Quantitative Analysis of Robustness in Systems Biology: Combining Global and Local Approaches

THÈSE N^{O} 4907 (2010)

PRÉSENTÉE LE 10 DÉCEMBRE 2010

À LA FACULTÉ INFORMATIQUE ET COMMUNICATIONS
LABORATOIRE DE SYSTÈMES NON LINÉAIRES
PROGRAMME DOCTORAL EN INFORMATIQUE, COMMUNICATIONS ET INFORMATION

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Marc HAFNER

acceptée sur proposition du jury:

Prof. J.-Y. Le Boudec, président du jury
Prof. M. Hasler, Prof. H. Köppl, directeurs de thèse
Prof. F. Naef, rapporteur
Prof. A. Wagner, rapporteur
Prof. R. Weiss, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2010

Abstract

To characterize the behavior and robustness of cellular circuits is a major challenge for Systems Biology. Many of the published methods that address this question quantify the local robustness of the models. In this thesis, I tried to underpin the inappropriateness of such local measures and proposed an alternative solution: a glocal measure for robustness that combines both global and local aspects. It comprises a broad exploration of the parameter space and a further refinement based on different local measures. The method is general and such glocal analysis could be applied to many problems.

Along with the theoretical and formal aspects of this new analysis method, I developed sampling algorithms that efficiently investigate the generally high-dimensional parameter space of models. To show the usefulness of my method, I applied it on different models of cyclic systems such as the circadian clock and the mitotic cycle. I first considered two models of the cyanobacterial circadian clock and compared their robustness properties. Also in the context of circadian rhythms, I studied the effect of additional feedback loops on the robustness properties in relation with entrainment. Models of the mitotic cycle are also used to assess the effect of an additional positive feedback loop on circuit robustness to parameter changes and molecular noise. Finally, I established some principles for the design of a synthetic circuit based on robustness.

The thesis carries on with a discussion that emphasizes the advantages of the glocal method for robustness analysis: in all works, correlations between parameter values and local robustness can be found. Such results facilitate our understanding of the biochemical systems and can be a guide for new experiments to discriminate models or give directions for altering the robustness of the systems. I conclude by discussing potential applications for my method and possible improvements.

Keywords: robustness, systems biology, sampling methods, biological oscillators, feedback loops.

Résumé

La caractérisation de la robustesse des réseaux biochimiques est actuellement un des défis majeurs en biologie des systèmes. La plupart des méthodes couramment utilisées pour étudier cette question sont basées sur des analyses locales. Ce travail tente de démontrer que cette approche est inappropriée et que des méthodes alternatives doivent être développées. J'ai donc proposé une méthode glocale pour la quantification de la robustesse qui combine des mesures globales et locales. Cette méthode comprend une large exploration de l'espace des paramètres, complétée par une caractérisation locale de la robustesse grâce à différentes mesures. Les principes de cette méthode sont généraux et de telles analyses glocales peuvent être utilisées pour une multitude de problèmes.

En parallèle avec l'élaboration des concepts et du formalisme de cette nouvelle méthode, j'ai développé des algorithmes pour l'échantillionnage de l'espace des paramètres. Afin de démontrer l'utilité de ma méthode, je l'ai appliquée sur différents modèles de cycles biologiques. Premièrement, j'ai comparé la robustesse de deux modèles de cycles circadiens de la cyanobactérie. Dans le même contexte, j'ai étudié l'effet d'une boucle de feedback additionnelle sur la robustesse à l'entraînement des cycles circadiens. Pour continuer la comparaison de différentes topologies, j'ai utilisé un modèle du cycle mitotique et j'ai caractérisé la robustesse aux changements des paramètres et au bruit moléculaire lorsque qu'un feedback positif est ajouté. Finalement, j'ai dérivé des principes basés sur la robustesse pour le design de circuits en biologie synthétique .

Dans les différents travaux, l'analyse glocale met en valeur des corrélations entre les paramèteres et la robustesse locale. Ces résultats, qui ne peuvent pas être obtenus avec des méthodes classiques, aident à comprendre les systèmes étudiés et peuvent servir de guide pour établir des nouvelles expériences ou donner des directions pour altérer la robutesse des systèmes. Pour conclure, d'autres applications pour ma méthode et de possibles améliorations sont proposées.

Mots-clés: robustesse, biologie des systèmes, méthodes d'échantillionnage, oscillateurs biologiques, boucles de feedback.

Acknowledgments

I first would like to express my gratitude to my advisors, Prof. Heinz Koeppl, now at ETH Zurich, Prof. Andreas Wagner from University of Zurich and Prof. Martin Hasler from EPFL. They all contributed to this thesis with their advices and support. Then, I would like to thank the different collaborators I have been working with during these years: Pierre Sacré and Prof. Rodolphe Sepulchre from Université de Liège, Dr. Didier Gonze from Université Libre de Bruxelles, Elias Zamora from University of Zurich and a special thank to Prof. Ron Weiss from MIT who welcomed me in his lab during autumn 2009. I also would like to thank all the members of the Laboratory of Nonlinear Systems at EPFL for the great working atmosphere and SystemsX.ch for the funding of this project. Finally, this work would not have been possible without the support of my family and Tonia to whom I am grateful.

Contents

List of Figures

Chapter 1

Introduction

In this chapter, I first define Computational Systems Biology and biochemical modeling in general terms. Then, in Section 1.2, I introduce some formalism for the modeling of biochemical systems and expand on the state of the art of algorithms used for stochastic simulations. In Section 1.3, the recent literature on robustness analysis is reviewed. Section 1.4 is dedicated to biological oscillators, more specifically circadian clocks. The Section 1.5 briefly introduces synthetic biology and the last section of this chapter discusses the different contributions of this thesis.

1.1 Introduction to Computational Systems Biology

Systems Biology is a field that emerged in the last fifteen years. It has risen as an inter-disciplinary science where mathematical analysis came to complement biological experiments. Systems Biology can be described as the study of the interactions between various components involved in biological processes. The scale can go from atomic interactions to ecosystems. The birth of Systems Biology was induced by the advances in molecular biology in 1980s and the rise of functional genomics in 1990s. With an understanding of the cellular processes at the genetic and molecular levels, rigorous mathematical models based on experimental data started to emerge. At the beginning of the 2000s, the completion of various genome projects, the large increase in data from the

high-throughput experiments in the omics technologies (e.g. genomics and proteomics), and the increased computational power are the factors that may have triggered the emergence of Systems Biology [1].

One branch of Systems Biology is Computational Systems Biology which could be defined as the *quantitative modeling of the biochemical process* occurring in living cells. It aims to characterize the interactions between proteins, genes and other cellular components. The size of the studied systems can range from small gene network with several interactions to metabolic networks where thousands of species are found. The most important aspect of Computational Systems Biology may be the use of quantitative models to understand these systems.

1.1.1 Models in Computational Systems Biology

A model is a conceptual representation, therefore a simplification, of a system. The loss of detail is compensated by the gain in clarity and the possibility to implement the system in a computer. A model being an abstraction of a system, each system can be modeled in different ways depending on the desired results. In this sense, the scope of any model is limited and this limitation has to always be considered when analyzing any computational results.

In Computational Systems Biology, the variables of the mathematical models usually correspond to mRNA molecules and proteins [1, 2, 3, 4]. These different molecular species influence each other through, for example, transcriptional regulation (some proteins control the transcription of mRNAs), translation (mRNA is necessary to form proteins) or enzymatic activity (some reactions require a specific protein). When modeling one of these systems, the natural state variables are the concentrations of these species and the complexes that they may form. The dynamic interactions between the different molecules are encapsulated in equations that determine the fluxes at which they are synthesized or degraded, at which they associate, dissociate, or are transformed into other molecules. To each flux is associated a rate function that depends on the species concentrations and some biochemical parameters.

Any complex reaction or process can be decomposed in a sequence of simple, unimolecular or bimolecular reactions following mass-action kinetics where the rate function is a polynomial expression of the concentrations of the reactants. Each intermediate stage or molecular complex represents a new species and this decomposition may result in a large number of variables. The strength

of a model is to prevent this combinatorial explosion by simplifying some reactions in a single step. This is done in general using non-polynomial expressions for the rate functions [1, 3, 4]. Such rate function includes more parameters and mimics the actual dynamics. The Hill function is a well-known example of such simplification: a nonlinear term is used to approximate cooperativity effects in gene expression. The use of such simplifications should be cautious and in general, *"The modeler should always choose the correct level of detail to answer the question"* [5]. For example, if one wants to understand the effect on gene expression of changing the binding affinity of proteins to promoters, the use of Hill functions is not appropriate as this step is highly simplified. But on the contrary in the context of gene networks, strong simplifications can be made to keep a reasonable number of variables in the system without significantly altering the results.

When the topology of the model is fixed, i.e. the interactions between the different species are established, and when the rate functions are determined, the dynamics depends only on the rate constants [2, 1]. These rate constants, or model parameters, reflect different external factors such as temperature or ATP concentration. An important challenge in Computational Systems Biology is to find the parameters that best fit the data. Indeed, the values of these parameters can drastically change the qualitative behavior of the system. For example, a model may show oscillations only for specific parameter values. A central part of my work is to analyze the effect of changing the parameters on the robustness properties of a system.

1.2 Stochastic Simulations in Systems Biology

All the interactions and their corresponding rate constants give rise to a system of ordinary differential equations (ODE) [6], whose state variables are the concentrations of each molecular species as in classical chemical kinetics [7]. There is an extensive theory for ODE integration and many tools from control theory and dynamical system theory are available [3]. I will not discuss the details of the theory behind ODE integration as I only used algorithms from standard libraries in this thesis. On the other hand, I will spend some time on the different stochastic simulation algorithms used for the integration of biochemical systems.

When comparing biochemical processes to the ones encountered in chemical engineering, some cellular species such as transcription factors or mRNA are

present in very few copies. Under these circumstances, the continuous approach of ODEs falls short and only a discrete approach capturing the stochastic nature of chemical events can properly describe such dynamics [8, 9]. In this section, I will explain the formalism and the main algorithms used to perform stochastic simulations of such continuous-time Markov jump processes.

This section is organized as follows. First, I will formally introduce the traditional law of mass action, the corresponding rate equations and a formalism that will be used later. Then I will discuss the Markov jump process associated with stochastic chemical kinetics. Based on that, a Monte Carlo sampling algorithm developed by Gillespie [10] along with its many variants is explained. Subsequently, the first approximate algorithm, the τ-leaping method is derived and some algorithmic aspects of it are highlighted. Later, further assumptions are made to obtain the chemical Langevin equation and its simulation is briefly discussed.

1.2.1 The Law of Mass Action

Chemical reactions, and as a consequence most events occurring in cellular processes, can be represented as transformations of molecular species $S_i \in \{S_1, \ldots, S_n\}$. A reaction R_j, $j \in \{1, \ldots, m\}$ with a rate constant $k_j \in \mathbb{R}_+$, is written in a general form as

$$R_{1j}S_1 + \cdots + R_{nj}S_n \xrightarrow{k_j} P_{1j}S_1 + \cdots + P_{nj}S_n \tag{1.1}$$

where $R_{ij} \in \mathbb{N}_0$ is the stoichiometric coefficient for the species S_i as a reactant and $P_{ij} \in \mathbb{N}_0$ its coefficient as a product [11].

In the most simplified approach, all reactions of a process follow the law of mass action, originally proposed by Waage and Guldberg (1864): the rate of a reaction is the product of the concentration of the reactants and a constant. Thus the mass-action rate function $v_j : \mathbb{R}_+^n \times \mathbb{R}_+^m \to \mathbb{R}_+$ of reaction R_j is given by

$$v_j(\mathbf{z}, \mathbf{k}) = k_j \prod_{i=1}^{n} z_i^{R_{ij}}, \tag{1.2}$$

with $\mathbf{k} \equiv (k_1, \ldots, k_m)^T$ and where the concentration of each species S_i is denoted as $z_i(t)$, $i \in \{1, \ldots, n\}$. The state of the system is thus the time function $\mathbf{z} : \mathbb{R}_+ \to \mathbb{R}_+^n$.

With the stoichiometric matrix $\mathbf{N} \in \mathbb{Z}^{n \times m}$, the elements of which are $N_{ij} = P_{ij} - R_{ij}$, that represents the net effect of reaction R_i on the species S_j,

the general ODE system can be written in vectorial form as

$$\frac{\mathrm{d}\mathbf{z}}{\mathrm{d}t} = \mathbf{N}\mathbf{v}(\mathbf{z}, \mathbf{k}). \tag{1.3}$$

In the case of reaction that do not follow mass action kinetics such as Michaelis-Menten simplifications, equation (1.3) remains valid, but the reaction rates cannot be expressed as equation (1.2).

1.2.2 The Chemical Master Equation

The formalism introduced above assumes a continuous number of molecules for each species type. The molecules being integer, such equations are a good approximation only for large copy numbers. In chemical engineering, for instance, where the number of molecules is in the order of the Avogadro's number, the simulations made with ODEs are perfectly consistent with the experimental setting. But for biomolecular systems, the number of molecules in a cell could be in the order of a few hundred (for example transcription factors or messenger-RNA or micro-RNA) to a handful for gene copies. With so few molecules, the continuous approximation is not valid anymore and major differences are observed when the deterministic results are compared to experimental data in specific cases [12, 13, 14].

To capture the discrete nature of molecules and the stochasticity of reactions, a formalism based on continuous-time Markov jump processes can be adopted. The Markov property, i.e. the probability that a reaction occurs depends only on the current state and not on the history of the system, is a consequence of the physics of the chemical reactions [15]. To link the species' multiplicity with its concentration we introduce the reaction volume Ω. The state variables can now assume positive integer values, i.e., the system is in the state $\mathbf{X}(t) = \mathbf{x}$, if, for all species S_i, $X_i = x_i$ at time t with $\mathbf{x} \subseteq \mathbb{N}^n$. The t-indexed random variable X_i is related to the concentrations used in the ODEs as $X_i = \Omega z_i$. The probability for a system to be in such a state is $\Pr(\mathbf{X}(t) = \mathbf{x})$. Finally, the central definition for the Markovian formulation can be introduced. The likelihood to switch from a state $\mathbf{x}_0$ at time t_0 to the state $\mathbf{x}$ at time t is the conditional probability $\Pr(\mathbf{X}(t) = \mathbf{x}|\mathbf{X}(t_0) = \mathbf{x}_0) \equiv \Pr(\mathbf{x}, t|\mathbf{x}_0, t_0)$. In the following, an expression for this transition probability is derived.

To switch from one state to another, reactions have to occur. But a reaction R_j can only change the system in a specific way. If the state prior to the reaction is $\mathbf{x}_0$, then the next state will be $\mathbf{x} = \mathbf{x}_0 + \boldsymbol{\nu}_j$ where $\boldsymbol{\nu}_j$ is the stoichiometric

Reaction type (with rate constant k_j)	Corresponding stochastic rate constant c_j	Corresponding propensity a_j
$\emptyset \rightarrow$ products	$k_j\Omega$	$k_j\Omega$
$S_1 \rightarrow$ products	k_j	$k_j \quad x_1$
$S_1 + S_2 \rightarrow$ products	k_j/Ω	$k_j/\Omega \quad x_1x_2$
$S_1 + S_1 \rightarrow$ products	$2k_j/\Omega$	$k_j/\Omega \quad x_1(x_1-1)$
$S_1 + S_2 + S_3 \rightarrow$ products	k_j/Ω^2	$k_j/\Omega^2 \quad x_1x_2x_3$

Table 1.1: Elementary reactions of different arity (left) and the corresponding propensities (right); rescaling of reaction rate constants k_j to the stochastic rate constants c_j (middle).

vector corresponding to the j-th column of the stoichiometric matrix. That is, $\boldsymbol{\nu}_j = \mathbf{N}\mathbf{e}_j$, whereas $\mathbf{e}_j$ is the j-th basis vector of an m-dimensional vector space. With this, the evolution of the conditional probability can be expressed by the balance equation

$$\begin{aligned} \Pr(\mathbf{x}, t+\mathrm{d}t|\mathbf{x}_0, t_0) &= \Pr(\mathbf{x}, t|\mathbf{x}_0, t_0)\Pr\left(\text{no reaction in } [t, t+\mathrm{d}t]\right) \\ &+ \sum_{j=1}^{m} \Pr(\mathbf{x}-\boldsymbol{\nu}_j, t|\mathbf{x}_0, t_0)\Pr\left(\text{one reaction } R_j \text{ in } [t, t+\mathrm{d}t]\right). \end{aligned} \tag{1.4}$$

The probability for R_j to occur in the state $\mathbf{x}$ within a time interval dt is $a_j(\mathbf{x})dt + \mathcal{O}(dt^{\geq 2})$, where the function $a_j : \mathbb{N}_0^n \rightarrow \mathbb{R}_+$ is referred to as *propensity* or *hazard* of reaction R_j. According to the law of mass action the propensity of reaction R_j is the combinatorial number of ways the product can be formed. That is,

$$a_j(\mathbf{x}) = c_j \prod_{i=1}^{n} \binom{x_i}{R_{ij}},$$

with c_j, the stochastic rate constant. How that rate is related to the rate constant k_j is exemplified in the Table 1.1 for a few elementary reactions. The correspondence is obtained by noting that

$$\binom{x_i}{R_{ij}} \approx \frac{x_i^{R_{ij}}}{R_{ij}!} \quad \text{for} \quad x_i \gg R_{ij}.$$

With this, the terms of equation 1.4 can be expressed for sufficiently small δt as

$$\Pr\left(\text{no reaction in } [t, t+\delta t] at state \mathbf{x}\right) = 1 - \sum_{j=1}^{m} a_j(\mathbf{x})\delta t \tag{1.5}$$

and

$$\Pr\left(\text{one reaction } R_j \text{ in } [t, t+\delta t] at state \mathbf{x}\right) = a_j(\mathbf{x}-\boldsymbol{\nu}_j)\delta t \tag{1.6}$$

With equations 1.4 to 1.6 and δt taken to the limit, we obtain the chemical master equation (CME) [15, 16, 17]

$$\begin{aligned}\frac{\partial \Pr(\mathbf{x}, t|\mathbf{x}_0, t_0)}{\partial t} &= \lim_{\delta t \to 0} \frac{\Pr(\mathbf{x}, t+\delta t|\mathbf{x}_0, t_0) - \Pr(\mathbf{x}, t|\mathbf{x}_0, t_0)}{\delta t} \\ &= \sum_{j=1}^{m} [a_j(\mathbf{x} - \boldsymbol{\nu}_j)\Pr(\mathbf{x} - \boldsymbol{\nu}_j, t|\mathbf{x}_0, t_0) \\ &\quad - a_j(\mathbf{x})\Pr(\mathbf{x}, t|\mathbf{x}_0, t_0)] \,. \end{aligned} \tag{1.7}$$

1.2.3 Stochastic Simulation Algorithms

The CME completely determines the conditional probability $\Pr(\mathbf{x}, t|\mathbf{x}_0, t_0)$, therefore, given an initial probability distribution $\Pr(\mathbf{x}_0, t_0)$ the evolution of this probability can be solved. However, except for very simple cases, the CME is a set of ODEs that is exponentially large with respect to n and neither analytical nor efficient computational solutions are available. One way to extend the applicability of the direct solution of (1.8) is to approximate the solution by restricting the state space of the CME to those states that carry significant probability mass [18, 19].

