rates of the chemical reactions from two different systems in the support of π are unrelated.

2.2 Remarks on the semantic model

We have described the behaviour of a system by means of an evolution sequence, namely a sequence of distributions of systems. Clearly, computing exactly these distributions is undoable. However, by adapting to our context the work in [4,5,6], we can easily obtain a randomized algorithm allowing us to estimate the evolution sequence of a system S. This randomized algorithm, detailed in Section 4, works as follows: (i) we apply N times a procedure allowing us to compute a h-steps trajectory. Essentially, this coincides with applying N times the classical Gillespie algorithm [10]. For $i = 1, \ldots, N$, we obtain the trajectory $S^i_0, S^i_1, \ldots, S^i_h$, where all S^i_0 coincide with S. Clearly, for all $j = 1, \ldots, h$ it holds that the samples $S^1_j, \ldots, S^N_j$ obtained at step j are independent and identically distributed; (ii) for each $j = 1, \ldots, h$, we use $S^1_j, \ldots, S^N_j$ to derive the empirical distribution $\hat{\pi}_j$ defined simply by $\hat{\pi}_j(S') = \frac{|\{i|S^i_j = S' \text{ and } 1\leq i\leq N\}|}{N}$; (iii) by applying the weak law of large numbers to i.i.d. samples we infer that when N goes to infinite then $\hat{\pi}_j$ converges weakly to π_j, where π_j are the distributions such that $\delta(S) \Longrightarrow \pi_1 \Longrightarrow \pi_2 \ldots$.

3 Behavioral distances over systems

In [4,5,6] a notion of *behavioral distance* over cyber-physical systems was proposed, aiming to express how well the cyber components fulfill their tasks. In this section, we customize that theory for our model, clearly with a different target. In particular, given two systems that differ by the number of molecules of species, say $S_1 = n_1[A_1]_{L^1_1} \parallel \cdots \parallel n_k[A_k]_{L^1_k}$ and $S_2 = n_1[A_1]_{L^2_1} \parallel \cdots \parallel n_k[A_k]_{L^2_k}$, we aim to *quantify the differences over their evolution sequences* $S_1 \Longrightarrow \pi^1_1 \Longrightarrow \pi^1_2 \ldots$ and $S_2 \Longrightarrow \pi^2_1 \Longrightarrow \pi^2_2 \ldots$. To this purpose, we proceed as follows: (i) first we introduce the concept of distance between systems, which deals with the difference between the number of molecules of the species; (ii) then we lift this notion to a notion of distance between distributions, so that we can quantify the distance between the distributions π^1_i and π^2_i that are reached by S_1 and S_2, respectively, at step i, and, finally, (iii) we lift this distance to a distance between the two evolution sequences.

We start with a notion of distance between systems that focuses on a single specie. We assume to know the minimum and maximum level $\min(n)$ and $\max(n)$ that a given species n may reach during the computation. Notice that these values can be estimated by applying our randomized algorithm quickly described above in Section 2.2 and detailed below in Section 4.

Definition 2 (n_i-distance over systems). *Assume systems* $S_1 = n_1[A_1]_{L_1^1} \parallel \cdots \parallel n_k[A_k]_{L_k^1}$ *and* $S_2 = n_1[A_1]_{L_1^2} \parallel \cdots \parallel n_k[A_k]_{L_k^2}$. *Then the* distance between S_1 and S_2 with respect to species n_i, *or* n_i*-distance between* S_1 *and* S_2 *for short, is* $\mathbf{d}_{n_i}(S_1, S_2) = \frac{|L_i^1 - L_i^2|}{\max(n_i) - \min(n_i)}$.

Clearly, $\mathbf{d}_{n_i}(S_1, S_2)$ is a value in the interval $[0, 1]$. Moreover, $\mathbf{d}_{n_i}$ is a *1-bounded pseudometric*, namely a function $\mathbf{d}_{n_i} : \mathcal{S} \times \mathcal{S} \to [0, 1]$ such that for all systems S_1, S_2, S_3 the following properties hold: (i) $\mathbf{d}_{n_i}(S_1, S_1) = 0$, (ii) $\mathbf{d}_{n_i}(S_1, S_2) = \mathbf{d}_{n_i}(S_2, S_1)$, and (iii) $\mathbf{d}_{n_i}(S_1, S_3) \leq \mathbf{d}_{n_i}(S_1, S_2) + \mathbf{d}_{n_i}(S_2, S_3)$. Notice that pseudometric $\mathbf{d}_{n_i}$ is not a *metric*, since, in general, it does not hold that $\mathbf{d}_{n_i}(S_1, S_2) = 0$ implies $S_1 = S_2$.

The second step to obtain the evolution distance consists in lifting distances on systems to distances on distributions over systems. Among the several notions of lifting for pseudometric proposed in the literature (see [16] for a survey), following [5] we opt for that known as *Wasserstein distance* or *Kantorovich-Rubinstein pseudometric*, since it preserves the properties of the ground pseudometric and is computationally tractable via statistical inference.

In order to recall formally that notion, we need to introduce the notion of *matching*, also known as *measure coupling* [24], for pairs distributions.

Definition 3 (Matching, [24]). *Assume a set* X. *A* matching *for a pair of distributions* (π, π') *over* X *is a distribution* ω *over the product state space* $X \times X$ *with left marginal* π, *i.e.* $\sum_{x' \in X} \omega(x, x') = \pi(x)$ *for all* $x \in X$, *and right marginal* π', *i.e.* $\sum_{x \in X} \omega(x, x') = \pi'(x')$ *for all* $x' \in X$. *We let* $\Omega(\pi, \pi')$ *denote the set of all matchings for* (π, π').