An approach that is much more scalable is to perform Monte Carlo simulations of the reaction system. In this case, we draw a random sample path of the Markov jump process that represents a succession of reactions occurring randomly according to their propensities. The theory and algorithm for exact stochastic simulations were introduced by Gillespie in 1976 [20] and was further detailed in [15]. The probability $p(\tau, j|\mathbf{x}, t)\delta\tau$ is defined as the probability that the next reaction in the system will be the j-th one and that it will occur in the time interval $[t+\tau, t+\tau+\delta\tau]$, $\tau \in [0, \infty)$ given that the system is in the state $\mathbf{X}(t) = \mathbf{x}$ at time t. This joint probability can be decomposed. That is, the probability $p_0(\tau, \mathbf{x})$ that no reaction occurs in the interval $[t, t+\tau]$ multiplied by the probability $a_j(\mathbf{x})\delta\tau$ that the reaction R_j occurs in the interval $[t+\tau, t+\tau+\delta\tau]$.

$$p(\tau, j|\mathbf{x}, t)\delta\tau = p_0(\tau, \mathbf{x})a_j(\mathbf{x})\delta\tau. \tag{1.8}$$

To obtain an expression for the density $p_0(\tau, \mathbf{x})$, the distribution of the time intervals between two reactions has to be evaluated. Focusing on a system with a single reaction R_1, the probability that this reaction does not occur in the interval $[t+\tau, t+\tau+\delta\tau]$ is $p_0(\tau+\delta\tau, \mathbf{x}) = p_0(\tau, \mathbf{x})(1 - a_1(\mathbf{x})\delta\tau)$. Therefore,

the evolution of $p_0(\tau, \mathbf{x})$ is

$$\begin{aligned} \frac{\mathrm{d}p_0(\tau, \mathbf{x})}{\mathrm{d}\tau} &= \lim_{\delta\tau \to 0} \frac{p_0(\tau + \delta\tau, \mathbf{x}) - p_0(\tau, \mathbf{x})}{\delta\tau} \\ &= -a_1(\mathbf{x}) p_0(\tau, \mathbf{x}), \end{aligned} \tag{1.9}$$

which implies, with the condition $p_0(0, \mathbf{x}) = 1$ and equation (1.8), that the probability density for the next reaction to occur at time τ is

$$p(\tau, \mathbf{x}) = a_1(\mathbf{x}) e^{-a_1(\mathbf{x})\tau}. \tag{1.10}$$

This distribution of the next event follows the Poisson law with a mean equal to $a_1(\mathbf{x})\tau$. Such a distribution is found in many physical systems with random events such as radioactive degradation and its properties will be used later in this chapter [16]. Going back to the whole reaction system with this result, the distribution for the probability that no reaction occurs during the time interval τ, $p_0(\tau, \mathbf{x})$, is the product of the probabilities that no single reactions occurs in this interval,

$$p_0(\tau, \mathbf{x}) = e^{-a_0(\mathbf{x})\tau},$$

with $a_0(\mathbf{x}) = \sum_{j=1}^{m} a_j(\mathbf{x})$. The desired density probability that the j-th reaction occurs at time τ can then be expressed as

$$p(\tau, j|\mathbf{x}, t) = a_j(\mathbf{x}) e^{-a_0(\mathbf{x})\tau}. \tag{1.11}$$

Using the definition of conditional probability, the joint probability density function can be factorized into $p(\tau, j|\mathbf{x}, t) = p_1(\tau|\mathbf{x}, t) p_2(j|\tau, \mathbf{x}, t)$ where the term $p_1(\tau|\mathbf{x}, t)\mathrm{d}\tau$ is the probability that the next reaction will occur in the interval $[t + \tau, t + \tau + \delta\tau]$, regardless of which reaction it might be, and the term $p_2(j|\tau, \mathbf{x}, t)$ is the probability that this next reaction is the j-th one. It can be seen in

$$\begin{aligned} p_1(\tau|\mathbf{x}, t) &= \sum_{j=1}^{m} p(\tau, j|\mathbf{x}, t) \\ &= a_0(\mathbf{x}) e^{-a_0(\mathbf{x})\tau} \quad \text{with} \quad \tau \in [0, \infty) \end{aligned} \tag{1.12}$$

and

$$\begin{aligned} p_2(j|\tau, \mathbf{x}, t) &= \frac{p(\tau, j|\mathbf{x}, t)}{\sum_{j'=1}^{m} p(\tau, j'|\mathbf{x}, t)} \\ &= \frac{a_j(\mathbf{x})}{a_0(\mathbf{x})} = p_2(j|\mathbf{x}, t) \quad \text{with} \quad j \in \{1, \ldots, m\} \end{aligned} \tag{1.13}$$

that p_1 depends only on τ and p_2 depends only on j. Therefore, the two random variables τ and j, that follow the distributions 1.12 and 1.13, respectively, can be drawn independently.

1.2.4 Implementation of Gillespie's Algorithm

In this part, different implementations of Gillespie's stochastic simulation algorithm will be discussed. All these methods are called exact as the simulations are made without any approximation. As a consequence, the results of those variants are equivalent.

The First Reaction Method

The straightforward way to implement Gillespie's algorithm is called the *first reaction method*. For each reaction R_j, a value τ_j for the next occurrence is chosen according to the exponential distribution $\tau_j \sim e^{-a_j}$. Then, the smallest τ_j is chosen and the corresponding reaction is executed. After the update of the state and the advance in time, all propensities must be evaluated again and a whole set of τ_j has to be drawn again. For each reaction, m new random numbers need to be drawn, which is computationally demanding. Due to its computational cost, this implementation of the algorithm is practically never used.

The Direct Method

The original implementation proposed by Gillespie is called the *direct method* [15] and takes advantage of the two independent probability distribution discussed above (Eq. 1.12 and 1.13). By drawing two uniform random numbers r_1, r_2, the time to the next reaction τ is given by

$$\tau = \frac{1}{a_0(\mathbf{x})} \ln\left(\frac{1}{r_1}\right), \tag{1.14}$$

and the next occurring reaction R_J is determined by the following condition

$$\sum_{j=1}^{J} a_j(\mathbf{x}) > r_2 a_0(\mathbf{x}). \tag{1.15}$$

The system is then updated to $t \leftarrow t + \tau$ and $\mathbf{x} \leftarrow \mathbf{x} + \boldsymbol{\nu}_j$. Algorithm 1 is a pseudo-code of this implementation. For each reaction, only two random numbers are drawn, which is much more efficient than the first reaction method. However, all propensities are evaluated at each time step and further improvements have been made to limit the number of those operations.

```
// Initialization
t ← 0;
X ← Initial conditions ;
// Main loop
while t < T do
    a_0 ← 0 ;                          // Evaluation of the propensities
    for j ← 1 to m do
        a_j ← propensity for R_j with X(t) ; // According to Table 1.1
        a_0 ← a_0 + a_j ;
    end
    // next reaction time (Eq. 1.12)
    Draw a random number r_1 ∼ U[0,1], τ ← − ln(r_1)/a_0 ;
    // Reaction selection (Eq. 1.13)
    Draw a random number r_2 ∼ U[0,1], R ← a_0 r_2 ;
    for j ← 1 to m do
        R ← R − a_j ;
        if R ≤ 0 then
            J ← j ;                    // J is the chosen reaction
            Break ;
        end
    end
    X ← X + ν_J ;                      // Reaction execution
    t ← t + τ ;
end
```

Algorithm 1: Pseudo-code for simulation of the Gillespie algorithm with the direct method. The time is t, the state variable $\mathbf{X}$. The simulation is performed until time T.

The Next Reaction Method

One direction to improve efficiency was proposed by Gibson and Bruck [21]. It is based on the first reaction method, but a minimal number of propensities (and therefore τ_j) are evaluated at each time step. Knowing the expression for the propensities, a dependency graph D can be created to describe which state variables are updated when a particular reaction fires and consequently, which propensities must be recalculated. For example, let R_j be the next reaction to occur after time τ_j. If the state variables involved in the expression for a_k were not changed by R_j, τ_k can be retained. Else, if some components X_i, on which the propensity R_k depends on, are changed by R_j, a_k has to be re-evaluated. The random number for the next occurrence of this reaction

τ_k can be updated as $\tau_k \leftarrow (a_{j,old}/a_{j,new})(\tau_k - \tau_j) + \tau_j$. In general, Monte Carlo method simulations assume statistically independent random numbers and such a re-use of τ_k seems not legitimate. However, in this particular case [21], it has been proved to be valid with the change from relative to absolute time. For this method, τ_j is the absolute time of the next occurrence instead of the time interval until the next occurrence. In summary, only one random number is drawn and a minimal number of propensities are evaluated at each time step, reducing drastically the computational cost of the simulation.

A further enhancement has been implemented in the next reaction method. The different τ_j are sorted such that the search for the next reaction is fast. An efficient way to do this is to place all reaction times in a binary tree such that all children have a larger τ_j than its parent. For large systems, the cost of updating the tree is advantageously balanced by the gain in finding the minimal τ_j.

The Sorting Direct Method

The idea of avoiding the computation of all propensities at each step has been applied also to the direct method. An additional optimization to limit the number of operations on the summation occurring in the reaction choice (Eq. 1.15) has been proposed by Cao *et al.* [22]. Therein, the summation starts with the reaction with the largest propensities such that the random value $r_2 a_0$ is reached earlier. This requires *a priori* knowledge of the most frequent reaction. This aspect has been further enhanced in the *sorting direct method* by McCollum *et al.* in [23] that includes a simple dynamic sorting of the reactions. The pseudo-code of the method is presented in Algorithm 2. This method needs two random numbers per step, but with the most efficient random number generators such as the Mersenne Twister algorithm, also known as *mt19937* [24], it proves to be at least as efficient as the next reaction method [23].

1.2.5 Approximate Simulation Methods – τ-leaping

Even when optimized, exact simulation algorithms scale badly with the number of molecules in the reaction system. With a high copy number, the propensities are large and the time for the next reaction τ is small. It results in a large number of steps for a small advance in physical simulation time. Approximation methods have been proposed to counteract this problem [25, 10]. With a large number of molecules, the propensities can be considered

```
// Initialization
t ← 0;
X ← Initial conditions ;
O ← [1 ... m] ;                                  // Reaction search order
D ← dependency matrix;
a_0 ← 0 ;                          // Initialization of the propensities
for j ← 1 to m do
    a_j ← propensity for R_j with X(0) ;    // According to Table 1.1
    a_0 ← a_0 + a_j ;
end
// Main loop
while t < T do
    // next reaction time (Eq. 1.12)
    Draw a random number r_1 ~ U[0,1], τ ← −ln(r_1)/a_0 ;
    // Reaction selection (Eq. 1.13)
    Draw a random number r_2 ~ U[0,1], R ← a_0 r_2 ;
    for j ← 1 to m do                  // Loop according to the ordering
        R ← R − a_{O(j)} if R ≤ 0 then
            RO ← j ;                // The j-th ordered reaction is chosen
            Break ;
        end
    end
    J ← O(j) ;                                  // J is the chosen reaction
    X ← X + ν_J ;                                        // Reaction execution
    t ← t + τ ;
    if RO ≠ 1 then                  // Dynamic ordering of the reactions
        swap O(RO − 1) and O(RO)
    end
    foreach j ∈ D(J) do                // Update necessary propensities
        a_0 ← a_0 − a_j ;
        a_j ← propensity for R_j with X(t) ; // According to Table 1.1
        a_0 ← a_0 + a_j ;
    end
end
```

Algorithm 2: Pseudo-code for Gillespie's algorithm with the sorting direct method. The time is t, the state variable $\mathbf{X}$. The simulation is performed until time T.

constant over a small time step. This is the main idea of the *tau-leaping methods*. For a defined time step τ, the number of reactions σ_j occurring is chosen according to a Poisson distribution $\mathcal{P}$ with a mean equal to the $a_j\tau$. The different σ_j are assumed to be independent for all reactions. As for first order methods to integrate ODEs, the update of the system state from $\mathbf{X}(t) = \mathbf{x}$ is then made with finite time leap

$$\mathbf{X}(t+\tau) = \mathbf{x} + \sum_{j=1}^{m} \sigma_j \boldsymbol{\nu}_j \quad \text{with } \sigma_j \sim \mathcal{P}(a_j(\mathbf{x})\tau). \tag{1.16}$$

The τ-leaping method is a valid approximation if the following two conditions hold. If (1) the time step τ is small enough such that the propensities change marginally for the number of reactions that occurs during τ and (2) that no state variable can reach negative values due to a too large jump. However, choosing a small τ will result in very few reactions occurring during each time step and therefore reproducing Gillespie's algorithm. The efficiency and accuracy of τ-leaping algorithms depend crucially on the choice of τ. Most algorithms have a dynamic assignment of τ as a function of the propensities. For instance, they ensure that the relative variation of a_j during the time step τ, i.e. $\frac{\Delta_\tau a_j}{a_j}$, remains below a threshold ϵ with $0 < \epsilon \ll 1$. The basic algorithm is given as pseudo-code in Algorithm 3.

Determination of the Leap Size

The condition proposed by Cao *et al.* [26] sets the threshold on the species instead of the propensities. That is, $\Delta_\tau X_i \leq \max\{\epsilon_i x_i, 1\}$, where $\epsilon_i = \epsilon/g_i$ and g_i is equal to the order of the reaction in which S_i is participating (see [26] for details). As $\Delta_\tau X_i = \sum_{j=1}^{m} \sigma_j \nu_{ij}$ with $\sigma_j \sim \mathcal{P}(a_j(\mathbf{x}), \tau)$, we can estimate the average and the variance of $\Delta_\tau X_i$ using the properties of the Poisson distribution

$$\begin{aligned} \langle \Delta_\tau X_i \rangle &= \sum_{j=1}^{m} \nu_{ij}[a_j(\mathbf{x}), \tau] \\ \mathrm{var}\{\Delta_\tau X_i\} &= \sum_{j=1}^{m} \nu_{ij}^2[a_j(\mathbf{x}), \tau]. \end{aligned} \tag{1.17}$$

The above bound on $\Delta_\tau X_i$ can be considered satisfied if it is simultaneously satisfied by the absolute mean and the standard deviation of $\Delta_\tau X_i$

$$|\langle \Delta_\tau X_i \rangle| \leq \max\{\epsilon_i x_i, 1\}, \quad \sqrt{\mathrm{var}\{\Delta_\tau X_i\}} \leq \max\{\epsilon_i x_i, 1\}. \tag{1.18}$$

```
// Initialization
t ← 0;
X ← Initial conditions ;
ϵ ← precision ;
// Main loop
while t < T do
    // Evaluation of the propensities
    for j ← 1 to m do
        a_j ← propensity for R_j with X(t) ; // According to Table 1.1
    end
    τ ← time leap according to Eq. 1.19
    for j ← 1 to m do
        Draw a random number σ_j ∼ P(a_j τ);
        for i ← 1 to n do
            X_i ← X_i + σ_j ν_ij ;                    // Reaction execution
        end
    end
    t ← t + τ ;
end
```

Algorithm 3: Pseudo-code for the τ-leaping simulation. The time is t, the state variable $\mathbf{X}$. The simulation is performed until time T.

This gives two inequalities for τ that need to hold for all $i \in \{1, \ldots, n\}$

$$\tau \leq \frac{\max\{\epsilon_i x_i, 1\}}{\sum_{j=1}^{m} \nu_{ij} a_j(\mathbf{x})}, \quad \tau \leq \frac{\max\{\epsilon_i x_i, 1\}^2}{\sum_{j=1}^{m} \nu_{ij}^2 a_j(\mathbf{x})}. \tag{1.19}$$

As for ODE integrators, more sophisticated algorithms have been proposed to better simulate stiff reaction systems. An improvement is to replace the explicit update rule Eq. 1.16 with an implicit one [27]

$$\mathbf{X}(t+\tau) = \mathbf{x} + \sum_{j=1}^{m} \left[\sigma_j - a_j(\mathbf{x})\tau + a_j(\mathbf{X}(t+\tau))\tau\right] \boldsymbol{\nu}_j \quad \text{with } \sigma_j \sim \mathcal{P}(a_j(\mathbf{x})\tau). \tag{1.20}$$

In this expression, the mean of the Poisson distribution $\langle \mathcal{P}(a_j(\mathbf{x})\tau) \rangle = a_j(\mathbf{x})\tau$ is subtracted out and replaced by its values at the later time point $t + \tau$, but the variance has been left unchanged. Once the Poisson random numbers σ_j have been drawn, the equation is solved for $\mathbf{X}(t+\tau)$ using standard numerical techniques, such as Newton methods. This implicit update rule allows larger time steps for a stiff system without affecting accuracy.

1.2.6 Modified τ-leaping Procedure

Even if the conditions Eq. (1.19) for the leap size τ are satisfied, a Poisson random variable can realize arbitrarily large values and the update rule can thus lead to unphysical, negative copy numbers. A simple method is to reject any proposal leap that results in negative copy numbers. However, if rejections are too frequent, the method loses its efficiency. As a variation of the initial algorithm, the binomial τ-leaping method has been proposed to avoid negative molecular populations [28]. This method draws random numbers from a binomial distribution instead of a Poisson one. To obtain a valid approximation, some restrictions need to be applied. In particular, in situations where multiple reactions share common reactants, the leap conditions that need to be checked turn out be rather complex.