According to [24], a matching in $\Omega(\pi, \pi')$ may be understood as a transportation schedule for the shipment of probability mass from π to π'. Then, by exploiting that notion, we can introduce the Kantorovich lifting.

Definition 4 (Kantorovich metric, [11]). *The* Kantorovich lifting *of a pseudometric* d *over a set* X *is the pseudometric* $\mathbf{K}(d)$ *over distributions over* X *defined for all distributions* π, π' *by*

$$\mathbf{K}(d)(\pi, \pi') = \min_{\omega \in \Omega(\pi, \pi')} \sum_{x, x' \in X} \omega(x, x') \cdot d(t, t').$$

The reader familiar with probability theory, might have recognized the Kantorovich lifting $\mathbf{K}(d)(\pi, \pi')$ as the minimum expected value of the ground distance d, over the supports of π and π', with respect to the distribution ω. According to [24], $\mathbf{K}(d)(\pi, \pi')$ gives the optimal cost for shipping probability mass from π to π' when $d(x, x')$ is the unit cost for shipping mass from x to x'.

We notice that pseudometric $\mathbf{K}(\mathbf{d}_{n_i})$ preserves the 1-boundedness property of $\mathbf{d}_{n_i}$. From the distances with respect to all species $\mathbf{K}(\mathbf{d}_{n_1}), \ldots, \mathbf{K}(\mathbf{d}_{n_k})$, we can derive a unique notion of distance by exploiting weights that quantify the *relevance* that we want to assign to each specie.

Definition 5 (Distance between system distributions). *Assume* weights $w_1, \ldots, w_k \geq 0$ *such that* $\sum_{j=1}^{k} w_j = 1$. *The distance* $\mathbf{dd}$ *between system distributions induced by distances* $\mathbf{d}_{n_1}, \ldots, \mathbf{d}_{n_k}$ *and weights* $w_1, \ldots, w_k$ *is defined by* $\mathbf{dd}(\pi, \pi') = \sum_{i=1}^{k} w_i \cdot \mathbf{K}(\mathbf{d}_{n_i})(\pi, \pi')$.

Clearly, being a convex combination of 1-bounded pseudometrics, the function $\mathbf{dd}$ is a 1-bounded pseudometric. We argue that $\mathbf{dd}$ can be computed efficiently, by following [22]. In detail, computing $\mathbf{dd}(\pi, \pi')$ requires computing the Kantorovich liftings $\mathbf{K}(\mathbf{d}_{n_1})(\pi, \pi'), \ldots, \mathbf{K}(\mathbf{d}_{n_k})(\pi, \pi')$. Both π and π' are discrete distributions, let us assume that their cardinality is N. If π and π' are estimated by our randomized algorithm, then N will be a parameter of the algorithm. In computing each lifting $\mathbf{K}(\mathbf{d}_{n_i})(\pi, \pi')$, each system in the support of π and π' can be viewed as a real, obtained as the cardinality of the molecules of n_i divided by $\max(n_i) - \min(n_i)$. Therefore, π and π' can be viewed as multisets of reals of cardinality N. This property is exploited in [22] as follows. In $O(N \log N)$ these two multisets can be ordered. Let $u_1 \leq \cdots \leq u_N$ and $v_1 \leq \cdots \leq v_N$ be the ordered sequences. Notice that the ground distance $\mathbf{d}_{n_i}$ between two reals in the support of π and π' is a real (the absolute value of their difference, namely $\mathbf{d}_{n_i}(u_i, v_j) = |u_i - v_j|$). Following [22], we have that $\mathbf{K}(\mathbf{d}_{n_i})(\pi, \pi') = \sum_{i=1}^{N} \mathbf{d}_{n_i}(u_i, v_j)$. Therefore, computing $\mathbf{dd}$ can be done in $O(k \cdot N \log N)$.

We now need to lift $\mathbf{dd}$ to a distance on evolution sequences. To this end, we observe that the evolution sequence of a system includes the distributions over systems induced after each computation step. Following [5] we define the evolution distance as a sort of *weighted infinity norm* of the tuple of the Kantorovich distances between the distributions in the evolution sequences.

Definition 6. *[Evolution distance] For a pseudometric* $\mathbf{dd}$ *over distributions over* $\mathcal{S}$ *and an interval* $[\tau_1, \tau_2]$*, the* evolution distance *over* $[\tau_1, \tau_2]$ *is the mapping* $\mathbf{E}(\mathbf{dd})_{[\tau_1,\tau_2]}$ *such that, given systems* S_1 *and* S_2 *and their evolution sequences* $S_1 \Longrightarrow \pi_1^1 \Longrightarrow \pi_2^1 \Longrightarrow \ldots \pi_i^1 \ldots$ *and* $S_2 \Longrightarrow \pi_1^2 \Longrightarrow \pi_2^2 \Longrightarrow \ldots \pi_i^2 \ldots$ *we have*

$$\mathbf{E}(\mathbf{dd})_{[\tau_1,\tau_2]}(S_1, S_2) = \max_{\tau_1 \leq i \leq \tau_2} \mathbf{dd}(\pi_i^1, \pi_i^2)$$

Since $\mathbf{dd}$ is a 1-bounded pseudometric, we can easily derive the same property for $\mathbf{E}(\mathbf{dd})_{[\tau_1,\tau_2]}$.

We remark that, as argued in [6], the choice of defining the evolution metric as the pointwise maximal distance in time between the evolution sequences of systems is natural and reasonable when one aims to evaluate the highest distance on the behaviour of systems. However, in different contexts it would be more appropriate to use a different *aggregation function* over the tuple of Kantorovich distances instead of max. The definition of $\mathbf{E}(\mathbf{dd})_{[\tau_1,\tau_2]}$ in Definition 6 could be replaced by a definition parametric with respect to aggregation functions. In order to prevent a too heavy notation, we decided to opt for the present formulation.

3.1 Robustness

Technically, in order to formalize the notion of robustness we need two distances, one taking into account input species and the other taking into account the output species. These will be the input distance $\mathbf{dd}_i$ and the output distance $\mathbf{dd}_o$ over distributions, and will be defined as in Definition 5 by giving positive weights to only input and only output species, respectively. The definition of robustness is parametric with respect to these two distances, two thresholds η_1 and η_2 and two intervals I_1 and I_2: a system S is robust with respect to these parameters if whenever we consider a system S' whose input distance from S in interval I_1 remains below η_1, then the output distance observed in interval I_2 remains below η_2.

Definition 7. *A system* S *is* $(\mathbf{dd}_i, \mathbf{dd}_o, I_1, I_2, \eta_1, \eta_2)$*-robust if for all systems* S'*:*

$$\mathbf{E}(\mathbf{dd}_i)_{I_1}(S, S') \leq \eta_1 \textit{ implies } \mathbf{E}(\mathbf{dd}_o)_{I_2}(S, S') \leq \eta_2$$

Clearly, we will use $I_1 = [0, 0]$ if the input distance is relevant only in the initial state of systems and we use $I_2 = [\tau_1, h]$ if we want to observe the behaviour of systems up to a time horizon h.

4 Estimating the evolution distance

In this section we follow [4,5,6] and propose a technique to estimate the evolution distance via statistical techniques. First, we show how we can estimate the evolution sequence of a given system S. The idea was quickly anticipated in Section 2.2. Then, we evaluate the distance between two systems S_1 and S_2 on their estimated evolution sequences.

Given a system S_0 and an integer h we can use the function $\textsc{Simulate}(S_0, h)$, defined in [6] and adapted to our purpose in Figure 2, to sample a sequence of systems of the form $seq = (S_0, S_1, \ldots, S_h)$ that represents h-steps of a computation starting from S_0. Each step of the sequence is computed by using function $\textsc{SimStep}$, also defined in Figure 2. Here, we assume that $\mathcal{R} = \{a_1, \ldots, a_{|\mathcal{R}|}\}$ and we let $\textsc{rand}$ be a function that allows us to get a uniform random number in $(0, 1]$. Essentially, $\textsc{SimStep}$ computes one step of Gillespie algorithm. Our version of $\textsc{SimStep}$ is significantly different with respect to that provided in [6], which was proposed for cyber physical systems and had to deal with the interactions between the cyber and the physical components of the systems. Our version of $\textsc{Simulate}$ returns also the minimum and maximum level of each species n in all generated systems. In computing the arrays m and M with the minimal and maximal levels of species we exploit the function proj_i mapping a system to the level of species n_i, namely $\mathsf{proj}_i(n_1[A_1]_{L_1} \parallel \cdots \parallel n_k[A_k]_{L_k}) = L_i$.

To compute the empirical evolution sequence of a system S the function $\textsc{Estimate}$ proposed in [6] and adapted to our purposes in Figure 3 can be used. The function $\textsc{Estimate}(S, h, N)$ invokes N times the function $\textsc{Simulate}$ in Figure 2 in order to sample N sequences of systems $(S_0^i, \ldots, S_h^i)$, for $i = 1, \ldots, N$, each modeling h steps of a computation from $S = S_0$. Thus, a sequence of tuples of samples $\{E_0, \ldots, E_h\}$ is computed, where each E_j is the tuple $(S_j^1, \ldots, S_j^N)$ of systems observed at time j in each of the N sampled computations. Notice that, for each $j \in \{0, \ldots, h\}$, the samples $S_j^1, \ldots, S_j^N$ are independent and identically

```
function SIMULATE(n_1[A_1]_{L_1} ‖ ··· ‖ n_k[A_k]_{L_k}, h)
    step ← 1
    m ← [0, ..., 0]
    M ← [0, ..., 0]
    S ← n_1[A_1]_{L_1} ‖ ··· ‖ n_k[A_k]_{L_k}
    seq ← (S)
    while step ≤ h do
        S ← SIMSTEP(S)
        seq.append(S)
        for i = 1, ..., k do
            m(i) = min(m(i), proj_i(S))
            M(i) = max(M(i), proj_i(S))
        end for
        step ← step + 1
    end while
    return (seq, m, M)
end function
```

```
function SIMSTEP(n_1[A_1]_{L_1} ‖ ··· ‖ n_k[A_k]_{L_k})
    for j = 1, ..., |R| do
        for i = 1 ... k do
            w_{j,i} ← { C(L_i, r)   if a_j?^r ∈ A_i or a_j?^r!^p ∈ A_i
                      { 1           if a_j!^p ∈ A_i
                      { #           otherwise
        end for
        w_j = c_{a_j} · ∏_{i=1}^{k} w_{j,i}
    end for
    for j = 1, ..., |R| do
        p_j = { w_j / Σ_{j=1}^{|R|} w_j   if w_j ≠ #
              { 0 otherwise
    end for
    u ← RAND()
    let j s.t. Σ_{l=1}^{j-1} p_l < u ≤ Σ_{l=1}^{j} p_l
    for i = 1, ..., k do
        L'_i ← L_i + { −r if a_j?^r ∈ A_i
                     { −r + p if a_j?^r!^p ∈ A_i
                     { +p if a_j!^p ∈ A_i
                     { 0 otherwise
    end for
    return n_1[A_1]_{L'_1} ‖ ··· ‖ n_k[A_k]_{L'_k}
end function
```