Another approach, that establishes a link to exact Monte Carlo methods, was proposed by Cao *et al.* [29]. The first step is to recognize that negative values of a consumed reactant are likely to arise when the population is already small. The second step is to determine the reactions, called *critical* ones, in which reactants may be depleted. That classification can be made by setting a threshold n_c (typically between 2 and 20) on the number of times the reaction R_j can occur. That is, reaction R_j is critical if

$$L_j = \min_{\substack{i \in \{1,\dots,n\} \\ \nu_{ij} < 0}} \left\lceil \frac{x_i}{|\nu_{ij}|} \right\rceil < n_c \quad \text{and } a_j > 0. \tag{1.21}$$

In this context, the critical reactions R_j are likely to generate negative copy numbers and should therefore be simulated with the exact Gillespie algorithm. Practically, the system is decoupled. That is, according to equation (1.19), a value τ' is chosen for the noncritical reaction and in parallel, a value τ'' is chosen for the critical reactions in the same way as the direct method of the Gillespie algorithm (Eq. (1.14)). Then, the smallest of τ' and τ'' is used to advance the system and the noncritical reaction are executed with random Poisson occurrence and, only in the case $\tau'' < \tau'$, one critical reaction is executed (chosen according to equation (1.15) restricted to the critical reactions). Algorithm 4 is a pseudo-code of this method. With this distinction, the critical reactions can only occur once during a time step and therefore reactants can never reach negative values. For large values of n_c, all reactions eventually become critical and this method converges to the Gillespie algorithm using the direct method.

```
// Initialization
t ← 0 ;
X ← Initial conditions ;
ε ← precision ;
n_c ← threshold for critical reactions ;
// Main loop
while t < T do
    a_0 ← 0 ;                                   // Evaluation of the propensities
    for j ← 1 to m do
        a_j ← propensity for R_j with X(t) ;    // According to Table 1.1
        a_0 ← a_0 + a_j ;
    end
    R_c ← critical reactions according to Eq. (1.21) τ' ← time leap according to
    Eq. (1.19) for R_j ∉ R_c a_0^c ← sum of a_{j_c} for critical reactions ;
    Draw a random number r_1 ∼ U[0,1], τ'' ← -ln(r_1)/a_0^c ;
    if τ' < τ'' then  // Case where no critical reaction occurs during
    τ'
        τ ← τ' ;
        for j s.t. R_j ∉ R_c do
            Draw a random number σ_j ∼ P(a_j τ);
            for i ← 1 to n do
                X_i ← X_i + σ_j ν_ij ;      // Noncritical reaction execution
            end
        end
        t ← t + τ ;
    else          // Case where a critical reaction occurs before τ'
        τ ← τ'' ;
        for j s.t. R_j ∉ R_c do
            Draw a random number σ_j ∼ P(a_j τ);
            for i ← 1 to n do
                X_i ← X_i + σ_j ν_ij ;      // Noncritical reaction execution
            end
        end
        // Critical reaction selection (Eq. 1.13)
        Draw a random number r_2 ∼ U[0,1], R ← a_0^c r_2 ;
        for j s.t. R_j ∈ R_c do
            R ← R − a_j ;
            if R ≤ 0 then
                J ← j ;                         // J is the chosen reaction
                Break ;
            end
        end
        X ← X + ν_J ;                   // Critical reaction execution
        t ← t + τ ;
    end
end
```

Algorithm 4: Pseudo-code for the modified τ-leaping procedure. The time is t, the state variable $\mathbf{X}$. The simulation is performed until time T.

1.2.7 Hybrid Simulation Algorithms

For reaction systems with species fluctuating at different orders of magnitude, the difference of time scales between the reactions is so large that one encounters the problem of stiffness. In deterministic simulations this problem has been circumvented with, for instance, the Michaelis-Menten approximation in the case of enzymatic reactions. There, the fast reactions are approximated by their steady-state value. In stochastic simulations such stiffness implies that most computational time is dedicated to these fast reactions – dramatically slowing down the simulations.

The approach similar to the Michaelis-Menten approximation has also been proposed for stochastic simulations. Such hybrid simulations was first introduced by Haseltine *et al.* [30] and in parallel by Rao *et al.* [31]. The reaction system is partitioned into fast reactions and slow ones under the condition that species involved in fast reactions should have large copy numbers. The partition can be dynamically adjusted, depending on the evolution of the species, but a significant difference between the reaction rates of the two classes should exist otherwise the system may not be appropriate for hybrid simulations.

Even if the details differ from a method to another, the slow reactions are usually simulated using Gillespie's direct method. The way of calculating the propensities depends on the assumptions for the fast reactions. In particular, Haseltine *et al.* [30] propose a Langevin approximation (see below), or an approximation based on the ODE-based reaction rate equation. The quasi-steady-state approach in [31] generates the values for the species through the probability distribution at the steady state. Even though the fluctuations of the fast species are not exactly reproduced, the accuracy of this hybrid algorithm has been demonstrated to be high for a proper partitioning [30, 31].

The main drawback of both methods above lies in the fact that fast reactions including species with low copy numbers cannot be in the fast partition. Cao *et al.* [32] overcame that problem by partitioning the species along with the reactions. Their work led to the *slow-scale stochastic simulation algorithm* (ssSSA) [33] that is currently the most rigorous framework for the hybrid simulation of stochastic reaction systems. As the implementation of the ssSSA is rather involved, the reader is referred to the original article for implementation details. For the discussed case studies, Cao *et al.* [34] report an increase in simulation speed over exact methods of almost three orders of magnitude with no perceptible loss of accuracy.

1.2.8 Langevin Approximation

In the light of τ-leaping, often an additional assumption can be made for reaction systems. More specifically, for systems with a large quantity of molecules a significant number of reactions occur during a single time step. For the following Langevin approximation, not only the time step τ needs to be small enough to satisfy the above τ-leaping conditions, but the expected number of firings for each reaction R_j during τ should also be large. That is to say $a_j(\mathbf{x})\tau \gg 1$ for all $j \in \{1, \ldots, m\}$. Under this condition, the Poisson random variable in equation (1.16) can well be approximated by a Gaussian random variable with both the mean and the variance equal to $a_j(\mathbf{x})\tau$. Equation (1.16) then reads

$$\mathbf{X}(t+\tau) = \mathbf{x} + \sum_{j=1}^{m} \sigma_j \boldsymbol{\nu}_j \quad \text{with } \sigma_j \sim \mathcal{N}(a_j(\mathbf{x})\tau, a_j(\mathbf{x})\tau). \tag{1.22}$$

Notice that the integer random variable is now replaced by a real normal random variable and therefore the state variables X_i are now real numbers, which is only appropriate for large copy numbers.

Using the property of the Gaussian distribution $\mathcal{N}(m, \sigma^2) = m + \sigma\mathcal{N}(0, 1)$, the equation can be further simplified to obtain the *chemical Langevin equation* (CLE)

$$\mathbf{X}(t+\tau) = \mathbf{x} + \sum_{j=1}^{m} \boldsymbol{\nu}_j a_j(\mathbf{x})\tau + \sum_{j=1}^{m} \boldsymbol{\nu}_j \sqrt{a_j(\mathbf{x})\tau}\, \Delta N_j, \tag{1.23}$$

with $\Delta N_j \sim \mathcal{N}(0, 1)$. This equation is valid under the two assumptions that (1) no propensities function changes its value significantly during τ, yet (2) every reaction occurs multiple times during one time step τ. It is usually possible to find a τ that fulfills these criteria if the populations of all species are large. If not, the chemical Langevin equation cannot be used to simulate the system. The CLE can also be written as a stochastic differential equation (SDE) when the increment τ is taken to be infinitesimal

$$\mathrm{d}\mathbf{X} = \sum_{j=1}^{m} \boldsymbol{\nu}_j a_j(\mathbf{X}(t))\mathrm{d}t + \sum_{j=1}^{m} \boldsymbol{\nu}_j \sqrt{a_j(\mathbf{X}(t))}\mathrm{d}W_j, \tag{1.24}$$

with $\mathrm{d}W \sim \mathcal{N}(0, \mathrm{d}t)$, the increments of a Wiener process $W(t)$. Different algorithms can be found to integrate SDEs [35], the simplest thereof is the *Euler-Maruyama* scheme that is equivalent to (1.23) for a fixed step size τ.

1.3 Robustness in Systems Biology

In general terms, robustness is the ability of a system to maintain its function or structure despite external or internal perturbations [36]. Robustness has to be understood as a relative word: when one talks about a robust system, one has to state which function of the organism is robust and also to refer to which type of perturbations the function is insensitive. In the context of Systems Biology, a system function can be associated with a particular mode of operation of a system. A system function for a circadian oscillator for instance, can be that it exhibits stable oscillations with a period around 24h and amplitude within a predetermined range. This example already indicates that, in general, a system's function requires multiple system characteristics to be left invariant, e.g., amplitude and period of oscillations. The perturbations a system can face are perturbations in parameters, network structure that comprises mutations [37, 38], molecule copy number (i.e. molecular noise [39, 40, 41, 42]), external concentrations, etc. One should note that variations in parameters could also account for fluctuations of environmental factors such as temperature [43], pH-value or ionic strength [44].

Robustness has been put forward in engineering as an important design criterion for any system and has also been widely observed in Biology as an intrinsic property of many systems. This feature is probably more important in biological systems than in engineered ones as the environment can be controlled to a lesser extent. Indeed, there is strong evidence that natural selection is favoring robust biological systems [45]. Many properties of biochemical systems show some robustness to the different types of perturbations described above [46, 47, 48, 49, 50, 51, 52]. If robustness has be observed and described, the quantification and analysis of robustness in biochemical systems is still in its infancy.

In the field of complex systems, first attempts have been made to formalize the notion of robustness of a complex system [53, 54, 47, 49]. Although the notion of robustness already exists in control engineering, its definition is too restrictive for biological systems. The function to be maintained in robust control is normally stability, while for biology this may also be the case but higher-level system functions could be involved. For instance, a particular system function for a metabolic network might be a predetermined biomass yield. While changing nutrient supply it might be necessary to control the metabolic network from one stable operating point to another. This could be

achieved by destabilizing the original operating point. Thus, stability is not always equivalent to robustness but is only an instance of the latter.

In this thesis, I will mainly focus on the robustness of small biochemical models to parameter variation and molecular noise. As discussed above, the behavior of a model depends on its parameters. The study of robustness in this context is to answer the question *"How changes of parameter values affect the behavior of the systems?"*. I will now review the two analysis trends at this level: global methods, which assess the volume in parameter space that is compliant with the proper functioning of the system, and local methods, which in contrast determine robustness for a given parameter vector of the studied model.

1.3.1 Global Methods for Robustness Analysis

When focusing on robustness to changes in biochemical parameters that define system behavior, a biological system's robustness is a reflection of the topology and size of its viable space, i.e. the volume, using the proper normalization, where parameters generate a behavior of interest [55, 56]. Global methods try to characterize this volume and therefore the first question we ask, is *"How many parameter combinations allow a system's desired behavior?"*. A small viable volume forces a precise tuning of biochemical parameters. On the contrary, a large viable volume allows the system to successfully face changes in environmental conditions, because its parameters can adapt, sometimes by orders of magnitude, without impairing its function. Hence, robustness is associated with larger viable volumes.

The second question is the geometry of the viable space which plays another important role in a system's robustness. It has been widely observed that the possible range of values in the viable space is very different for each parameter, a property called sloppiness [57]. Viable volumes with irregular geometries are more prone to be left with a little variation in the stiff direction. Therefore, geometries which permit moderate fluctuations in any parameter direction without leaving the viable volume are more robustness.

In evolutionary terms, different ways of performing the same function – for instance, by conserved pathways with homologous yet different proteins [46] – can be traced back to a common ancestor and are thus reachable from each other in an evolutionary tree [45]. A connected viable volume improves the system's evolvability and allows neutral evolutionary trajectories that may

drive the system towards viable parameter points with high local robustness. Therefore, the robustness of a biological system is reflected in the geometry and size of its viable space.

Another approach that is at the edge of local methods is bifurcation analysis [58, 59, 60]. This type of methods characterizes how qualitative model properties, such as stability of steady-states, change as one of the model parameter is varied while others are kept constant.

1.3.2 Local Methods for Robustness Analysis

Most robustness analyses in literature are based on local methods [47, 48, 61, 62, 41, 42]. Local methods can be split in different classes, but they are all based on the same principle: their 'local' denomination meaning that they all analyze a specific model with a single parameter vector.

The first class of local methods comprises analyses used in control theory. These methods mainly use sensitivity analysis of a system function to infinitesimal parameter variations. The function can be a given component concentration at steady state [51, 63] or the period and phase of an oscillator [62, 61, 50]. The frequency response has also been analyzed with this method. [64]. Temperature compensation analyses can be comprised in this category as they use infinitesimal derivatives [43, 65, 66]. Usually, the specific properties are written as a function of the equations of the model and differentiated by the parameters and the absolute or relative value is taken as a robustness measure. The second class of local methods uses small finite variation [48] or random perturbations around the given parameter vector [67] to assess robustness. In this case, a sufficient number of random perturbations must be analyzed to obtain a statistically significant measure. Another class comprises methods that evaluate the effect of the molecular noise on either a steady state value [68, 69] or on the period of an oscillator [41, 42]. These methods rely on stochastic simulations of the system using either Gillespie's algorithm [15] or approximate method such as τ-leaping or Langevin simulations [10]. Other methods can be considered as local, but I will halt the list to discuss other properties linked to robustness.

1.3.3 Robustness and Parameter Identifiability

Besides its long-term scientific relevance as a design principle, the study of robustness is linked to the issues of quantitative biology with parameter identi-

fiability and model discrimination. The link between robustness and parameter fitting in Systems Biology becomes clear when instantiating the corresponding questions they want to answer: global robustness analyses answer the question *"How much variation in the parameters can the system accept while maintaining its output value?"*, whereas parameter fitting answers *"What is the parameter vector that can fit the measured output value?"*. The main problem of parameter fitting is identifiability: it is widely observed that measures cannot constrain the parameter space in some directions [57]. Today's common practice is to ignore this uncertainty and for instance publish models with single point estimates for their parameters. In this regard robustness analysis offers a possible solution for this controversial fine-tuning of biological models toward interval computation yielding parameter regions that are consistent with experimental data.

Such observations about identifiability (or lack of) raise the possibility that local robustness itself could be used to discriminate between models [46, 47]. On one hand, difference in robustness could be used to design new experiments that test specifically this criterion. Such back and forth interactions between model analysis and experiments are a key element to improve models. On the other hand, in the absence of other experimental criteria, a model (or a specific parameter vector) would be judged superior if it is more robust than other models to some class of perturbations [50].

1.4 Biological Oscillations and Circadian Clocks

When studying robustness in biology, several case studies are recurrently seen in the literature. In the context of adaptation and maintenance of a steady state, the chemotaxis capacity of *Escherichia coli* [70] or the heat shock response to prevent protein denaturation [51, 71] have been extensively modeled and analyzed. For the study of biological oscillations [72], two types of systems have been mainly investigated: first, the mitotic cycle [73] which leads to the study of the cell cycle [74]. The second type comprises circadian clocks in different organisms ranging from the cyanobacteria to the *Drosophila* and mammals [75]. I will now focus on the later and explain its role with an emphasis on the mechanisms of the cyanobacterial clock.

1.4.1 Role of the Circadian Clock

Most living organisms exhibit a circadian rhythm with a 24-hour periodicity, synchronized with the alternation of day and night. It is ubiquitous and governs many physiological functions of the organism. Circadian rhythms are of special interest because everybody feels the effect of its internal clock every day, and even more when suffering of jet-lag. The circadian clock is also linked to the cell cycle and drugs, especially in cancer treatment, have different effect depending on the hour of intake [76]. Moreover, circadian rhythms have been found in almost all eukaryotic organisms and even in some bacteria. This prevalence is even more striking as the topology of the systems and some proteins and genes have been evolutionary conserved.

Three major properties are used to define a circadian clock. First, like other biological oscillations, circadian rhythms have an endogenous nature and thus persist under constant environmental conditions: the free-running period in continuous darkness is close to 24 hours. The second property of the circadian clocks is its ability to be entrained by light. Two different theories explain the effect of light: either the transition from dark to light triggers some mechanism that shifts the cycle to maintain it in phase with light, or the effect is over the entire exposure to light. For the latter, the system is working in a different regime during the day than in darkness. But the important aspect is that the circadian clock is able to adapt to the daylight phase within a few cycles. The third characteristic of circadian clocks is their relative insensitivity to temperature changes. Normally, the rate of an isolated chemical reaction doubles when the temperature increases by ten degrees. On the contrary, the circadian rhythms are temperature compensated, meaning that the period remains constant even with temperature changes. This property is nearly unique among biological rhythms and therefore focuses interest in the modeling community [43]. These properties are useful for modeling and robustness analysis as they provide well-defined criteria.

1.4.2 The Cyanobacterial Circadian Clock

Cyanobacteria, from Greek *cyano* which means blue, are a type of bacteria that obtain their energy through photosynthesis. They account for about a quarter of Earth's photosynthetic productivity and are critical in the food chain of many ecosystems. For example, the tiny marine cyanobacterium *Prochlorococcus* was discovered in 1986 and accounts for more than half of the photosynthesis of the open ocean [77]. Being a major component of phytoplankton, it

forms the basis of the ecological pyramid of the aquatic ecosystems. Ancestors of cyanobacteria had probably played a critical role in the oxygenation of the Earth's atmosphere about two and a half billion years ago.

Besides their photosynthesis activities, cyanobacteria are the only group of organisms able to reduce nitrogen and carbon in aerobic conditions. This activity is vital for many higher organisms, such as plants, that are not able to use atmospheric nitrogen. For example, they serve as natural rice paddy fertilizer and are essential in many other crop cultures. Photosynthesis and nitrogen fixation, but also other primary metabolic activities such as vitamin and cofactor biosynthesis, allowed cyanobacteria to inhabit practically every environment and also made them common participants in symbiotic associations.

Knowing that photosynthesis and nitrogen fixation are two incompatible mechanisms raises the question of how cyanobacteria can balance both activities without compartmentalization. Circadian rhythm was not proposed as a control mechanism as it was expected that an organism that could grow and divide at a rate faster than 24 hours would not be able to maintain a circadian regulatory regime. This conviction came from the fact that eukaryotic models for circadian cycles are based upon intercellular communication and nuclear sequestration of key clock components. But physiological studies regarding the temporal separation of cyanobacterial nitrogen fixation and oxygenic photosynthesis suggested that cyanobacteria have endogenous timing mechanisms [78].

In 1990, Huang *et al.* experimentally showed that the cyanobacterial strain *Synechococcus* sp. RF-1 has an internal clock that fulfills all three properties for a circadian clock (sustained oscillations with a period near 24 hours in constant conditions, entrainment by light and temperature compensation). Cyanobacteria are the simplest organisms that display circadian rhythms and therefore provide a model system for the circadian clock. To determine the underlying genetic and molecular basis for cyanobacterial circadian rhythms, experimentalists used the cyanobacterial *Synechococcus elongatus* PCC 7942 as this strain was more suitable for high throughput experiments [78].

Note that *S. elongatus* does not fix nitrogen (and may not require a temporal separation), but it still has a circadian clock which provides a selective growth advantage [79]: mutant strains with circadian oscillators of different period lengths were put into direct competition with one another under differing light-dark (LD) cycles. The strains with endogenous circadian period

lengths best matching the period of the LD cycle prevailed in the competition. For example, when two strains were mixed in nearly equal proportions and grown together, the period matching strain completely overtook its competitor in about seven generations [79]. These experiments show that the circadian clock is advantageous for cyanobacteria beyond the temporal separation of the photosynthesis and nitrogen fixation processes.

The Kai Proteins

The molecular mechanism of a cyanobacterial circadian clock was first reported in *Synechococcus elongatus* PCC 7942 in 1998 by Ishiura *et al.* [80]. The *S. elongatus* clock comprises the products of at least three genes, *kaiA*, *kaiB*, and *kaiC*, named after the Japanese word *kaiten* for cycle. The three corresponding proteins, KaiA, KaiB and KaiC, form the core of the machinery for generating and maintaining rhythms with a free-running period of approximately 24-25 hours. Mutations in any *kai* gene can result in strains with different period lengths and deletion of any *kai* gene results in arrhythmic strains.