Fig. 2: Functions used to simulate the behavior of a system S. The set $\mathcal{R}$ is $\mathcal{R} = \{a_1, \ldots, a_{|\mathcal{R}|}\}$.

distributed. Each E_j can be used to estimate the distribution $\pi_{S,j}$, namely the jth element of the evolution sequence $\delta(S) = \pi_{S,0} \Longrightarrow \pi_{S,1} \Longrightarrow \ldots \Longrightarrow \pi_{S,k}$. For any j, with $0 \leq j \leq h$, we let $\hat{\pi}^N_{S,j}$ be the distribution such that for any system S' we have

$$\hat{\pi}^N_{S,j}(S') = \frac{|E_j \cap \{S'\}|}{N}.$$

Finally, we call $\hat{\pi}^N_S = \hat{\pi}^N_{S,0} \ldots \hat{\pi}^N_{S,h}$ as the *empirical evolution sequence*. Notice that by applying the weak law of large numbers to the i.i.d samples, we get that when N goes to infinite $\hat{\pi}^N_{S,j}$ converge weakly to $\pi_{S,j}$.

The algorithms in Figures 2 and 3 have been implemented in Python and are available at https://github.com/dmanicardi/spebnr. This implementation can be viewed as a customisation of the tool SPEAR (https://github.com/quasylab/spear) implementing the algorithms in [6].

We have seen that function ESTIMATE allows us to collect independent samples at each time step from 0 to a deadline h. We proceed with showing how these samples can be used to estimate the distance between two systems S_1 and S_2.

Following [22], to estimate the Kantorovich lifting $\mathbf{K}(d)$ of a distance d between two (unknown) distributions over reals π_1 and π_2, one can use N independent samples $\{e^1_1, \ldots, e^N_1\}$ taken from π_1 and $\ell \cdot N$ independent samples $\{e^1_2, \ldots, e^{\ell \cdot N}_2\}$ taken from π_2, with ℓ a natural suitable chosen. After that, one orders $\{e^1_1, \ldots, e^N_1\}$ and $\{e^1_2, \ldots, e^{\ell \cdot N}_2\}$, thus obtaining the two sequences of values $u_1 \leq \cdots \leq u_N$ and $v_1 \leq \ldots 1 \leq v_{\ell \cdot N}$, respectively. The value $\mathbf{K}(d)(\pi_1, \pi_2)$ can be approximated as $\frac{1}{\ell N} \sum_{z=1}^{\ell N} \max\{v_z - u_{\lceil \frac{z}{\ell} \rceil}, 0\}$.

In our case, since we need to estimate the Kantorovich lifting $\mathbf{K}(\mathbf{d}_{n_i})$ for species n_i between the distributions over systems $\pi_{S_1,j}$ and $\pi_{S_2,j}$ that are reached in j steps from S_1 and S_2, respectively, we have to proceed as follows. We obtain the set of reals $\{e^1_1, \ldots, e^N_1\}$ by extracting N samples from $\pi_{S_1,j}$, and, then, by taking from each sample the level of species n_i and by dividing it by $\max(n_i) - \min(n_i)$. Analogously, we obtain the reals $\{e^1_2, \ldots, e^{\ell \cdot N}_2\}$ by extracting $\ell \cdot N$ samples from $\pi_{S_2,j}$, and, then, by taking from each sample the level of species n_i and by dividing it by $\max(n_i) - \min(n_i)$.

Clearly, having the estimation of $\mathbf{K}(\mathbf{d}_{n_i})(\pi_{S_1,j}, \pi_{S_2,j})$ for all species, we can derive an estimation for $\mathbf{dd}(\pi_{S_1,j}, \pi_{S_2,j})$ and, then, for $\mathbf{E}(\mathbf{dd})_{[a,b]}(S_1, S_2)$. The whole procedure is realized by functions DISTANCE and COMPUTEH in Figure 4, available at https://github.com/dmanicardi/spebnr. The former takes as

```
function ESTIMATE(S, h, N)
    m = [0, ..., 0]
    M = [0, ..., 0]
    ∀j : (0 ≤ j ≤ h) : E_j ← ∅
    counter ← 0
    while counter < N do
        ((S_0, ..., S_h), m, M) ← SIMULATE(S, h)
        ∀j : (0 ≤ j ≤ h) : E_j ← E_j ∪ {S_j}
        ∀l : (0 ≤ l ≤ k) : m(l) = min(m(l), m(l))
        ∀l : (0 ≤ l ≤ k) : M(l) = max(M(l), M(l))
        counter ← counter + 1
    end while
    return ({E_0, ..., E_h}, m, M)
end function
```

Fig. 3: Function used to obtain N samples of the evolution sequences of a configuration.

```
function DISTANCE(S_1, S_2, W, h, N, ℓ, a, b)
    ({E_{1,0}, ..., E_{1,h}}, m_1, M_1) ← ESTIMATE(S_1, h, N)
    ({E_{2,0}, ..., E_{2,h}}, m_2, M_2)) ← ESTIMATE(S_2, h, ℓN)
    ∀l : (l = 1, ..., k) : m(l) = min(m_1(l), m_2(l))
    ∀l : (l = 1, ..., k) : M(l) = max(M_1(l), M_2(l))
    dist ← 0
    for all step ∈ [a, b] do
        for all w_i ≠ 0 in W do
            ρ = proj_i/(M(i) − m(i))
            dist_{step,i} ← COMPUTEH(E_{1,step}, E_{2,step}, ρ)
        end for
        dist_step = Σ_{i|w_i≠0} w_i · dist_{step,i}
        dist ← max{dist, dist_step}
    end for
    return dist
end function
```

```
function COMPUTEH(E_1, E_2, ρ)
    (S_1^1, ..., S_1^N) ← E_1
    (S_2^1, ..., S_2^{ℓN}) ← E_2
    ∀j : (1 ≤ j ≤ N) : u_j ← ρ(S_1^j)
    ∀z : (1 ≤ z ≤ ℓN) : v_z ← ρ(S_2^z)
    re index {u_j} s.t. u_j ≤ u_{j+1}
    re index {v_z} s.t. v_z ≤ v_{z+1}
    return (1/ℓN) Σ_{z=1}^{ℓN} max{v_z − u_{⌈z/ℓ⌉}, 0}
end function
```

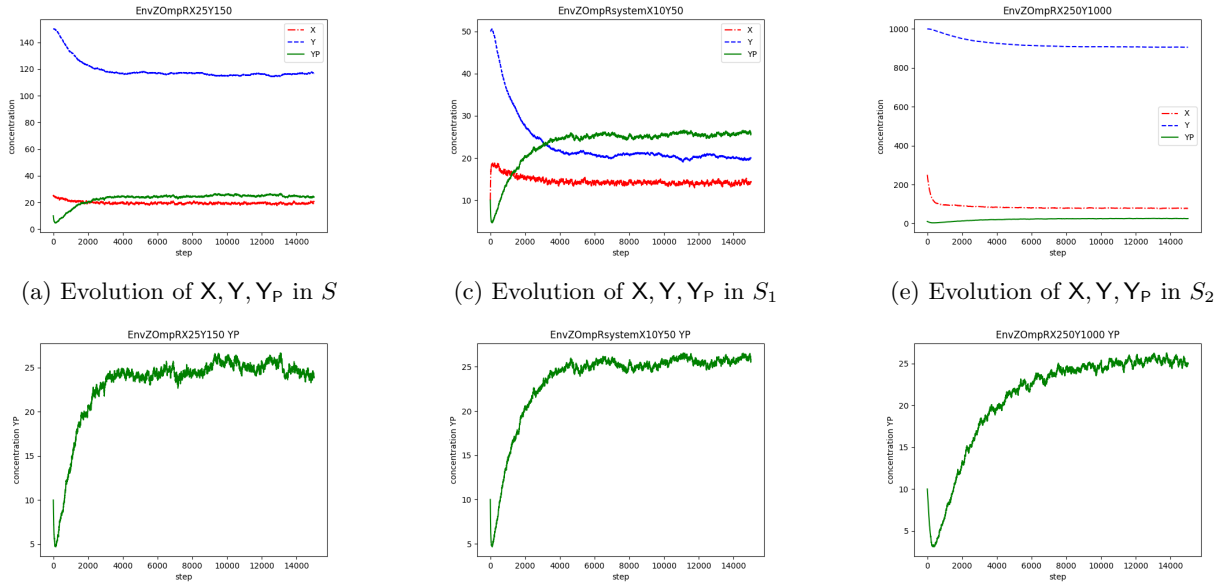
Fig. 4: Functions used to estimate the evolution metric on systems.

input the two systems S_1 and S_2 to compare, the weights assigned to species $W = (w_1, \ldots, w_k)$, the parameter h giving the number of computation steps that are observed, the parameters N and ℓ used to obtain the samplings of computation, a and b. Function DISTANCE collects the samples $E_{1,0}, \ldots, E_{1,h}$ and $E_{2,0}, \ldots, E_{2,h}$ of possible computations of length h from S_1 and S_2. Then, for each $step \in [a, b]$, the n_i distance at time $step$ is computed via the function $\textsc{ComputeH}(E_{1,step}, E_{2,step}, \mathsf{proj}_i/(\mathsf{M}(i) - \mathsf{m}(i)))$, where $\mathsf{m}(i)$ and $\mathsf{M}(i)$ are the minimum and maximum level of species n_i in all systems that are generated.

Due to the sorting of $\{\nu_h \mid h \in [1, \ldots, \ell N]\}$ the complexity of function COMPUTEH is $O(\ell N \log(\ell N))$ (cf. [22]). We refer the interested reader to [21, Corollary 3.5, Equation (3.10)] for an estimation of the approximation error given by the evaluation of the Kantorovich distance over N samples.

5 The EnvZ/OmpR Osmoregulatory Signaling System in E. Coli

In this section, we apply the theory presented in previous sections to the EnvZ/OmpR Osmoregulatory Signaling System in Escherichia Coli introduced in Example 1. This system was already analyzed in [15], within the deterministic model. As reported in [15], an interesting question is whether the initial concentrations of input species EnvZ and OmpR, which are represented in Figure 1 by X and Y, respectively, impact on the long-run concentration of output species phosphorylated OmpR, represented by $\mathsf{Y_P}$. We can answer to that question by studying the robustness of systems with respect to an input distance $\mathbf{dd}_i$ and an output distance $\mathbf{dd}_o$ defined so that they capture the differences over X and Y, and over $\mathsf{Y_P}$, respectively.



(a) Evolution of $\mathsf{X}, \mathsf{Y}, \mathsf{Y_P}$ in S

(c) Evolution of $\mathsf{X}, \mathsf{Y}, \mathsf{Y_P}$ in S_1

(e) Evolution of $\mathsf{X}, \mathsf{Y}, \mathsf{Y_P}$ in S_2

(b) Evolution of $\mathsf{Y_P}$ in S

(d) Evolution of $\mathsf{Y_P}$ in S_1

(f) Evolution of $\mathsf{Y_P}$ in S_2

Fig. 5: Simulation of system S and its perturbed versions S_1 and S_2

The analysis in [15] concluded that the concentration of output species *at steady state* does not depend on the initial value of input species. In other words, at steady state an original system and its perturbed versions obtained by varying the initial concentration of input species X and Y do not exhibit any difference on the concentration of output species $\mathsf{Y_P}$. Our analysis will allow us to study the step-by-step distances between the original system and the perturbed one, which are abstracted away in the steady state analysis.