Other genes may be involved in the control of the clock and its capacity to be entrained by external factors. However, the Kai proteins are sufficient to maintain the cyclic activity without any gene expression as it was first shown by Nakajima *et al.* in 2005. For this experiment, three Kai proteins were purified and mixed them *in vitro* with ATP and Mg^{2+} to reconstitute the cyanobacterial clock [81]. The *in vitro* oscillator showed the two main features of a circadian clock, i.e., stable oscillation with a free-running period around 24h and temperature compensation. After this experiment, a large community showed interest in understanding [82, 83, 84, 85, 86] and modeling the cyanobacterial circadian clock [87, 88, 89, 90, 91, 92].

The central protein of the clock, and the most intriguing one, is KaiC. It forms hexamer in presence of ATP and its monomeric form is a duplicate version of a recA/dnaB-like gene [93]. RecA is a DNA recombinase and DnaB is a DNA helicase and the similarity with KaiC could imply that KaiC might also act upon DNA, which was indeed shown in [94]. The two domains of KaiC (CI for N-terminal and CII for the C-terminal) contain shared regions that include Walker A and B motifs involved in ATP binding and hydrolysis [93]. The CII domain of each KaiC monomer is believed to have at least two phosphorylation sites (Serine 431, S) and (Threonine 432, T). The phosphorylation pattern

seems to be a key factor for the Kai cyclic activity. This will be discussed in more details after introducing the two other components KaiA and KaiB.

KaiA is present as a dimer (and can be always considered as such) and binds to the KaiC hexamer [82]. KaiA binds to KaiC at the C-terminal (CII domain) with a supposed stochiometry of one KaiA dimer to one KaiC hexamer. KaiB is reported to act most probably as a tetramer. Detailed studies in [83] showed that KaiB most likely associated with KaiC in form of two dimers in a ring shape that attaches on top of the KaiC hexameric barrel in CII side.

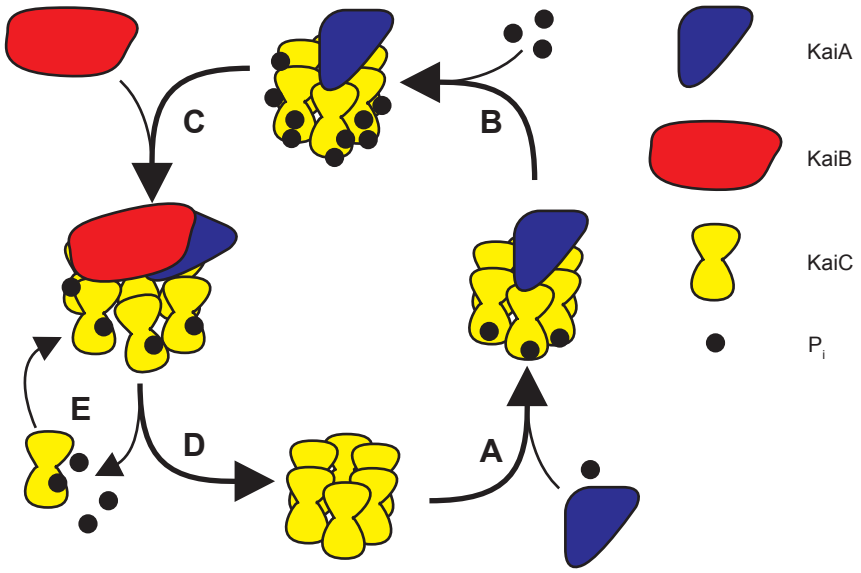

Figure 1.1: Schematic representation of the Kai system. (A) KaiA attach to the unphposphorylated KaiC hexamer and subunits start to be phosphorylated. (B) KaiA enhances phosphorylation until most of the subunits are doubly-phosphorylated. (C) When the hexamer is highly phosphorylated, KaiB binds to it. (D) With KaiB inhibiting KaiA effect, the subunits dephosphorylate. (E) While KaiB is bound, some of the subunits can detach and be exchanged between hexamers.

The output of the *in vitro* system is the phosphorylation state of the KaiC hexamers that oscillates between a low and a high phosphorylation states. The two T and S sites are phosphorylated in a sequence: for each monomer, starting with the unphosphorylated state, first the T site is phosphorylated, which then allows phosphorylation of the S site. Similarly, the dephosphorylation of the S site occurs only after T is dephosphorylated [91]. A substitution of one of those two sites with Alanine (equivalent of an unphosphorylated amino acid) results in a complete loss of rhythmic activity.

These reactions are modulated by the binding of KaiA and KaiB. The binding of KaiA is required for the phosphorylation of the T and S sites on KaiC (Fig. 1.1B): fully phosphorylated KaiC completely dephosphorylates within 24h in the absence of KaiA. KaiB alone does not have a effect on this activity [95]. The effect of KaiB is only seen in the presence of KaiA: when the S sites of the subunits of the hexamer are phosphorylated, KaiB binds and inhibits KaiA action, allowing the hexamer to dephosphorylate [91] (Fig. 1.1C-D). Moreover, KaiB being bound, KaiA is trapped on KaiC and can hardly unbind the complex before KaiB does.

This sequential mechanism seems to be the key of the circadian oscillations. Yet other phenomena have been observed and postulated to enhance oscillations. First, during the dephosphorylation phase, when KaiB is bound to the hexamer, KaiC subunits detach and can be exchanged between two hexamers [95, 92] (Fig. 1.1C). Second, configurational change due to the binding of KaiB has also been hypothesized because molecular dynamics simulations showed that the association of KaiB to KaiC causes the inner channel of the hexamer to increase in diameter. This physical change may weaken the bond between the KaiC monomers which could allow monomer exchange [83]. Third, the close location of the two phosphorylation sites at the interface between the CII lobes of adjacent subunits [96, 97] can be a sign of cooperativity in the phosphorylation activity. Moreover, a third phosphorylation site spatially close to the S site may be interfering [96, 98]. Fourth, the installation of both the kinase and phosphatase activities within the same protein may aid in temperature compensation [86, 85].

Modeling the Cyanobacterial Circadian Clock

Different models have tried to capture the complex mechanism of this system. The first class of models do not distinguish between the two T and S sites and there sequential phosphorylation. In order to generate a cycle, particular mechanisms have to be included in these models. For example, Mehra *et al.* [87] proposed a positive feedback (autophosphorylation activity) along with different stages of dephosphorylation to generate a delay that results in oscillations. An allosteric model was proposed in [88] to recreate a sequence of reactions that results in a cycle. In this first class of models, different hypothesis were tested using robustness as a discrimination criterion [89]. However these models may not be coherent with the experimental rates of binding and unbinding reactions of KaiA and KaiB [86].

The models of the second class make a distinction between the two T and S sites. If the phosphorylation sequence ensures a cycle for the phosphorylation of each KaiC subunits, the issue that these models need to solve is the synchronization of the different subunits. Rust *et al.* [90] proposed that KaiA sequestration by the KaiB-KaiC complex acts as a feedback. Mori *et al.* [92] used a stochastic model to show that monomer exchange is a necessity to maintain oscillations. Similarly, Emberly and Wingreen proposed a model where the hexamers form clusters when highly phosphorylated [99]. Finally, Eguchi *et al.* based the synchronization mechanism of their model on a conformational change in addition of monomer exchange [100, 101].

These models try to properly capture the complexity of the interactions between these three proteins, but for the moment, none of the above hypotheses have been validated and the following questions are still open. First, the role of the hexamer and the possible cooperativity between the subunits is not clear. Second, the slow reactions that set the pace of the 24-hour clock are not always consistent with experimental rates. Third, the synchronization could be dependent on different mechanisms or their combination. To test and discriminate these models, atop of the experimental data, a few robustness criteria could be used. For example, the temperature compensation is a key property. Moreover, the system is rather insensitive to variations of the concentration of the different Kai proteins.

1.4.3 The Circadian Clocks in Eukaryotes

The circadian clocks in eukaryotes differ fundamentally from the cyanobacterial one as they include a transcription step. In all eukaryotic clocks, the central mechanism to produce oscillations is a negative feedback that occurs through a protein that inhibits its own expression directly, or indirectly [102, 103, 75].

In the first model of the circadian clock in *Drosophila*, Goldbeter proposed that the protein Period (PER) inhibits its own expression [104]. The necessary delay to generate oscillations is induced by two stages of phosphorylation of PER that are required for the protein to enter the nucleus. Even if this simple model is enough to explain the periodicity of PER expression, more complex models have been published to include newly discovered proteins [105, 106, 107]. PER associates with protein Timeless (TIM) to act as a complex inhibiting their own expression. In contrast to PER, TIM degradation is enhanced by light, allowing the clock to be entrained by the day and night alternation [105].

Further studies showed that other proteins Cycle (CYC) and Clock (CLK) are also interacting with PER and TIM. In fact, the CYC-CLK complex is promoting PER and TIM expressions and is at the same time degraded by the PER-TIM complex forming the negative feedback [106, 107].

The presence of multiple and interlocked feedback loops is a common feature in eukaryotic circadian clocks [102]. In *Neurospora*, the White Collar proteins are also phosphorylated and form complexes with the protein Frequency which controls the expression of the formers [75, 108]. In *Arabidopsis*, similar pattern are observed with the proteins CCA1 (Circadian Clock Associated 1) which is phosphorylated before inhibiting its own expression and regulating TOC (Timing Of CAB expression) [109]. Finally, in mammals, a multitude of parallel loops exist: the PER protein is found in three different copies that have slightly different regulation [103, 75]. In parallel to the PER/CRY loop, a loop with BMAL1/CLOCK is formed and interferes with PER/CRY expression [60, 110, 103]. If such profusion of loops has been argued to enhance robustness [102], it has never been properly shown.

1.4.4 Other Biological Cycles

Although feedbacks have been widely studied in the context of circadian clocks, they are also present in many other systems [111, 2]. Even before the protein regulation was known, Goodwin proposed a model for oscillators based on a unique negative feedback that revealed to be a good generic model for circadian clocks [112].

Feedbacks are also key elements for oscillations of the mitotic cycle [113, 73, 114]. The core of the mitotic oscillator is the Cdc2-Cyclin B complex. Cdc2 activates the anaphase-promoting complex (APC); the APC then promotes Cyclin degradation and resets Cdc2 to its inactive state. The positive feedback, critical to form a relaxation oscillator, generates much sharper peaks. This mechanism is actually based on two positive feedback loops: active Cdc2-cyclin B is stimulating its activator Cdc25 and at the same time inactivating its inhibitors Wee1 and Myt1.

Other examples of biological cycles include the p53/Mdm2 oscillator which induces oscillations of p53 in response to stress [115] or the Delta/Notch oscillator involved in somitogenesis [116]. All these different ways to obtain oscillations influence the properties and robustness of the system [50, 117].

1.5 Brief Introduction to Synthetic Biology

In this section, I will give a small review of Synthetic Biology, a new field that developed in parallel to Systems Biology over the last decade. The ideas about altering the genome were already present in the 90s where genetically modified organisms were produced. At this time, the goal was to make an organism produce something that it was not supposed to produce: insulin for a bacteria or pesticides for crops. These manipulations were not well controlled as the incorporated genes were constantly expressed.

The better understanding of the genome that came with gene sequencing allowed more subtle manipulations. The discovery of some promoters and inhibitors permitted to implement regulations between the artificial genes. With such tools, ideas about synthetic circuits, a biological equivalence of logical circuits found in electronics, rose. In 2000, two papers were published and can be considered as milestones in the recognition of Synthetic Biology. First, Elowitz and Leibler build a synthetic oscillator in bacteria made of three genes forming a repression loop [118]. Second, the group from J. Collins constructed a toggle switch that comprises two self-repressing genes [119].

Since then many promoters and inhibitors were identified or engineered in several model organisms (bacteria, yeast, *C. elegans* and mammalian cells). A variety of relatively small synthetic gene networks [120] have been successfully implemented and characterized. These include oscillators [118, 121, 122, 123], toggle switches [119, 124], and intercellular sender/receiver or quorum sensing communication systems [125, 126, 127]. Some recent projects have successfully integrated a few of these 'standard' modules and have also interfaced them with endogenous pathways to program more sophisticated behaviors [128, 129, 130, 131, 132, 133].

However, understanding how to integrate a significant number of modules to perform complex tasks remains one of the most important challenges facing synthetic biology [120]. In this sense, the results in Systems Biology, using a backwards engineering approach, can be also applied to the Synthetic Biology as forward engineering. For example, currently, Synthetic Biology is mainly based on trial and error, but the building of large systems will necessitate design principles such as modularity that are present in biological systems. Along with the principles, *in silico* analyses such as the ones developed in Systems Biology will be essential tools for the realization of large and realizable circuits. Specifically, the aspect of robustness as discussed in this thesis could be a critical design criterion [123].

1.6 Thesis Outline

Most of the current works on quantification of robustness use local analysis. In the next chapter, I will explain the shortcomings of the current state of the art methods for such analysis. To overcome these limitations, I developed new notions for a glocal measure of robustness. This novel method explores both global and local aspects: it comprises a broad exploration of the parameter space comparable to some global approaches, but goes beyond current analyses as it comprises a further refinement based on different local measures. The concepts of the method are general and the glocal analysis could be applied to many problems in Systems Biology as shown in the following publication [134]:

- M. Hafner, H. Koeppl, M. Hasler and A. Wagner, *'Glocal' robustness analysis and model discrimination for circadian oscillators*, in PLoS Computational Biology (2009), vol. 5(10), e1000534.

In Chapter 3, I will describe the algorithms developed for my glocal analysis. Different algorithms for parameter fitting can be found in the current literature, but few of them, besides brute force sampling, can cope with interval constraints and provide a region of the parameter space instead of a single result. The first section of this chapter is dedicated to two sampling algorithms that provide a uniform distribution of parameter vectors that fulfilled some systemic properties used for the glocal analysis. A first sampling algorithm was published in [134]. It allows a more efficient sampling than brute force sampling without many adjustments. The next iteration of the sampling algorithm was developed with E. Zamora to overcome shortcomings of the first method and published in [135]:

- E. Zamora-Sillero, M. Hafner, A. Ibig, J. Stelling and A. Wagner, *Efficient characterization of high-dimensional parameter spaces for systems biology*, in BMC Systems Biology (2010), submitted.

This chapter also contains a section dedicated to algorithms for the different local robustness quantifiers used for circadian clock analyses [134]. Each quantifier measures the robustness of a model with respect to perturbations in either the parameters, the concentrations or the molecule copy numbers.

Chapter 4 contains the results of the application of the glocal method. In the first section, I consider two models of the cyanobacterial circadian clock and compare their robustness properties [134]. The results show the importance of a glocal approach for robustness assessment and model comparison: if a single

parameter vector shows good local robustness, this result is not representative of the possible parameter space and alteration of some parameters can strongly decrease the robustness. With these results, a model can be discriminated in favor of the other one.

The second part of the results (Section 4.2) is also devoted to circadian rhythms with a study of the robustness of two models of the *Drosophila* circadian clock with respect to entrainment. Using the glocal approach, this work shows that an additional feedback loop enhances the robustness of the system to parameter variations. The local analysis, based on the phase response curve, gives further insights on the regulations of the different clock components. This work was published as a proceeding paper of the WCSB conference in 2010 [136]:

- M. Hafner, P. Sacré, L. Symul, R. Sepulchre and H. Koeppl, *Multiple feedback loops in circadian cycles: Robustness and entrainment as selection criteria*, in Proceedings of the Seventh International Workshop on Computational Systems Biology (2010), edited by M. Nykter, P. Ruusuvuori, C. Carlberg and O. Yli-Harja, pp. 43-46.

Section 4.3 follows-up on the study of multiple feedbacks in oscillatory systems. I will present the results of an algorithm designed to study the evolution of new structures in biochemical systems. The idea behind this work is to show that evolution of motifs is possible without disturbing the systemic properties of the system. This algorithm was applied to a generic model of the mitotic cycle containing one positive and one negative feedback loop and was published in the proceeding of the FOSBE conference in 2009 [137]:

- M. Hafner, H. Koeppl and A. Wagner, *Evolution of feedback loops in oscillatory systems* in Proceedings of the Third International Conference on Fundations of Systems Biology in Enginnering (2009), pp. 157-160, http://arxiv.org/abs/1003.1231.

In the next part of Chapter 4 (Section 4.4), the second sampling algorithm [135] was applied to the generic model of the mitotic cycle. In this work, we showed that the sampling procedure is able to efficiently find the non-convex parameter space formed by the two feedbacks loops. The results provide an additional answer for the presence of multiple feedback loops in oscillatory systems: the combination of a positive and a negative feedback improves the robustness of the system to parameter changes.

The fifth section of the results (Section 4.5) brings another argument for

the robustness advantage of multiple feedback loops. In collaboration with D. Gonze, we study the effect of molecular noise on a mitotic cycle model. This model, based on a negative feedback loop, shows a strong increase in robustness when a positive feedback loop is included. In order to obtain results that are parameter-independent, we used the glocal method and confirmed the advantage of the positive feedback loop. This work was published in [138]:

- D. Gonze, M. Hafner, *Positive feedbacks contribute to the robustness of the cell cycle with respect to molecular noise*, in Advances in the Theory of Control, Signals and Systems, with Physical Modelling, Lecture Notes in Control and Information Sciences, Vol. 407, (2011) pp. 283-295, Edited by J. Lévine, P. Müllhaupt, Springer, Berlin.

Finally, Section 4.6 is dedicated to Synthetic Biology. In collaboration with M. Miller, we design a large synthetic circuit using robustness analyses from Systems Biology. We found general principles for building a system that can robustly maintain homeostasis regulation. An adaptation of the glocal approach was also used to provide directions for parameter optimization. Some experimental results of the of Ron Weiss' group have also been included in the article [139]:

- M. Milles*, M. Hafner*, E. Sontag, S. Subramanian, P. Purnick, N. Davidsohn and R. Weiss, *Design of a large scale synthetic biological circuit to maintain artificial tissue homeostasis*, in preparation. *equal contribution.

The thesis concludes with a discussion that links the different results under the aspect of evolution and network design. Some perspective and possible follow-up works are also discussed. Finally, in the appendix, some technical details which are not essential for the general understanding are presented to complete some parts of the thesis.

Chapter 2

The Glocal Analysis

In this chapter, I will present my main theoretical contribution: a new method for the study of robustness based on a 'glocal' analysis of the parameter space. As discussed in section 1.3 of the previous chapter, current robustness analyses can be subdivided into global and local methods, both having their limitations. In my glocal analysis, I combine the two complementary approaches and provide a more objective measure of the robustness of a system. My method identifies the region of a high-dimensional parameter space where a circuit displays a specific behavior (either observed or desired). It does so via a Monte Carlo approach that will be discussed in the chapter 3. This global analysis is then supplemented by local analyses, in which circuit robustness to specific perturbations is determined for each of the thousands of parameter vectors sampled in the global analysis.