Initial concentration	Reaction constants	Chemical Reactions
$\mathsf{X} = 25\diamondsuit \quad \mathsf{Y} = 150\diamondsuit \quad \mathsf{XT} = 0$	$c_{a_1}, c_{a_2}, c_{a_3}, c_{a_4}, c_{a_7}, c_{a_8}, c_{a_{10}} = 0.5$	$\mathsf{XD} \underset{a_2}{\overset{a_1}{\rightleftharpoons}} \mathsf{X} \quad \mathsf{X_P} + \mathsf{Y} \underset{a_7}{\overset{a_6}{\rightleftharpoons}} \mathsf{X_P Y} \quad \mathsf{XDY_P} \xrightarrow{a_{10}} \mathsf{XD} + \mathsf{Y_P}$
$\mathsf{X_P} = 0 \quad \mathsf{X_P Y} = 0 \quad \mathsf{Y_P} = 10$	$c_{a_5}, c_{a_{11}} = 0.1$	$\mathsf{XT} \underset{a_4}{\overset{a_3}{\rightleftharpoons}} \mathsf{X} \quad \mathsf{X_P Y} \xrightarrow{a_8} \mathsf{X} + \mathsf{Y_P} \quad \mathsf{XDY_P} \xrightarrow{a_{11}} \mathsf{XD} + \mathsf{Y}$
$\mathsf{XDY_P} = 0 \quad \mathsf{XD} = 50$	$c_{a_6}, c_{a_9} = 0.02$	$\mathsf{XT} \xrightarrow{a_5} \mathsf{X_P} \quad \mathsf{XD} + \mathsf{Y_P} \xrightarrow{a_9} \mathsf{XDY_P}$

Table 3: The initial concentrations, the reaction constants and the chemical reactions of EnvZ/OmpR system. The concentration of input species is marked by $\diamondsuit$ and vary to estimate robustness

Let us focus on system S in Equation 1, whose initial species concentration are the same as in [15] and are reported in Table 3, together with the set $\mathcal{R}$ of reactions and the reaction constants. We start our analysis by studying the distances between S and two systems that were obtained in [15] by perturbing its initial state: system S_1 starts with $\mathsf{X} = 10$ and $\mathsf{Y} = 50$, system S_2 starts with $\mathsf{X} = 250$ and $\mathsf{Y} = 1000$. In [15] evidence is given that, at steady state, S, S_1 and S_2 have the same level of $\mathsf{Y_P}$. By applying function Simulate in Figure 2, we can simulate the evolution of these systems. For each system, we have applied $N = 100$ times the function Simulate with parameter $h = 15000$. In Figure 5 we report the step-by-step average value obtained for all relevant species, namely X, Y and $\mathsf{Y_P}$, and we highlight the values obtained for $\mathsf{Y_P}$.

By applying function Distance in Figure 4 we can estimate the evolution distance between the original system S and the two perturbed ones. We define the input distance $\mathbf{dd}_i$ so that only the weights associated with X and Y are positive, in particular we decided that both weights are 0.5 to reflect that the two species

(a) step-by-step values for $\mathbf{dd}_i(S, S_1)$ and $\mathbf{dd}_o(S, S_1)$ (b) step-by-step values for $\mathbf{dd}_i(S, S_2)$ and $\mathbf{dd}_o(S, S_2)$

Fig. 6: Evolution of $\mathbf{dd}_i$ and $\mathbf{dd}_o$ between S and its perturbed versions S_1 and S_2

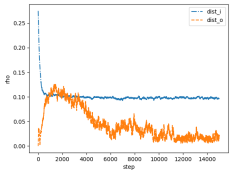
have the same relevance. Then, we define the output distance $\mathbf{dd}_o$ so that only the weight associated with $\mathsf{Y_P}$ is positive. In Table 4 we report the results obtained by applying DISTANCE with the following parameters: $N = 50$ runs for the original system, $\ell \cdot N = 100$ runs for the perturbed systems, $h = 15000$ steps for each run. For each perturbed system, in Table 4 we report: (i) the input distance between the perturbed system and S, which is simply computed by focusing on the initial level of input species in S and the perturbed systems, and the minimal and maximal level that are stored in m and M by DISTANCE; (ii) the maximal output distance $\mathbf{dd}_o$ computed by DISTANCE in interval $[7500, 15000]$; (iii) a pointer to a figure describing the step-by-step evolution of input and output distance between S and the perturbed system. Notice that $\mathbf{dd}_i(S, S_2)$ is one order of magnitude higher than $\mathbf{dd}_i(S, S_1)$, whereas $\mathbf{E}(\mathbf{dd}_o)_{[7500,15000]}(S, S_1)$ and $\mathbf{E}(\mathbf{dd}_o)_{[7500,15000]}(S, S_2)$ are close each other and are quite low numbers. This suggests that even if one changes the initial level of X and Y in a light or a heavy way, the step-by-step variation of $\mathsf{Y_P}$ is light in all cases.