2.1 Current Methods for Robustness Analysis

In the literature, the methods assessing robustness are either based on the global or the local scale. Global methods characterize the size or volume of a model's parameter space that generates a behavior of interest [55, 56]. In these works, only general criteria for the behavior are taken into account and a boolean function is applied to the parameter space (criteria are fulfilled or not). Another approach is bifurcation analysis [58, 59, 60]. This type of analysis characterizes how qualitative model properties, such as stability of

steady-states, change as model parameters are varied [140]. The structure of a bifurcation diagram can be influenced by variation in parameters that are not considered, which limits this approach. Moreover, if, at certain bifurcation points, the qualitative behavior changes radically, the quantitative behavior and the biological effect may not be very relevant: for example, at certain types of Hopf bifurcations, oscillations appears but may be of very small amplitude carrying no information for cellular processes.

In contrast to global methods, local methods analyze how perturbations affect model behavior for one specific parameter vector. Their main limitation is precisely this: they may not reflect model behavior under all possible parameter vectors. Most robustness analyses in literature are local as models are usually published with a single parameter vector. Examples include sensitivity analysis [62], which studies the effect of perturbations for a given parameter set on the model behavior, and its application to circadian oscillators [48, 49, 62]. These methods are usually based on the linearization of a system and therefore hold for variations of only a few percent of the parameter values. Comparative study showed that infinitesimal sensitivity may be insufficient to capture the behaviors of the models [141]. Other works use stochastic simulations to estimate the robustness of a system to molecular noise [41, 42] and results also remain bound to the choice of the parameter vector and can lead to contradictory conclusions [41, 142]. Efforts to extend a local analysis to systematic parameter variations in more than one or two dimensions [59, 60] are often limited by computational cost.

To summarize, global methods assess the volume in parameter space that is compliant with the proper functioning of the system. As all parameter sets are considered as equivalent, the only information that these methods provide is the geometry of the volume in the parameter space for which some criteria are fulfilled. Local methods, in contrast, study the model for a given parameter set and determine its robustness. Local methods are fundamentally biased due to the *a priori* choice of a particular parameter set. As biological constants, such as kinetic rates, are hard to measure precisely and, moreover, may fluctuate *in vivo*, local robustness analyses are not representative and could even be irrelevant.

2.2 'Glocal' Concept

To overcome the limitations of current analyses, I propose a novel 'glocal' (contraction of global and local) method for the quantification of robustness.

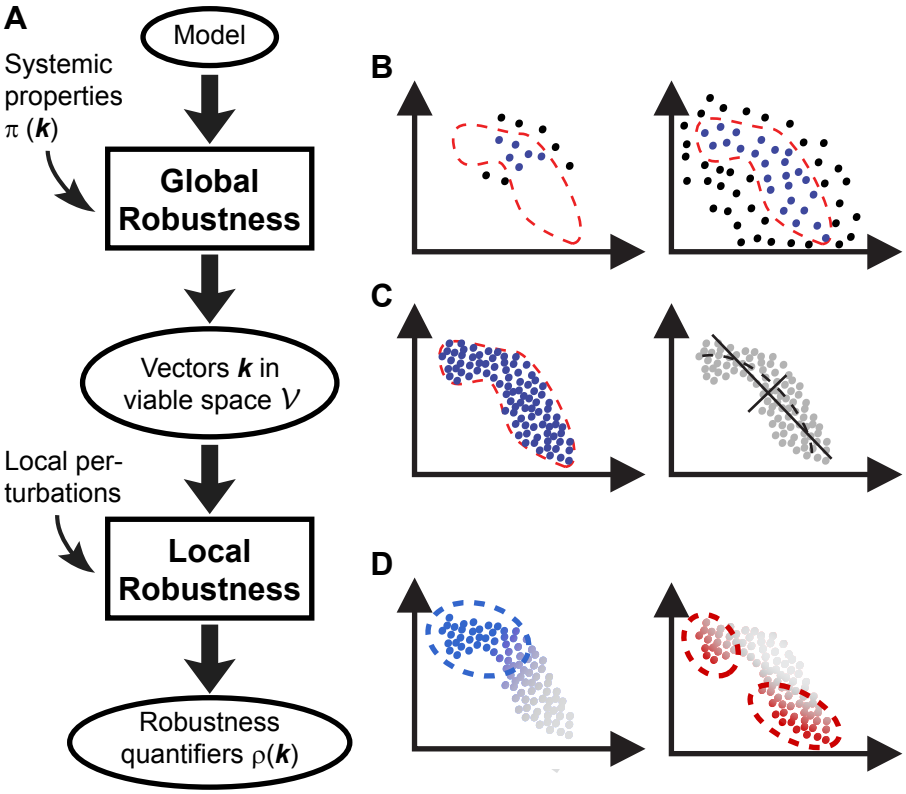

Figure 2.1: Glocal robustness analysis flow for a hypothetical two dimensional parameter space. (A) A model and systemic properties serve as inputs for the global step of the analysis. This global analysis is composed of (B) and (C) and yields viable parameter vectors $\mathbf{k}$ for the model in addition to the size V of the viable space $\mathcal{V}$. Different local perturbations are applied to these parameter vectors (D) in order to quantify their local robustness $\rho(\mathbf{k})$. (B) A sampling algorithm (see *Methods*) is used to find the viable region (enclosed by the red dashed line) in the parameter space. It yields viable parameter vectors (blue circles). (C) A Monte Carlo integration is performed to estimate the volume of the viable region. It yields uniformly parameter vectors (blue points) that can be used for principal component analysis or high-order correlation function (right). (D) Local analyses are performed on all viable parameter vectors and help identify correlations between parameters and local robustness values (color intensity for two different quantifiers: left-blue, and right-red) to provide regions of high robustness in the parameters space.

Conceptually, my method goes beyond global analysis as the region in the parameter space is further quantitatively assessed in terms of local robustness (Figure 2.1A). It is also much more representative than the local methods as the analysis is performed over a region of the parameter space instead of a specific point.

The first stage of the glocal analysis is similar to global methods as it aims to estimate the volume occupied by parameters for which a model yields a specific biological behavior. This behavior could either be based on experimental evidences if one wants to match a model to a biological system or it could be other criteria that fulfill design requirements of a synthetic system. Because such a search becomes very challenging in high-dimensional parameter spaces, the Monte Carlo integration is preceded by an iterative procedure that allows efficient sampling (see Section 3.1). This algorithm differs from parameter fitting in the sense that it provides an ensemble of parameter sets uniformly distributed in the region where the model is consistent with the specified features instead of a single parameter set (Figure 2.1B-C).

The second stage evaluates the robustness of the model behaviors for each of the previously generated parameter sets to different kinds of perturbations (Figure 2.1D). The perturbations depend on the system studied and could, for example, include concentration perturbations or molecular noise. In general, any property of the system that depends on the parameters can be used for the local analysis. As a consequence, this method allows quantification of the robustness properties over a large region of the parameter space. An advantage in term of computational cost comes from the efficiency of the Monte Carlo integration compared to a grid sampling of the parameter space.

The strength of the glocal method arises from the quantification of the local robustness on the global scale. With these results, correlations between the local robustness and the different parameters can be calculated. Such analyses can answer questions like *"does some regions of the viable parameter space show higher robustness?"* which could be complemented by *"can two different robustness properties be optimized at the same time by changing parameters?"*.

The relevance of this two-scale analysis of the robustness is two-fold. On the one hand, understanding the origin of robustness and fragility provides information for the design of new experiments which may allow discrimination between competing hypothetical mechanisms or models. It may as well lead to new targets for disrupting a system in the context of multidrug therapies. On the other hand, the knowledge on how parameters influence the robustness properties without altering the system is a guide for the design and optimization of synthetic circuits.

2.3 Formalism for Glocal Analysis

A model's behavior is determined by a certain number p of parameters, i.e., the parameter vector $\mathbf{k} \equiv (k_1, \ldots, k_p)^T \in \mathcal{K} \subset \mathbb{R}^p$. Note that the parameter values may be restricted to a subspace $\mathcal{K}$ of $\mathbb{R}^p$ where the boundaries depend on biophysical constraints or the scope of the model. Any robustness analysis needs to quantitatively characterize the system's function that is maintained under perturbations. We do this through a collection of systemic properties $\boldsymbol{\pi}(\mathbf{k})$ that are required to have values within predetermined intervals. Properties can take various forms. In calibrating a model to experimental time course data for instance, one may want to determine parameter regions exhibiting trajectories that stay within a predefined interval around the experimental time series reflecting the uncertainty of data acquisition. Later in this thesis, in the application of the robustness analysis to circadian models for the cyanobacteria, $\boldsymbol{\pi}$ will comprise the period π_T and amplitude π_A of the circadian oscillation of phosphorylated KaiC. Here, to emphasize the generality of the approach, I refer to some general and hypothetical vector of properties $\boldsymbol{\pi}$. I say the model with parameter vector $\mathbf{k}$ maintains its function and preserves $\boldsymbol{\pi}$ if $\boldsymbol{\pi}(\mathbf{k}) \in [\underline{\boldsymbol{\pi}}, \overline{\boldsymbol{\pi}}]$.

Any interval constraint of this kind on a systemic property partitions the parameter space into regions that are viable and those that are not. Augmenting the collection $\boldsymbol{\pi}$ with a new property and its constraint generally causes the viable region to shrink (see Figure 2.2). This semi-quantitative approach is particularly suitable to deal with the ubiquitous uncertainty of biological information as it uses interval constraints rather than optimality criteria as found in optimization techniques such as model calibration. It can leverage principles from interval analysis [143], semi-quantitative reasoning [144] and robust control theory [145]. Moreover, specification languages based on linear temporal logic could be used to define more complex constraints [146, 147].

After defining the properties and their viable intervals, the first step of this method involves the sampling of a large set $\mathcal{S}$ of vectors $\mathbf{k}$. The sampling can span several orders of magnitude for each component depending on the biologically possible values, $\mathcal{K}$ (calculations are made in the decadic logarithmic domain to account for relative variations). Only some viable parameter vectors forming a subset $\mathcal{V} \subset \mathcal{S}$ will generally preserve $\boldsymbol{\pi}$. We sample according to an iterative scheme (Figure 2.1B), where in each step the sampling distribution is adjusted based on the viable set of the previous step (see chapter 3). After the iterative search, a Monte Carlo integration (Figure 2.1C) yields a quantitative measure, V, of the size of the region in the parameter space in

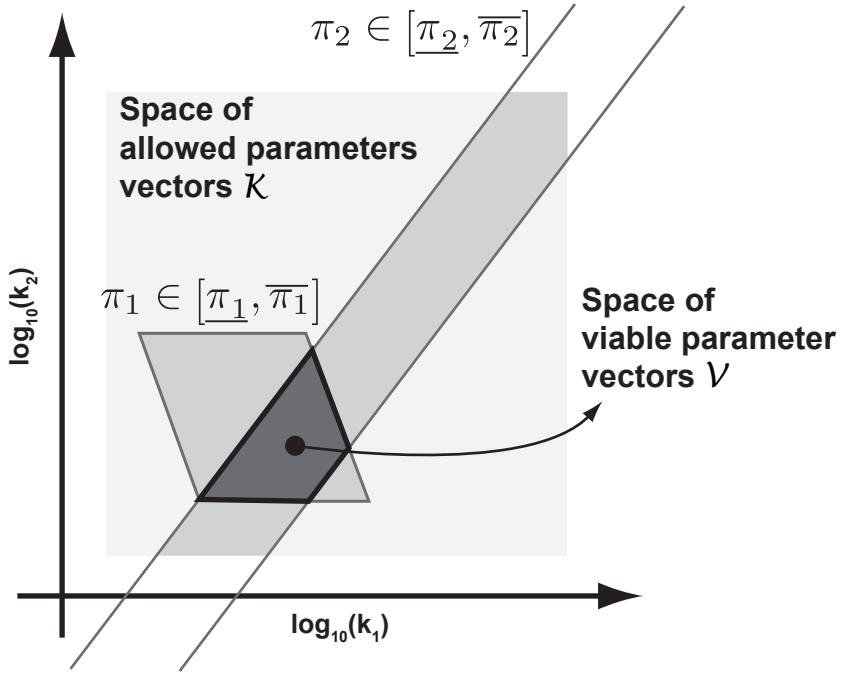

Figure 2.2: Illustration of the global approach based on interval constraints. The a priori sampling range $\mathcal{K}$ (light-gray) and two systemic properties π_1 and π_2 allowed to assume values in predetermined intervals induce constraints in parameter space and partition it into regions that fulfill some of the criteria and those that do not. The viable region $\mathcal{V}$ where all criteria are fulfilled is the intersection of the different regions. The parameter region preserving π_2 is unbounded, accounting for the situation of unidentifiability and indicates the necessity for an a priori sampling range $\mathcal{K}$.

which the systemic properties of the model are within the specified interval. The volume occupied by the set $\mathcal{V}$ provides a first, crude characterization of a model's robustness: the larger this volume is, the more parameters can fluctuate without disrupting the system's functions. With a normalization by the p^{th}-root to account for different numbers of parameters, the normalized viable volume $R = \sqrt[p]{V}$ can aid in model comparison and discrimination. A second measure of the robustness is the shape of this viable region: correlations between different parameters can strongly reduce the robustness of the system to perturbations perpendicular to these correlations. The principal component analysis (PCA) [148] or higher order statistical tools [149] could be used to estimate constraints in the viable parameter space.

The next step of the glocal approach takes advantage of all previously identified viable parameter vectors in order to carry out local robustness anal-

yses (Figure 2.1D). This is done by defining a vector of robustness quantifiers $\boldsymbol{\rho}(\mathbf{k})$ for each $\mathbf{k} \in \mathcal{V}$. This vector includes different assessment $\rho_i(\mathbf{k})$ of the robustness of model properties $\boldsymbol{\pi}$ to particular kinds of perturbations. In order to have an easy and visual comparison, the local robustness quantifiers are normalized to range from zero (minimal robustness) to one (maximal robustness). Concretely, in the study of the cyanobacterial circadian cycle, I will use complementary quantifiers $\rho_i(\mathbf{k})$ that measure the resilience of the models to parameter variations, perturbations of the total number of molecules, and to the effect of the molecular noise.

This mathematical formalism is helpful for further statistical analyses. First order correlations between local robustness values and parameter components can be easily calculated [148]. It gives an insight on how local robustness properties are influenced by parameter values. On one hand it can help further discrimination of parameters or models as it may restrict the viable space: the size of the region with high robustness can also be evaluated as

$$V(\rho_0) = \frac{|\{\mathbf{k} \in \mathcal{V} | \rho(\mathbf{k}) > \rho_0\}|}{|\mathcal{V}|} V \tag{2.1}$$

where $|.|$ is the number of elements in the set. On the other hand, it gives new directions either for experimental discrimination or system optimization such as how to improve the robustness of a system while maintaining its systemic properties. Finally, by performing different local analyses, the region with high overall robustness (average of all local quantifiers) can be found or, alternatively, it can show that the system may not be able to be highly robust against different perturbations at the same time. Such conclusions can only be drawn with a glocal analysis as described here.

Chapter 3

Methods

3.1 Algorithms for Efficient Sampling of the Parameter Space

The glocal analysis relies on a sampling of the parameter space yielding a large number of uniformly distributed parameter vectors in the viable region. Such search is computationally challenging due to the high dimensional parameter spaces, and the lack of prior knowledge about the size or geometry of viable regions. To characterize a viable space, some authors perform a uniform sampling of the whole space to identify regions where a model displays the desired systemic properties [46, 67, 150, 151].

Determining these systemic properties typically involves integration of the model equations, which can become very expensive when done for a high number of samples. Even more fundamentally, the "curse of dimensionality" [152] makes the fraction of the whole parameter space occupied by viable parameters decrease exponentially with increasing dimension, i.e., number of parameters. Therefore, brute force uniform sampling becomes quickly infeasible as model complexity increases. To avoid this problem, following the ideas of Monte Carlo integration [153] with importance sampling, dense sampling should be done in a subset $\mathcal{S}$ of the parameter space much smaller than the possible space $\mathcal{K}$. The construction of a region enclosing the viable space as tightly as possible prior to the Monte Carlo integration is the critical point for the algorithm efficiency and, as a consequence, precision.

I first present an algorithm that explores the parameter space by iterative Gaussian sampling and use principal component analysis (PCA) to optimize the search. Briefly, at every iteration, this method determines the mean value and the covariance matrix of the identified viable points in parameter space to guide further sampling. Although the algorithm is easy to implement and tune, its efficiency depends on the convexity of the viable region and the number of viable vectors obtained at each iteration.

In collaboration with Elias Zamora from University of Zurich, we developed an algorithm that overcomes these limitations. Specifically, it can more efficiently characterize non-convex and poorly connected viable spaces. The algorithm consists of two stages, namely a coarse-grained sampling of the viable space, which in turn delivers starting points for more detailed subsequent local exploration and approximation of the space's geometry. The sampled points also define a domain for subsequent Monte Carlo integration for volume computations. In this algorithm, to optimize the sampling of non-convex viable space, we implemented a stratified sampling [153, p. 412] where the integration is performed on different regions that cover the sampling region.

The proposed sampling approaches are complementary to traditional approaches such as model calibration, but are conceptually different as follow. The structure and parameters of most reported models are underdetermined with respect to the available experimental observations [154]. Moreover, some biochemical models are structurally unidentifiable [155]. That is, even in the presence of arbitrarily abundant and error free data, model parameters that yield the observed behavior cannot be uniquely identified. However, model calibration is designed to find the unique parameter vector that renders the model behavior the closest to the experimentally observed behavior. Consequently, often the single point-estimate for a model parameter, returned by a calibration procedure, may contain little information about the underlying biophysical process. The sampling approach alleviates this implicit degeneracy by making it explicit and instead returns a large number of parameter vectors spread over the whole region consistent with the observed behavior.

3.1.1 PCA Sampling Method

The formalism defined in the previous chapter will be used: a model that involves p parameters has a parameter vector $\mathbf{k} \equiv (k_1, \ldots, k_p)^T \in \mathbb{R}^p$. I choose to work in the decadic logarithmic domain to account for the large range of parameter values in biology and obtain a scale invariant measure. Therefore, in

this chapter, the expression $\mathbf{k}$ should be understood as $\log_{10}(\mathbf{k})$ unless specified. It also means that k_i can take negative values which also generalize the sampling procedures to any parametric models having a Boolean classification criterion. In practice, I will usually restrict $\mathbf{k}$ to the subspace $\mathcal{K}$ of possible values for the studied model. As the scope of a model is limited, it is natural to put some restriction on certain parameters depending on their biophysical significance or *a priori* knowledge. This is also a way to avoid artifacts due to unidentifiability [155].