System	Initial X	Initial Y	$\mathbf{dd}_i$ from S	$\mathbf{E}(\mathbf{dd}_o)_{[7500,15000]}$ from S	Figure showing $\mathbf{dd}_i$ and $\mathbf{dd}_o$ step-by-step
S_1	10	50	0.072407	0.051207	Figure 6a
S_2	250	1000	0.79145	0.056034	Figure 6b

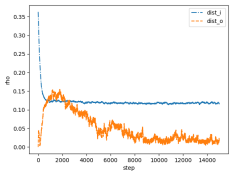
Table 4: Input and output distance between S and the two perturbed systems S_1 and S_2

However, in our setting, a more systematic analysis for studying how the initial concentrations of X and Y in a system S influence the concentration of $\mathsf{Y_P}$ in a temporal interval I can be conducted as follows: we fix the maximal input distance η_1 between system S and its perturbed versions and, then, we estimate for which η_2 the system S is $(\mathbf{dd}_i, \mathbf{dd}_o, I_1 = [0, 0], I, \eta_1, \eta_2)$-robust. More precisely, in order to estimate η_2, for a suitable n we sample n systems $S_1, \ldots, S_n$ satisfying $\mathbf{E}(\mathbf{dd}_i)_{[0,0]}(S, S_i) \leq \eta_1$ and we fix $\eta_2 = \max_{i=1,\ldots,n} \mathbf{E}(\mathbf{dd}_o)_I(S, S_i)$.

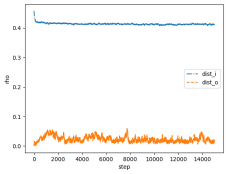
We have considered five different values for η_1: 0.3, 0.4, 0.5, 0.6, 0.7. For each choice for η_1 we have sampled $n = 20$ systems at input distance $\mathbf{dd}_i$ from S bounded by η_1. More precisely, we decided to sample these 20 systems so that they are at a $\mathbf{dd}_i$ distance from S in the interval $(\eta_1 - 0.1, \eta_1]$. Then, we have estimated the η_2 for which S is $(\mathbf{dd}_i, \mathbf{dd}_o, [0, 0], I, \eta_1, \eta_2)$-robust, for $I = [7500, 15000]$. In each experiment, we simulated runs consisting of $h = 15000$ reactions, and we considered $N = 50$ runs for the original system S and $\ell \cdot N = 100$ runs for the $n = 20$ perturbed systems. For each of the five choices for η_1, in Table 5 we report the value η_2 for which we have estimated the $(\mathbf{dd}_i, \mathbf{dd}_o, [0, 0], [7500, 15000], \eta_1, \eta_2)$-robustness, a pointer to the picture showing the step-by-step evolution of $\mathbf{dd}_i$ and $\mathbf{dd}_o$ between S and the perturbed system realizing η_2 and, finally, the initial concentration levels of X and Y of that system. Summarizing, at varying of η_1 in $[0.3, 0.7]$ we get values for η_2 that are close each other and are one order of magnitude lower than η_1. This suggests that S, that was classified as robust in the steady state analysis in [15], has a good level of robustness also in the step-by-step approach. The analysis in [15] allows one to conclude that at steady state the level of $\mathsf{Y_P}$ does not depend on the initial level of X and Y, our analysis allows to conclude that if one varies the initial level of $\mathsf{Y_P}$ then, step-by-step, the changes in $\mathsf{Y_P}$ are limited and smooth.



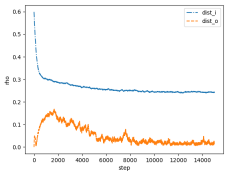
(a) $\mathbf{dd}_i$ and $\mathbf{dd}_o$, $\eta_1 = 0.3$



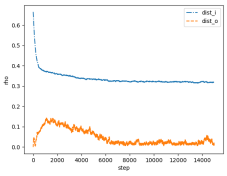
(b) $\mathbf{dd}_i$ and $\mathbf{dd}_o$, $\eta_1 = 0.4$



(c) $\mathbf{dd}_i$ and $\mathbf{dd}_o$, $\eta_1 = 0.5$



(d) $\mathbf{dd}_i$ and $\mathbf{dd}_o$, $\eta_1 = 0.6$



(e) $\mathbf{dd}_i$ and $\mathbf{dd}_o$, $\eta_1 = 0.7$

Fig. 7: EnvZ/OmR Osmoregulatory Signaling System: evolution of $\mathbf{dd}_i$ and $\mathbf{dd}_o$ at varying of η_1

The Python code used for the analysis can be found at https://github.com/dmanicardi/spebnr. For each value for η_1, the analysis took 150 minutes on a 1.70 GHz i7-1255U, with 16.00 GB RAM.

η_1	η_2	$\mathbf{dd}_i$ and $\mathbf{dd}_o$ between S and the system realizing η_2	Initial concentration of X of system realizing η_2	Initial concentration of Y of system realizing η_2
0.3	0.065862	Figure 7a	165	101
0.4	0.053621	Figure 7b	222	134
0.5	0.059483	Figure 7c	57	1033
0.6	0.078448	Figure 7d	269	500
0.7	0.059138	Figure 7e	263	669

Table 5: EnvZ/OmpR Osmoregulatory Signaling System: robustness at varying of η_1

6 Conclusion and Future Work

We have proposed a notion of robustness for biochemical networks that, essentially, evaluates the ability of a network to exhibit step-by-step limited variations on the quantity of a so called output species at varying of the initial concentration of some so called input species.