The PCA sampling method involves an iterative procedure, which I now describe. In each iterative step j, it generates a set $\mathcal{S}^{(j)}$, and identifies the viable subset $\mathcal{V}^{(j)}$. The first set $\mathcal{S}^{(1)}$ is a Monte Carlo sample of the parameter space obtained via a large ($n_{PC} > 10^4$) number of p-dimensional Gaussian random variates, centered on a known viable parameter vector (Figure 3.1A-B). We then determine the viable subset $\mathcal{V}^{(1)}$ of $\mathcal{S}^{(1)}$, which should comprise of the order of 100 to 1000 elements, depending on the dimension of the parameter space. The next step of the procedure consists of a PCA of the viable parameter set $\mathcal{V}^{(1)}$. PCA is a technique to identify dominant linear statistical structure in high-dimensional data sets [148]. We use it here to identify associations among viable parameters that can guide our sampling in subsequent iterations. Specifically, the set $\mathcal{S}^{(2)}$ and subsequent sets are generated from previous parameter sets as follows

$$\mathcal{S}^{(j)} = \left\{ \mathbf{k}_i = \langle \mathcal{V}^{(j-1)} \rangle + \lambda^{(j-1)} \, \boldsymbol{\xi}_i \mid i = 1, \ldots, n_{PC} \right\}, \tag{3.1}$$

for all $j > 1$, where $\langle \mathcal{V}^{(j-1)} \rangle$ stands for the element-wise mean of parameter vectors in the set $\mathcal{V}^{(j-1)}$ and $\boldsymbol{\xi}_i$ is the i-th realization of a p-dimensional Gaussian process with zero mean and covariance matrix $\mathbf{\Sigma}^{(j-1)}$. The size of $\mathcal{S}^{(j)}$, n_{PC} could be adjusted such that the number of viable vectors found at each iteration is in the order of 100 to 1000. The entries $\Sigma_{nm}^{(j-1)}$ are the pairwise covariances of parameters k_n and k_m in the set $\mathcal{V}^{(j-1)}$. We compute this matrix, whose eigenvectors are the principal axes of the set $\mathcal{V}^{(j-1)}$. The real valued factor $\lambda^{(j-1)}$ determines the variance of the j-th Gaussian process by scaling the standard deviations of the distribution along the PCA directions of the $(j-1)$-th iteration (Figure 3.1C-D). In this approach PCA avoids wasting sampling effort on parameter regions where viable parameter vectors are not likely to be found. The procedure is iterated until convergence or until a predefined number of iterations is reached. As described thus far, our procedure serves to identify major axes of viable parameter variation for sampling and the dispersion of the viable parameters along them.

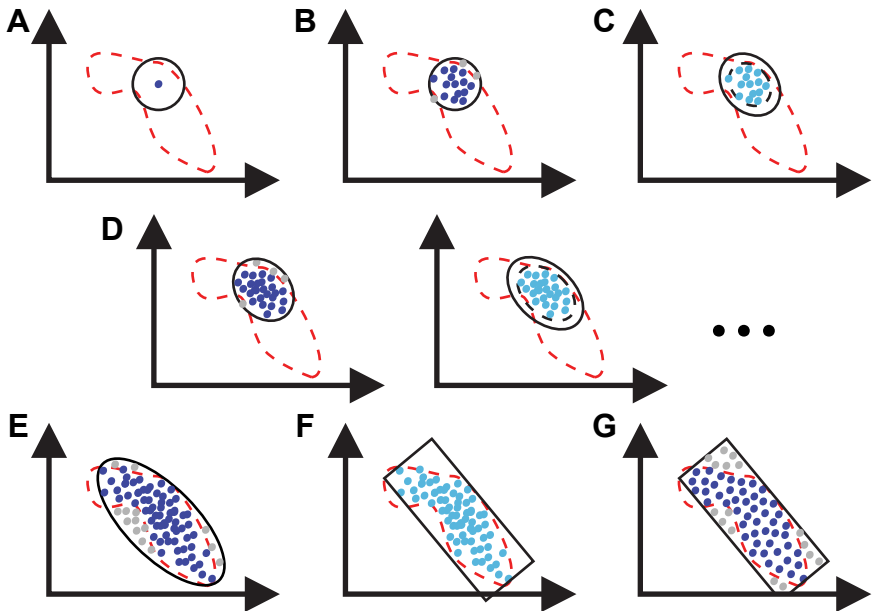

Figure 3.1: PCA sampling method flow for an hypothetical two-dimensional parameter space. (A-E) The iterative Monte Carlo sampling defines the range of the viable parameter space. (A-B) The first sampling iteration uses Gaussian random sampling with independently and identically distributed random variables around a givenparameter vector. The tested parameter vectors are viable (blue points) or not (gray points). (C-D) For subsequent iterative steps, sampling occurs according to the covariance matrix of viable parameters estimated in previous steps (light blue points). The procedure is iterated until convergence or until a predefined number of iterations is reached (E). (F-G) Monte Carlo integration. To estimate the volume in which viable parameter sets occur, we define a hyperbox $\mathcal{B}$ (rectangle in F) that contains all the viable parameters of the last iteration. The region is then uniformly sampled (G).

To establish global measures of robustness, we then perform a Monte Carlo integration (Figure 3.1F). Specifically, we construct a hyperbox $\mathcal{B}$ in parameter space whose axes are parallel to the PCA axes of the last iteration. In each dimension, the limits of this box are defined by the most extreme components of the viable parameters found in the last iteration of sampling along these axes. We then generate a set $\mathcal{S}$ of at least 10^5 parameter vectors sampled uniformly within $\mathcal{B}$. We then define the set of uniformly distributed viable points $\mathcal{V}_{uni} = \{\mathbf{k} \in \mathcal{S} | \boldsymbol{\pi}(\mathbf{k}) \in [\underline{\boldsymbol{\pi}}, \overline{\boldsymbol{\pi}}]\}$. This Monte Carlo integration yields a global measure of robustness for any one model: the viable volume, $V = (|\mathcal{V}_{uni}|/|\mathcal{S}|)(\mathrm{Vol}(\mathcal{B}))$, where $|.|$ denotes the number of elements in a set and $\mathrm{Vol}(\mathcal{B})$ the volume of $\mathcal{B}$. The rationale behind this measure is that with increasing robustness V, a perturbation of a parameter or parameter vector is increasingly likely to generate another viable parameter vector.

To compare models with different number of parameters, we define the normalized viable volume as robustness $R = \sqrt[p]{V}$. The value R represents the average variation in the viable space per parameter axis. Although it is almost certain that the viable range varies among the different axes, R can still be thought of as a parameter robustness of a model. For example, in the log-domain, a value of $R = 0.5$ ($V = 0.5^p$) means that on average we can change every parameter over half an order of magnitude (by around 32%) while preserving $\boldsymbol{\pi} \in [\underline{\boldsymbol{\pi}}, \overline{\boldsymbol{\pi}}]$.

3.1.2 Two-stage Sampling Method

The implementation of the previous algorithm is relatively easy, however it assumes that the viable region is roughly ellipsoidal. The existence of a non-convex viable region reduces the efficiency of the sampling which is critical for high-dimensional space. Moreover, this algorithm may not be able to fully sample disconnected regions. To improve the sampling procedure, in collaboration with Elias Zamora from University of Zurich, we developed a method based on two stages of sampling. First a coarse-grained sampling evaluates the extent of the viable region. Second, multiple expansions of ellipsoids, as in the previous method, are used to cover the viable region.

Adaptive Metropolis Sampling

We next describe the coarse-grained, global exploration of the viable space via an Adaptive Metropolis Monte Carlo sampling (AMC) (Figure 3.2). For this procedure, we have first to introduce a cost function E in addition to the previous formalism. The systemic properties $\boldsymbol{\pi}(\mathbf{k})$ for each parameter vector $\mathbf{k}$ have an associated value

$$E(\boldsymbol{\pi}(\mathbf{k})) = E(\mathbf{k}) : \mathbb{R}^p \longrightarrow \mathbb{R}^+ \tag{3.2}$$

that reflects how well the systemic properties of a model match the desired behavior. For a given $\mathbf{k}$, the lower the value of $E(\mathbf{k})$, the better the model behaves. We call a parameter vector $\mathbf{k}$ viable if it fulfills the condition $E(\mathbf{k}) < E_0$ ($E_0 > 0$) that is, if the cost function does not exceed some positive threshold E_0. It should be consistent with the viability criterion: $E(\mathbf{k}) < E_0$ iff $\boldsymbol{\pi}(\mathbf{k}) \in [\underline{\boldsymbol{\pi}}, \overline{\boldsymbol{\pi}}]$. For example, in circadian models, $\pi_T(\mathbf{k})$ could be the period of oscillation of the model, given the parameter vector $\mathbf{k}$. If the viable range is

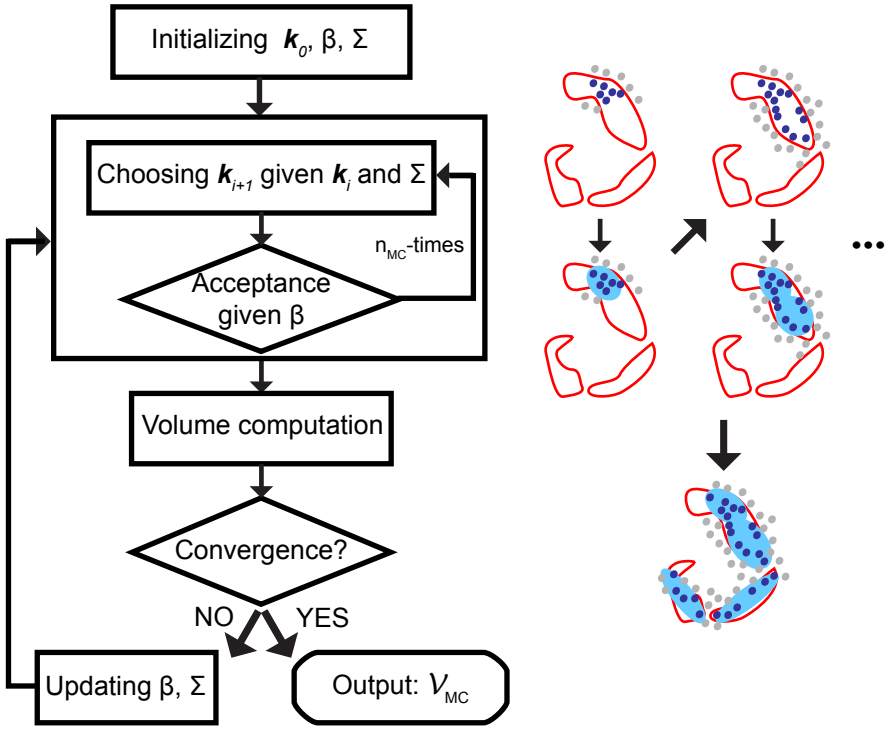

Figure 3.2: Flowchart representing the basic scheme of the Adaptive Metropolis Monte Carlo (AMC) algorithm. Given an initial parameter point $\mathbf{k}_0$, covariance matrix $\boldsymbol{\Sigma}$ and β, the algorithm carries out n_{MC} iterations in which every new parameter point is sampled from a normal distribution $\mathcal{N}(\mathbf{k}_i, \boldsymbol{\Sigma})$, and accepted or rejected based on Metropolis acceptances ratios. Every n_{MC} iterations the viable points (dark blue and gray points correspond to viable and non-viable sampled parameter points, respectively) found so far are grouped into clusters and the volume (light blue ellipsoids) of the ellipsoids that enclose the viable parameter points in each cluster are calculated. If the sum of these volumes converges the algorithm stops; if not, the covariance matrix Σ and β are updated, and n_{MC} new iterations are performed. The output of this stage is the set $\mathcal{V}_{MC}$ which includes all the viable parameter points found.

$[\underline{\boldsymbol{\pi}}, \overline{\boldsymbol{\pi}}] = [22h, 26h]$, the cost function could be $E(\mathbf{k}) = |\pi_T(\mathbf{k}) - 24|$ with the viability threshold $E_0 = 2$.

Following the basic concept of the Metropolis algorithm [153, 156], the cost function (3.2) is associated with the energy of a statistical mechanical system that is in contact with a thermal bath whose temperature changes with time. This system never reaches equilibrium, because every time the temperature changes the system is pushed into states far from equilibrium.

To simulate this system, we use an out-of-equilibrium Monte Carlo method

[157] where the new parameter vector $\mathbf{k}$ is drawn from the i-th parameter vector $\mathbf{k}_i$ according to the probability

$$p(\mathbf{k}) = \frac{1}{(2\pi)^{p/2}\det(\mathbf{\Sigma})} \exp\left[-\frac{1}{2}\left(\mathbf{k}-\mathbf{k}_i\right)\mathbf{\Sigma}^{-1}\left(\mathbf{k}-\mathbf{k}_i\right)'\right], \tag{3.3}$$

which is a Gaussian distribution with the mean $\mathbf{k}_i$ and a covariance matrix $\mathbf{\Sigma}$. The acceptance ratio is according to the Metropolis [156] algorithm:

$$A(\mathbf{k}_i \to \mathbf{k}) = \begin{cases} \exp\left[-\beta\left(E(\mathbf{k})-E(\mathbf{k}_i)\right)\right] & \text{if } E(\mathbf{k})-E(\mathbf{k}_i) > 0, \\ 1 & \text{otherwise,} \end{cases} \tag{3.4}$$

where β^{-1} is proportional to the temperature of a statistical mechanical system.

Given β and $\mathbf{\Sigma}$, the simulation starts from a known viable parameter vector $\mathbf{k}_0$. Then, from $\mathbf{k}_0$, a new $\mathbf{k}$ is drawn by sampling the distribution (3.3) centered on $\mathbf{k}_0$. If $E(\mathbf{k}) \leq E(\mathbf{k}_0)$, the new $\mathbf{k}$ is automatically accepted and becomes $\mathbf{k}_1$. In contrast, if $E(\mathbf{k}) > E(\mathbf{k}_0)$, $\mathbf{k}$ is accepted with a probability $A = \exp\left[-\beta\left(E(\mathbf{k})-E(\mathbf{k}_0)\right)\right]$, in which case it becomes $\mathbf{k}_1$. If $\mathbf{k}$ is rejected, then $\mathbf{k}_1 = \mathbf{k}_0$. This scheme is repeated for a predefined number of iterations n_{MC}.

After n_{MC} iterations the algorithm determines whether AMC sampling must stop. To do so, the viable parameter vectors found so far are divided into clusters whose number is defined with the procedure explained in the appendix A.2. Then, AMC calculates the ellipsoids with minimum volume that enclose the points grouped in each cluster and computes the sum of all ellipsoids volumes. The algorithm stops when the volume of all ellipsoids converges or a maximum number of iterations is reached. If either of these criteria are met, AMC sampling terminates and returns as its result the set $\mathcal{V}_{MC}$ of all viable parameter vectors it found. Otherwise, n_{MC} more iterations are carried out after updating β and Σ according to

$$\beta = \begin{cases} b\beta, & \text{if } f_v = 0, \\ \beta, & \text{if } 0 < f_v \leqslant f_0, \\ \beta/b, & \text{if } f_v > f_0, \end{cases} \qquad \mathbf{\Sigma} = \begin{cases} s\mathbf{\Sigma}, & \text{if } f_a > f_u \\ \mathbf{\Sigma}, & \text{if } f_l < f_a \leqslant f_u, \\ \mathbf{\Sigma}/s, & \text{if } f_a < f_l, \end{cases} \tag{3.5}$$

where f_v and f_a are the proportions of sampled viable parameter vectors and accepted transitions, respectively calculated over the last n_{MC} iterations. The values $\{b, s\} > 1$ are parameters of the algorithm to be specified by the user. Equation (3.5) implies the following update procedure. When Monte Carlo sampling is mainly confined to a viable region ($f_v > f_0$), β decreases and the frequency of accepted transitions with higher cost increases. If this makes the

frequency of accepted transitions larger than an upper limit ($f_a > f_u$), the covariance matrix $\mathbf{\Sigma}$ will become larger and the method will sample broader regions. In contrast, when the method has not found any viable parameter vector ($f_v = 0$), β increases and less parameters with higher cost are accepted in order to force the algorithm to sample regions with lower cost function. If this frequency falls below a lower limit ($f_a < f_l$), $\mathbf{\Sigma}$ decreases to maintain the desired frequency of accepted transitions by reducing the size of the jumps between two consecutive parameter vectors. The aim of this tuning is to sample specifically the border of the viable region and not its center, such that the whole region is quickly covered. Note that these update rules may trap the random walk in a region without any viable point. In this case, the algorithm should be reset to the values of the last iteration where viable points were found.

Ellipsoid-based Sampling

The basic idea the second stage of the parameter search, the ellipsoid-based sampling (EBS) is the same as the PCA sampling, but with multiple ellipsoids. To explore the parameter space in detail, multiple sampling regions are used with different centers and orientations to enclose the viable space in an optimal way (Figure 3.3).

The algorithm constructs a series of ellipsoids starting from different viable parameter points. The procedure we now describe is valid for each such ellipsoid expansions from the j initial viable point. It starts by selecting a viable parameter point $\mathbf{k}_j \in \mathcal{V}_{MC}$ in an adaptive way, as described below. In the first ellipsoid expansion, this point will typically be a viable point obtained from AMC. To construct the first ellipsoid from $\mathbf{k}_j$, the algorithm uses a bisection technique [153] to find $2p$ viable parameter points near the intersection between the boundary of the viable region and the straight lines parallel to the axes of the Cartesian coordinate system that pass through $\mathbf{k}_j$ (see Appendix A.4 for a technical description). Starting from $i = 0$, each step of the iterative ellipsoid construction proceeds as follows. From the $2p$ initial points if $i = 0$ or from the set of viable points $\mathcal{V}_j^i$ (that comprises the viable points found after the iteration i starting from the j-th initial viable point) if $i \neq 0$, the EBS constructs an ellipsoid L_j^i that encloses all the points in this set (see Appendix A.1. From this ellipsoid, it creates a new ellipsoid S_j^i that has the same orientation as L_j^i, but that has the lengths of its axes multiplied by a scaling parameter λ_i. Then, it uniformly samples a predefined number of parameter points n_{ES} from

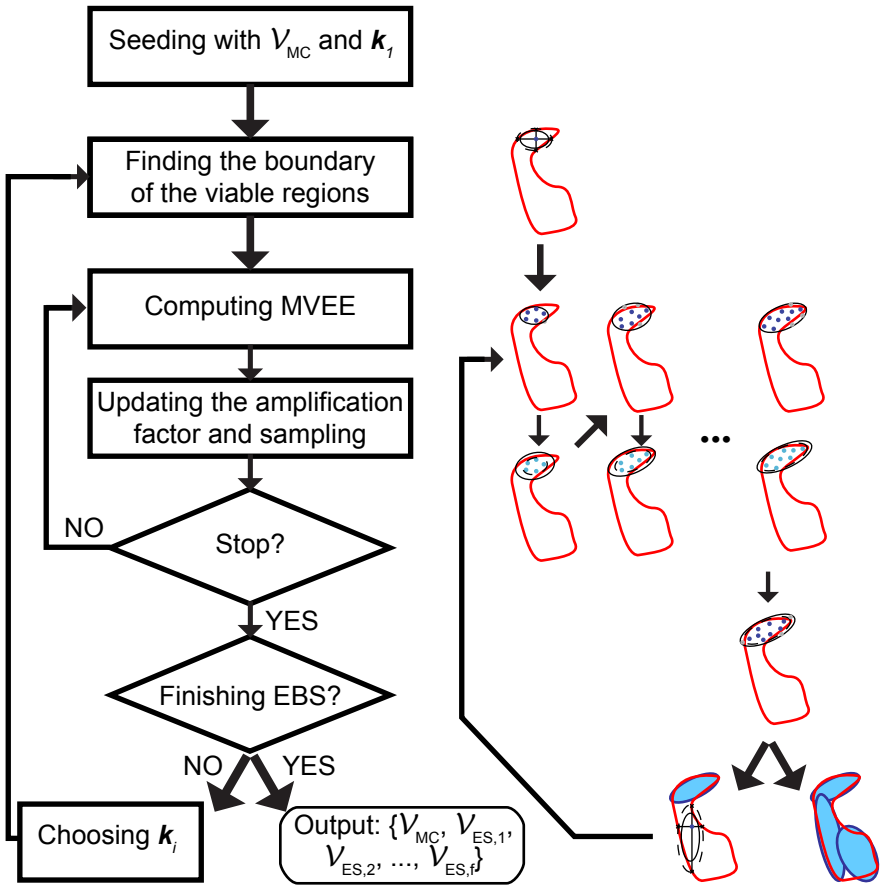

Figure 3.3: Flowchart for the ellipsoid-based sampling (EBS) procedure. Given $\mathcal{V}_{MC}$, the set of viable parameter points found by AMC, and an initial viable parameter point $\mathbf{k}_1$, the method finds viable parameter vectors (dark blue points, non-viable are in gray) near the boundary of the viable region. Then, it calculates the minimum volume enclosing ellipsoid (MVEE) that encloses those viable parameter points (dashed curves) and samples inside an ellipsoid with the same orientation but smaller axes (the sampling ellipsoid are represented by solid curves). After that, the method again calculates the MVEE of the viable points found so far (light blue points), and samples inside a scaled ellipsoid with the same orientation but larger axes. The exploration started in $\mathbf{k}_1$ finishes when the scaling factor tends to one or after a fixed number of iterations is reached. If this does not happen the method calculates the MVEE of the viable parameter points found and performs a new uniform sampling inside a new scaled ellipsoid. At the end of every new ellipsoid expansion, the algorithm checks if EBS must stop. It finishes if the algorithm does not find any new viable points in viable non-explored regions (viable explored regions are represented by light blue ellipsoids). If EBS does not stop, it carries out another ellipsoid expansion starting from a different point $\mathbf{k}_i$. The result of the EBS is the set of the viable parameter points found during all the ellipsoid expansions $\mathcal{V}_{MC} \cup \mathcal{V}_{ES}$ where $\mathcal{V}_{ES} = \mathcal{V}_{ES,1} \cup \mathcal{V}_{ES,2} \cup \cdots \cup \mathcal{V}_{ES,f}$.

this ellipsoid S_j^i. The union of the set of viable points in S_j^i with $\mathcal{V}_j^i$ then gives $\mathcal{V}_j^{i+1}$.

Selection of the scaling parameter λ_i is critical for the performance of the algorithm. We define it as:

$$\lambda_i = \begin{cases} \lambda_0 < 1, & \text{if} \quad i = 0 \\ \lambda_1 > 1, & \text{if} \quad i = 1, \\ \lambda_{i-1} + \dfrac{(\lambda_{i-1} - 1)}{\sigma}, & \text{if} \quad \left|\mathcal{V}_j^i\right| - \left|\mathcal{V}_j^{i-1}\right| > n_{ES} b_u,\ i > 1, \\ \lambda_{i-1} - \dfrac{(\lambda_{i-1} - 1)}{\sigma}, & \text{if} \quad \left|\mathcal{V}_j^i\right| - \left|\mathcal{V}_j^{i-1}\right| < n_{ES} b_l,\ i > 1, \\ \lambda_{i-1}, & \text{otherwise.} \end{cases} \tag{3.6}$$

where $\left|\mathcal{V}_j^i\right|$ indicates the number of elements in the set and b_l, b_u, and $\sigma < 1$ are parameters for lower and upper bounds, and for axis scaling, respectively.

The rationale behind equation (3.6) is as follows: Points in L_j^0 lie near the boundary of the viable space. In high dimensional spaces the curse of dimensionality may cause a large proportion of this ellipsoid volume to be filled by nonviable points. Setting $\lambda_0 < 1$ makes S_j^0 smaller than L_j^0, and makes it more likely that S_j^0 contains a larger proportion of viable parameter vectors, which will lead to a larger set $\mathcal{V}_j^0$. To explore a larger elliptic region around $\mathbf{k}_j$, the method then performs a second iteration with $\lambda_1 > 1$. All subsequent iterations depend on the number of viable points found in the last iteration $\left(\left|\mathcal{V}_j^i\right| - \left|\mathcal{V}_j^{i-1}\right|\right)$. Specifically, when this number of points is larger than some upper limit $n_{ES} b_u$, the scaling parameter grows by a factor $1/\sigma > 1$ to explore larger domains of parameter space. When the difference is below some lower limit $n_{ES} b_l$ – only few additional viable points have been found in the last iteration – shrinking the axes allows an efficient exploration of smaller regions. Thus, viable parameter points found in previous iterations guide and define the ellipsoid where the next sampling is carried out.

The iterative procedure started at $i = 0$ from $\mathbf{k}_j$ finishes when λ_i converges to 1 or after a fixed number of iterations is reached. The procedure's output is $\mathcal{V}_{ES,j}$ a set of sampled viable points that contains the $2p$ viable parameter points found near the boundary of the viable space, and the set of viable parameter vectors $\mathcal{V}_j^i$ updated in the last iteration.

If the EBS algorithm had carried out ellipsoids expansions around a single initial point as the PCA sampling, the method would have only been able to explore viable spaces whose geometry is well approximated by an ellipsoid. Therefore, if the viable space is non-convex, EBS starts ellipsoid expansions

from viable parameter points placed in different regions. To sample inside different ellipsoids that cover the whole viable space locally, we choose a new starting viable parameter point $\mathbf{k}_{j+1}$ from the set composed by $\mathcal{V}_{MC}$ and the union of $\mathcal{V}_{ES,i}$, $i = 1 \dots j$, that is, the set of viable points obtained after AMC exploration and previous ellipsoid expansions, respectively. To explore regions that have not yet been sampled, we preferentially select a $\mathbf{k}_{j+1}$ that is far away from the average of all previous starting points $\mathbf{k}_i$, $i = 1 \dots j$ (see Annex for details).

At the end of every new ellipsoid expansion j, the algorithm determines if EBS should stop. To do so, the viable parameter points found up to then $\{\mathcal{V}_{MC}, \mathcal{V}_{ES,1}, \mathcal{V}_{ES,2} \dots, \mathcal{V}_{ES,j}\}$ are divided into a predefined number of clusters. Then, EBS calculates the ellipsoids with minimum volume that enclose the points grouped in each cluster and computes the sum of all ellipsoids volumes. The algorithm stops when the volume of all ellipsoids converges or a maximum number of ellipsoid expansions is reached. After stopping, EBS returns its final result, the set of viable parameter points $\mathcal{V}_{MC} \cup \mathcal{V}_{ES}$ where $\mathcal{V}_{ES} = \mathcal{V}_{ES,1} \cup \mathcal{V}_{ES,2} \cup \dots \cup \mathcal{V}_{ES,f}$.

Monte Carlo Integration

As in the PCA sampling method, the final stage is a Monte Carlo integration to obtain a uniform sampling over the viable region and its size. We use the set $\mathcal{V}_{MC} \cup \mathcal{V}_{ES}$ to construct the domain of integration I for the Monte Carlo integration. The results of the EBS part ensure a better coverage of the viable region. To optimize the sampling of the domain, we split the region covered by I in a family of ellipsoids that cover the viable space. To determine these ellipsoids we group the set of viable parameter points $\mathcal{V}_{MC} \cup \mathcal{V}_{ES}$ into n_C clusters, and compute the ellipsoid I_i with minimum volume that encloses the viable points grouped in the i-th cluster (see Appendix A.2 for details).

In this procedure, the subspace I is given by the points of the parameter space enclosed by the n_C ellipsoids $I = \{\mathbf{k} \in \mathbb{R}^p \,|\mathbf{k} \in \cup_{i=1}^{n_C} I_i\}$ where I_i is the region of the parameter space enclosed by the ith ellipsoid. In general, the ellipsoids may intersect, so the volume of I is smaller than the sum of the volumes of I_i. To avoid the resulting inaccuracy in volume estimation, we introduce the following integrand valid for the ellipsoid I_i

$$f_i(\mathbf{k}) = \begin{cases} 0 & \text{if } \theta \in \cup_{j=1}^{i-1} I_j \\ 0 & \text{if } \boldsymbol{\pi}(\mathbf{k}) \notin [\underline{\boldsymbol{\pi}}, \overline{\boldsymbol{\pi}}] \\ 1 & \text{otherwise, i.e. viable} \end{cases} \tag{3.7}$$

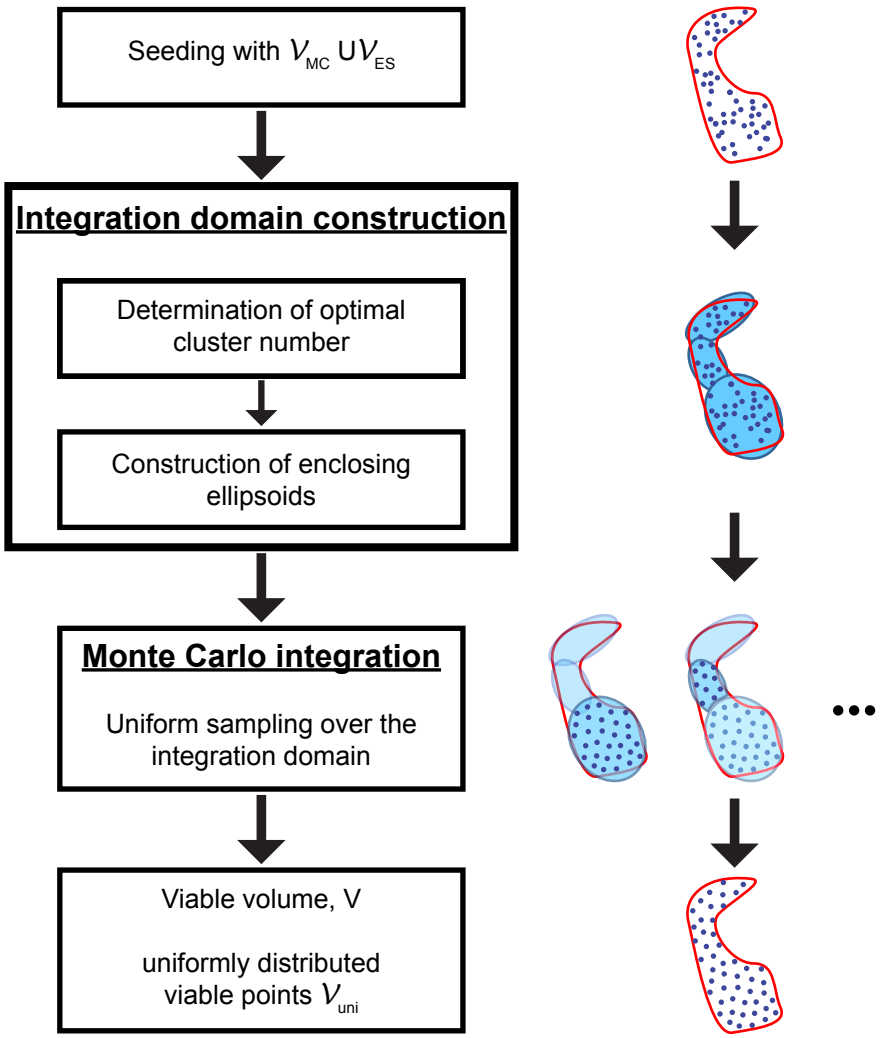

Figure 3.4: Flowchart representing the algorithm responsible for the viable volume estimation, and the acquisition of a set of uniformly distributed viable parameter vectors. A set of viable parameter points $\mathcal{V}_{MC} \cup \mathcal{V}_{ES}$ (uppermost set of blue points) cover the whole viable space (area enclosed by the red curve) seeds the algorithm. Then, the method groups these points into k-clusters (3 clusters in this case), and calculates the ellipsoids with minimum volume that enclose the points in each cluster (light blue ellipsoids). After that, it performs a Monte Carlo integration of every ellipsoid (the intersections between ellipsoids are sampled only once). The output of the algorithm is a set of uniformly distributed viable parameter vectors $\mathcal{V}_{uni}$ (bottom set of blue points), from which the viable volume V can be estimated.

This integrand evaluates the viability of the parameter vectors in the ellipsoid intersections only once and therefore, by sampling uniformly N parameter points in I, we can estimate the viable volume V as

$$V = \sum_{i=1}^{n_C} \int_{I_i} f_i(\mathbf{k}) d\mathbf{k} = \sum_{i=1}^{n_C} \left(\frac{\mathrm{Vol}(I_i)}{N_i} \sum_{j=1}^{N_i} f_i(\mathbf{k}_j) \right), \text{ with } \sum_{i=1}^{n_C} N_i = N \tag{3.8}$$

where N_i is the number of uniformly distributed parameter vectors inside I_i. This algorithm also yields a set of uniformly distributed viable parameter vector $\mathcal{V}_{uni} = \cup_{i=1}^{n_C} \{\mathbf{k} \in I_i | f_i(\mathbf{k}) = 1\}$ that should cover the whole region of viability.

3.1.3 Error in the Monte Carlo Integration

To estimate the sampling errors in the viable fractions and volumes, we note that $|\mathcal{V}_{uni}|$ (written as $|\mathcal{V}|$ in the following), as estimated by Monte Carlo integration is a binomially distributed random variable [148, 153]. An estimate of its standard deviation is $\Delta(|\mathcal{V}|) = \sqrt{\frac{|\mathcal{V}|(|\mathcal{S}|-|\mathcal{V}|)}{|\mathcal{S}|}}$. Of interest is the coefficient of variation or relative error, defined as the standard deviation divided by the mean. For $|\mathcal{V}|$, this relative error is given by $\frac{\Delta(|\mathcal{V}|)}{|\mathcal{V}|} = \frac{\Delta V}{V} = \sqrt{\frac{|\mathcal{S}|-|\mathcal{V}|}{|\mathcal{V}||\mathcal{S}|}}$. For the normalized quantity $R = \sqrt[p]{V}$, the relative error needs to be divided by p, i.e., it calculates as $\frac{\Delta R}{R} = \left(\frac{1}{p}\right) \sqrt{\frac{|\mathcal{S}|-|\mathcal{V}|}{|\mathcal{V}||\mathcal{S}|}}$ which scales as $\left(\frac{1}{p}\right) \frac{1}{\sqrt{|\mathcal{S}|}}$. Furthermore we estimate the necessary sample size $|\mathcal{S}|$ for a given relative accuracy δ and confidence. Applying Hoeffding's inequality [158], we obtain

$$\Pr\left\{ \left| 1 - \frac{\mathsf{E}(|\mathcal{V}|)}{|\mathcal{V}|} \right| \geq \delta \right\} \leq 2\, e^{-2\delta^2 \left(\frac{|\mathcal{V}|}{|\mathcal{S}|}\right)^2 |\mathcal{S}|},$$

where $\mathsf{E}(\cdot)$ denotes the expectation operator. Thus, estimating the sampling acceptance ratio $|\mathcal{V}|/|\mathcal{S}|$ from a sufficiently large ensemble and assuming it to be constant for the successive sampling, we can compute a lower bound for the necessary sample size. For example, asking for 10% accuracy with a confidence of 95% at an acceptance ratio of 1/20, Hoeffding's bound requires the sample size to be $|\mathcal{S}| > 60000$.

In the two-stage method the estimation of the error is the composition of the error on the integration in all individual ellipsoids where each of them gives a viable volume V_i. It can be approximated by the following equation:

$$\frac{\Delta V}{V} = \frac{\sum_{i=1}^{n_C} \Delta V_i}{\sum_{i=1}^{n_C} V_i} = \frac{\sum_{i=1}^{n_C} \mathrm{Vol}(I_i) \sqrt{\frac{\nu_i}{N_i^3}(\nu_i - N_i)}}{\mathrm{Vol}(I_i) \frac{\nu_i}{N_i}} \qquad \text{where } \nu_i = \sum_{j=1}^{N_i} f_i(\mathbf{k}_j) \tag{3.9}$$

We advice caution that, in practice, one can never be certain that the whole viable space is contained in the integration domain used by either the PCA or the two-stage sampling methods. The agreement between the actual viable volume and the estimated viable volume V, depends on the proportion of the viable volume that is enclosed in sampling region. Due to the high dimensionality, we have to balance between a large and conservative sampling region (high accuracy) and a tight region (higher efficiency) to keep a reasonable computational time.

We now briefly comment on how estimation errors scale with the number of dimensions p. The only possible general statement is that the ratio between the viable volume and the volume of the sampling region scale exponentially with p. Therefore, $\frac{|\mathcal{V}|}{|\mathcal{S}|} \sim \alpha^{-p}$ with α being dependent on the geometry of the viable volume. For example, $\alpha = 1$ if the viable volume is identical to the sampling region, and only in this trivial case does the error not depend on p. For example, if the viable parameter volume has an ellipsoidal shape and, in the case of the PCA method, the sampling region is a hyper-rectangle, α increases from 1.5 to 2.5 if the dimension increases from $p = 5$ to $p = 22$. The coefficient of variation (relative error) of the viable volume scales as $\frac{\alpha^{p/2}}{p\sqrt{|\mathcal{S}|}}$. The size and the shape of the sampling hyper-rectangle is crucial for low errors: a larger hyperbox means that an exponentially greater number of points needs to be sampled for high dimensional systems to ensure constant error. These observations underscore the usefulness of the preliminary search to define the integration region, as it dramatically reduces computational requirements.

3.2 Local Robustness Quantifiers

The second, local part of the glocal method assesses the robustness of every viable parameter $\mathbf{k}$ with different quantifiers. I will here present the five different local measures that I used in the article using the glocal method [134]. These quantifiers have been developed for their application to the models of the cyanobacterial circadian clock, but they proved to be useful in other oscillatory systems [159]. Note that the last two quantifiers requires a stable cycle and can therefore only be applied to oscillatory systems. The list is not extensive and other local robustness properties may be studied as I did concerning the entrainment of the *Drosophila* circadian clock (section 4.2).