Recently, *Robustness Temporal Logic* [7] (RobTL) has been proposed for the specification and analysis of distances between the behaviours of cyber-physical systems over a finite time horizon. Atomic propositions are defined by means of two simple languages: one to specify the effect of *perturbations* over an evolution sequence, and one to specify *distance expressions* between an evolution sequence and its perturbed version. In detail, atomic expressions are of the form $\Delta(\mathsf{exp}, \mathsf{pert}) \bowtie \eta$, with $\bowtie \in \{<, \leq, =, \neq, >, \geq\}$, and allow for comparing a threshold η with the distance, specified by an expression exp, between an evolution sequence and its perturbed version, obtained by applying a perturbation specified by pert, starting from a given time step. Boolean and temporal operators allows for extending these evaluations to the entire evolution sequence. Then, the tool Stark [8] offers: (i) languages for specifying systems, perturbations, distance expressions, and RobTL formulae; (ii) a module for the simulation of systems behaviours and their perturbed versions; (iii) a module for the evaluation of distances between behaviours; (iv) a statistical model checker for RobTL formulae. An interesting future work consists in extending RobTL in order to allow for specifying and analysing properties of biochemical networks, and, then, enriching the Stark tool accordingly in order to use it for the robustness analysis of biochemical networks.

References

[1] Barkai, N., Leibler, S.: Robustness in simple biochemical networks. Nature **387**, 913–917 (1997)

[2] Barnes, D.J., Chu, D.F.: Introduction to modeling for biosciences (2010)

[3] Calder, M., Gilmore, S., Hillston, J.: Modelling the influence of RKIP on the ERK signalling pathway using the stochastic process algebra PEPA. Trans. Comp. Sys. Biology **7**, 1–23 (2006). https://doi.org/10.1007/11905455_1, https://doi.org/10.1007/11905455_1

[4] Castiglioni, V., Loreti, M., Tini, S.: Measuring adaptability and reliability of large scale systems. In: International Symposium on Leveraging Applications of Formal Methods. pp. 380–396. Springer (2020)

[5] Castiglioni, V., Loreti, M., Tini, S.: A framework to measure the robustness of programs in the unpredictable environment. Logical Methods in Computer Science, to appear, available as arXiv preprint arXiv:2111.15319 (2021)

[6] Castiglioni, V., Loreti, M., Tini, S.: How adaptive and reliable is your program? In: International Conference on Formal Techniques for Distributed Objects, Components, and Systems. pp. 60–79. Springer (2021)

[7] Castiglioni, V., Loreti, M., Tini, S.: RobTL: A temporal logic for the robustness of cyber-physical systems. CoRR **abs/2212.11158** (2022). https://doi.org/10.48550/arXiv.2212.11158

[8] Castiglioni, V., Loreti, M., Tini, S.: Stark: A software tool for the analysis of robustness in the unknown environment. In: International Conference on Coordination Models and Languages. Springer (2023)

[9] Chong, S., Lanotte, R., Merro, M., Tini, S., Xiang, J.: Quantitative robustness analysis of sensor attacks on cyber-physical systems. In: Proceedings of the 26th ACM International Conference on Hybrid Systems: Computation and Control, HSCC 2023, San Antonio, TX, USA, May 9-12, 2023. pp. 20:1–20:12. ACM (2023)

[10] Gillespie, D.T.: Exact stochastic simulation of coupled chemical reactions. The journal of physical chemistry **81**(25), 2340–2361 (1977)

[11] Kantorovich, L.V.: On the translocation of masses. In: Dokl. Akad. Nauk. USSR (NS). vol. 37, pp. 199–201 (1942)

[12] Kitano, H.: Towards a theory of biological robustness. Molecular Systems Biology **3**(1), 137 (2007)

[13] Lanotte, R., Merro, M., Tini, S.: A probabilistic calculus of cyber-physical systems. Inf. Comput. **279**, 104618 (2021)

[14] Merro, M., Ballardin, F., Sibilio, E.: A timed calculus for wireless systems. Theor. Comput. Sci. **412**(47), 6585–6611 (2011)

[15] Nasti, L., Gori, R., Milazzo, P.: Formalizing a notion of concentration robustness for biochemical networks. In: Federation of International Conferences on Software Technologies: Applications and Foundations. pp. 81–97. Springer (2018)

[16] Rachev, S.T., Klebanov, L.B., Stoyanov, S., Fabozzi, F.J.: The methods of distances in the theory of probability and statistics (2013)

[17] Rizk, A., Batt, G., Fages, F., Soliman, S.: A general computational method for robustness analysis with applications to synthetic gene networks. Bioinform. **25**(12) (2009). https://doi.org/10.1093/bioinformatics/btp200, https://doi.org/10.1093/bioinformatics/btp200

[18] Rizk, A., Batt, G., Fages, F., Soliman, S.: Continuous valuations of temporal logic specifications with applications to parameter optimization and robustness measures. Theor. Comput. Sci. **412**(26), 2827–2839 (2011). https://doi.org/10.1016/j.tcs.2010.05.008, https://doi.org/10.1016/j.tcs.2010.05.008

[19] Shinar, G., Feinberg, M.: Structural sources of robustness in biochemical reaction networks. Science **327**(5971), 1389–1391 (2010)

[20] Shinar, G., Feinberg, M.: Design principles for robust biochemical reaction networks: what works, what cannot work, and what might almost work. Mathe. Biosci **231**, 39–48 (2011)

[21] Sriperumbudur, B.K., Fukumizu, K., Gretton, A., Schölkopf, B., Lanckriet, G.R.G.: On the empirical estimation of integral probability metrics. Electronic Journal of Statistics **6**, 1550–1599 (2021). https://doi.org/10.1214/12-EJS722

[22] Thorsley, D., Klavins, E.: Approximating stochastic biochemical processes with Wasserstein pseudometrics. IET Syst. Biol. **4**(3), 193–211 (2010). https://doi.org/10.1049/iet-syb.2009.0039

[23] Uri, A.: An introduction to systems biology: design principles of biological circuits. Chapman and Hall/CRC (2006)
[24] Villani, C.: Optimal transport: Old and new (2008)
[25] Zhou, K., Doyle, J.C.: Essentials of Robust Control. Prentice-Hall (1998)