3.2.1 Robustness to Perturbations of Parameters, ρ_P

The first local robustness quantifier $\rho_P(\mathbf{k})$ computes the fraction of local random perturbations of parameters that preserve $\boldsymbol{\pi}$. To estimate $\rho_P(\mathbf{k})$ for any specific model, one generates many random perturbations of each viable parameter vector $\mathbf{k}$, for example with a Gaussian distribution centered on $\mathbf{k}$, determine the model's behavior with these parameters, and define ρ_P as the fraction of perturbations preserving $\boldsymbol{\pi} \in [\underline{\boldsymbol{\pi}}, \overline{\boldsymbol{\pi}}]$. The standard deviation of the Gaussian distribution is best chosen such that (i) all values of ρ_P in the allowed interval $[0, 1]$ are observed, and that (ii) ρ_P can be distinguished most significantly for the two considered models.

For my application to the cyanobacterial circadian clock models, I perturbed 1000 times each viable parameter vector $\mathbf{k}$. In each of these perturbations, I multiplied each component k_i of $\mathbf{k}$ with a Gaussian random variate of mean one and standard deviation $\sigma = 0.2$. I chose this value, because it yields different values of ρ_P for different parameter vectors, thus allowing me to assess how robustness varies in different regions of parameter space, and because it permits discrimination of ρ_P among the two studied models.

Related to perturbations of the parameters, I address the robustness to temperature changes. Ideally, the Arrhenius equation has to be used [43, 87], however this approach requires knowledge of the activation energies of each reaction in a system, which is usually not available. I thus simply assume that an increase in temperature corresponds to a random increase of all parameters. This aspect of robustness is quantified with the same approach used for estimating ρ_P. Mean and standard deviation are the same, but perturbations are correlated, such that all parameters are multiplied with variates that are either above one or below one for a particular perturbation. Besides finite perturbations, the sensitivity of the period to such parameter changes [43, 160, 87] can be studied. Specifically, the values $\alpha_i = \partial \log(T)/\partial \log(k_i)$ can be calculated using our derivation for ρ_S (see section 3.2.5). In general a faster reaction decreases the period of the cycle ($\alpha_i < 0$) but in order to have temperature compensation in any cycle, at least one of the values α_i should be positive [43, 160, 87].

3.2.2 Robustness to Perturbations of Total Concentrations, ρ_C

The second local robustness quantifier $\rho_C(\mathbf{k})$ regards alterations in the total amount of key proteins. It is specific for the use with systems having

mass conservation. In the case of the *in vitro* experiments of the cyanobacterial circadian clock [92, 95], a pre-determined number of the Kai molecules is used. This number may vary *in vivo*, for example due to changes in cell volume caused by the cell division cycle and this quantifier evaluates the robustness of the model to variation of the ratio of proteins. It uses the same approach as the robustness to total parameter perturbations ρ_P: one generates a large number of perturbed concentrations, and numerically integrates the model with these perturbed concentrations. For a given parameter vector $\mathbf{k} \in \mathcal{V}$, ρ_C is defined as the fraction of these perturbations preserving $\boldsymbol{\pi}$.

3.2.3 Robustness to Molecular Noise, ρ_N

The third robustness quantifier, $\rho_N(\mathbf{k})$, reflects that chemical reactions are stochastic events [42, 51, 161]. In systems working at a steady state, coefficient of variation could be used as a measure or resilience to molecular noise. For oscillatory systems, I propose to define ρ_N as the fraction of trajectories that preserve the oscillator period π_T for a given viable $\mathbf{k}$. Yet, when stochasticity is involved, the definition of a period becomes problematic, because considerable amplitude variations from one period to the next may be present. The Hilbert transform method, which yields information about the phase of a trajectory [162, 163] helps circumvent this problem. It is preferable to the Fourier transform in situations where period needs to be estimated independently from amplitude. To quantify $\rho_N(\mathbf{k})$, I first simulate [15] stochastic trajectories over a large number of periods using $\mathbf{k}$ with a specific cell volume v. The quantifier ρ_N is equal to the ratio of the number of completed cycles with the correct period to the total number of completed cycles in a deterministic simulation of the same duration. Mathematically, if the desired or empirically observed period of an oscillatory system is T_0 and the period is allowed to vary over a certain range $[\underline{\pi_T}, \overline{\pi_T}]$, let $\mathcal{T}(\mathbf{k})$ be the set of observed period durations in the stochastic simulation of duration τ, for a circuit with parameter vector $\mathbf{k}$. With this notation, ρ_N is equal to $t(\mathbf{k})/z(\mathbf{k})$, where $t(\mathbf{k})$ is the total number of cycles with a period in the allowable range that occurred in the time τ, and z is a normalization constant defined as $z(\mathbf{k}) = \mathbf{max}\left\{\frac{\tau}{T_0}, |\mathcal{T}(\mathbf{k})|\right\}$.

The constant $z(\mathbf{k})$ is chosen such that ρ_N is smaller than one and that we avoid misclassification of trajectories having long duration without oscillation, and bursts of oscillations with the correct period. For example, consider a hypothetical oscillator with a desired period of $T_0 = 24h$, 10% allowable variation around this period, and a simulation length $\tau = 360h$. In a deterministic

(noiseless) simulation, the oscillator would complete exactly $\frac{\tau}{T_0} = 15$ cycles, but noise could change this number. Typically, it would cause the periods of the cycles to spread over a range larger than allowable, but the majority of cycles might have the correct period. If, for example, we observe 14 cycles, six of them with a period of $24.5h$, six with a period of $25.5h$ and two with a period of $28h$ (outside the allowable range), then ρ_N would calculate as $\rho_N = \frac{12}{15} = 0.8$. The second example concerns the special case where noise speeds up the oscillation. If 16 cycles of period $22.5h$ are observed, then the maximum in the normalization function above is important to have ρ_N contained between 0 and 1, because $\rho_N = \frac{16}{max(15,16)} = 1$. Lastly, consider the case where molecular noise generates pauses in the cycle. Given a trajectory starting with two cycles having an acceptable duration of $24h$, then a long cycle (pause) of $260h$ and again 2 cycles with acceptable period, ρ_N calculates as $\frac{4}{15} = 0.267$ with our normalization. The much simpler normalization that divides by the number of completed cycles would yield an artificially inflated value of $\rho_N = \frac{4}{5} = 0.8$.

3.2.4 Attraction of the Cycle, ρ_A

The fourth robustness quantifier, $\rho_A(\mathbf{k})$ (for attraction of the cycle), measures how fast the oscillator returns to its cycling behavior when its trajectory is transiently perturbed. In biological systems, most regular oscillations are limit cycle oscillations [72], where a system returns to its pre-perturbation nominal oscillatory behavior after a perturbation of its trajectory. Some such oscillations may be much more robust, in the sense that they would converge very rapidly to this nominal behavior, whereas others may take a long time to 'absorb' the effects of a transient perturbation.

A commonly used analytical approach of estimating this convergence to oscillatory behavior uses Floquet theory [164]. Briefly, a system's convergence to a cycle can be estimated through its largest Floquet multiplier, which can be thought of the fraction of a small trajectory perturbation that remains after one cycle. For stable cycle, Floquet multipliers assume values between zero and one, and the smaller the multiplier, the faster a perturbation is absorbed. To calculate the largest Floquet multiplier, the variational equations [164] of the studied models is integrated over one cycle, and μ is the largest eigenvalue. The quantifier $\rho_A(\mathbf{k})$ is equal to $1 - \mu$. This measure of robustness corresponds to the fraction of the perturbation that is 'absorbed' by the system after one cycle. This quantifier is complementary to ρ_N for the following reasons. First, ρ_A is a deterministic measure and thus remains useful for very large molecule copy

numbers. Second, it characterizes a system's response to arbitrary transient state perturbations, not just to fluctuations due to molecular noise.

3.2.5 Sensitivity of the Period, ρ_S

The fifth quantifier, $\rho_S(\mathbf{k})$ (for sensitivity analysis of period), assesses the effect of an infinitesimal change of an individual parameter or parameter vector on the period T of a system. Specifically, it is related to the gradient-vector $\partial T/\partial \mathbf{k}$. A component i of this vector with large absolute value indicates a parameter k_i that affects T to a great extent. The robustness measure $\rho_S(\mathbf{k})$ is defined as $\left(1 + \left\| \frac{\partial \log(T)}{\partial \log(|\mathbf{k}|)} \right\|\right)^{-1}$. The logarithm in this expression occurs, because the relative effect of one parameter on the period is of interest. The expression $\|\mathbf{x}\|$ denotes the Euclidian norm $\sqrt{\sum_i x_i^2}$ of the vector $\mathbf{x}$. The quantifier ρ_S assumes values between zero and one. The larger the value of ρ_S, the more robust a model is. A value of $\rho_S = 0.5$ means that a one percent change in a parameter vector results in a one percent change in the period.

In order to estimate ρ_S, I need to derive the first order approximation of the quantity $\frac{\partial T}{\partial k_i}$, i.e., the response of a dynamical system's period T to an infinitesimal change in one component k_i of its parameter vector $\mathbf{k}$. For simplicity of notation, I first derive the expression for a single parameter k and then extend it to a parameter vector $\mathbf{k}$.

For an ordinary differential system $\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathbf{F}(\mathbf{x}(t), k)$ with a parameter k that has a periodic solution $\boldsymbol{\xi}(t, k) = \boldsymbol{\xi}(t + T(k), k)$ of period $T(k)$, I define the nominal parameter k_0, the corresponding nominal solution $\boldsymbol{\xi}_0(t) = \boldsymbol{\xi}(t, k_0)$ and the nominal period $T_0 = T(k_0)$. The infinitesimal increment of the parameter is $\mathrm{d}k$. With this notation, I can write $\mathrm{d}T = T(k_0 + \mathrm{d}k) - T_0$ and $\mathrm{d}\boldsymbol{\xi}(t) = \boldsymbol{\xi}(t, \mathrm{k}_0 + \mathrm{d}k) - \boldsymbol{\xi}_0(t)$. I have a family of solutions $\boldsymbol{\phi}(t, \mathbf{x}_0, k) = \mathbf{x}(t, k)$, with $\mathbf{x}(0) = \mathbf{x}_0$. Then the sensitivity matrix $\mathbf{M}(t)$ can be defined as $(\mathbf{M}(t))_{ij} = \frac{\partial \phi_i}{\partial x_j}(t, \mathbf{x}_0, k)$ and the parameter sensitivity matrix is $(\mathbf{V}(t))_i = \frac{\partial \phi_i}{\partial k}(t, \mathbf{x}_0, k)$.

$\mathbf{M}(t)$ is the solution of the variational equation

$$\frac{\mathrm{d}\mathbf{M}}{\mathrm{d}t} = \frac{\mathrm{d}\mathbf{F}}{\mathrm{d}\mathbf{x}}(\xi_0(t), k_0)\mathbf{M}(t) \ , \quad \mathbf{M}(0) = \mathbf{I}$$

and $\mathbf{V}(t)$ the solution of the variational equation

$$\frac{\mathrm{d}\mathbf{V}}{\mathrm{d}t} = \frac{\mathrm{d}\mathbf{F}}{\mathrm{d}\mathbf{x}}(\xi_0(t), k_0)\mathbf{V}(t) + \frac{\partial \mathbf{F}}{\partial k}(\xi_0(t), k_0), \quad \mathbf{V}(0) = \mathbf{0}$$

Then up to the first order approximation

$$\mathrm{d}\boldsymbol{\xi}(t) \cong \mathbf{M}(t)\mathrm{d}\boldsymbol{\xi}(0) + \mathbf{V}(t)\mathrm{d}k$$

For simplicity, choose H the Poincaré section as the hyperplane that goes through $\boldsymbol{\xi}_0(0)$ and is perpendicular to $\mathbf{e} = \frac{\mathrm{d}\boldsymbol{\xi}_0}{\mathrm{d}t}(0) \Big/ \left\| \frac{\mathrm{d}\boldsymbol{\xi}_0}{\mathrm{d}t}(0) \right\|$. Then $\mathbf{P} = \mathbf{I} - \mathbf{e}\,\mathbf{e}^T$ is the orthogonal projection on the hyperplane H.

Note that by definition $\boldsymbol{\xi}_0(0) = \boldsymbol{\xi}_0(T_0) \in H$. I suppose also, without restriction or loss of generality that $\boldsymbol{\xi}(0,k) \in H$ which implies that $\mathbf{P}\mathrm{d}\boldsymbol{\xi}(0,k) = \mathrm{d}\boldsymbol{\xi}(0,k)$. However, in general:

$$\mathbf{P}\mathrm{d}\boldsymbol{\xi}(T_0, k_0 + \mathrm{d}k) \neq \mathrm{d}\boldsymbol{\xi}(T_0, k_0 + \mathrm{d}k)$$

but I have

$$\boldsymbol{\xi}(T(k_0 + \mathrm{d}k), k_0 + \mathrm{d}k) = \boldsymbol{\xi}(0, k_0 + \mathrm{d}k) \in H$$

This can be expressed as $\mathbf{e}^T\left(\boldsymbol{\xi}(T_0 + \mathrm{d}T, k_0 + \mathrm{d}k) - \boldsymbol{\xi}_0(0)\right) = 0$ and with the first order approximation

$$\begin{aligned}
\boldsymbol{\xi}(T_0 + \mathrm{d}T, k_0 + \mathrm{d}k) - \boldsymbol{\xi}_0(0) &\approx \frac{\partial \boldsymbol{\xi}}{\partial t}(T_0, k_0)\mathrm{d}T + \frac{\partial \boldsymbol{\xi}}{\partial k}(T_0, k_0)\mathrm{d}k \\
&= \frac{\partial \boldsymbol{\xi}}{\partial t}(0, k_0)\mathrm{d}T + \mathbf{M}(T_0)\frac{\partial \boldsymbol{\xi}}{\partial k}(0, k_0)\mathrm{d}k + \mathbf{V}(T_0)\mathrm{d}k
\end{aligned}$$

I have

$$\begin{aligned}
0 &\approx \mathbf{e}^T\left(\mathbf{e}\left\|\frac{\partial \boldsymbol{\xi}}{\partial t}(0, k_0)\right\|\mathrm{d}T + \mathbf{M}(T_0)\frac{\partial \boldsymbol{\xi}}{\partial k}(0, k_0)\mathrm{d}k + \mathbf{V}(T_0)\mathrm{d}k\right) \\
\mathrm{d}T &\approx -\frac{\mathbf{e}^T\left(\mathbf{M}(T_0)\frac{\partial \boldsymbol{\xi}}{\partial k}(0, k_0)\mathrm{d}k + \mathbf{V}(T_0)\mathrm{d}k\right)}{\left\|\frac{\partial \boldsymbol{\xi}}{\partial t}(0, k_0)\right\|}
\end{aligned}$$

therefore

$$\begin{aligned}
\frac{\partial \boldsymbol{\xi}}{\partial t}(T_0, k_0)\mathrm{d}T &\approx -\mathbf{e}\mathbf{e}^T\left(\mathbf{M}(T_0)\frac{\partial \boldsymbol{\xi}}{\partial k}(0, k_0)\mathrm{d}k + \mathbf{V}(T_0)\mathrm{d}k\right) \\
\boldsymbol{\xi}(T_0 + \mathrm{d}T, k_0 + \mathrm{d}k) - \boldsymbol{\xi}_0(0) &\approx (\mathbf{I} - \mathbf{e}\mathbf{e}^T)\left(\mathbf{M}(T_0)\frac{\partial \boldsymbol{\xi}}{\partial k}(0, k_0)\mathrm{d}k + \mathbf{V}(T_0)\mathrm{d}k\right) \\
&= \mathbf{P}\left(\mathbf{M}(T_0)\frac{\partial \boldsymbol{\xi}}{\partial k}(0, k_0)\mathrm{d}k + \mathbf{V}(T_0)\mathrm{d}k\right)
\end{aligned}$$

And because of

$$\begin{aligned}\boldsymbol{\xi}(T_0+\mathrm{d}T,k_0+\mathrm{d}k)-\boldsymbol{\xi}_0(0) &= \boldsymbol{\xi}(0,k_0+\mathrm{d}k)-\boldsymbol{\xi}_0(0)\\ &\approx \frac{\partial\boldsymbol{\xi}}{\partial k}(0,k_0)\mathrm{d}k=\mathbf{P}\frac{\partial\boldsymbol{\xi}}{\partial k}(0,k_0)\mathrm{d}k\end{aligned}$$

I get

$$\begin{aligned}\mathbf{P}\left(\mathbf{M}(T_0)\frac{\partial\boldsymbol{\xi}}{\partial k}(0,k_0)\mathrm{d}k+\mathbf{V}(T_0)\mathrm{d}k\right) &\approx \mathbf{P}\frac{\partial\boldsymbol{\xi}}{\partial k}(0,k_0)\mathrm{d}k\\ \frac{\partial\boldsymbol{\xi}}{\partial k}(0,k_0)\mathrm{d}k &= \left[\mathbf{I}-\mathbf{P}\mathbf{M}(T_0)\right]_H^{-1}\mathbf{P}\mathbf{V}(T_0)\mathrm{d}k\end{aligned}$$

Where $[\mathbf{A}]_H^{-1}$ is the inverse of the matrix $\mathbf{A}$ restricted to the hyperplane H.

If I use the last equation in the derivation of $\mathrm{d}T$, I find

$$\frac{\partial T}{\partial k}\approx-\frac{\mathbf{e}^T\left(\mathbf{M}(T_0)[\mathbf{I}-\mathbf{P}\mathbf{M}(T_0)]_H^{-1}\mathbf{P}\mathbf{V}(T_0)+\mathbf{V}(T_0)\right)}{\left\|\frac{\partial\boldsymbol{\xi}}{\partial t}(0,k_0)\right\|}$$

Which can also be expressed with $\mathbf{k}$ being a vector with $\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t}=\mathbf{F}(\mathbf{x}(t),\mathbf{k})$ and $\mathbf{V}$ being a matrix as

$$\frac{\partial T}{\partial \mathbf{k}}\approx-\frac{\mathbf{e}^T\left(\mathbf{M}(T_0)[\mathbf{I}-\mathbf{P}\mathbf{M}(T_0)]_H^{-1}\mathbf{P}\mathbf{V}(T_0)+\mathbf{V}(T_0)\right)}{\left\|\frac{\partial\boldsymbol{\xi}}{\partial t}(0,\mathbf{k}_0)\right\|}$$

With the last equation, the robustness quantifier ρ_S can be calculated. In practice, I numerically integrate the variational equations for $\mathbf{M}(t)$ and $\mathbf{V}(t)$ over one period using MATLAB, which allows us to estimate $\frac{\partial T}{\partial\mathbf{k}}$. To increase the precision of this estimate, I start the integration at different points of the cycle and average the results. Finally, I use the following relation to calculate $\rho_S(\mathbf{k})$:

$$\frac{\partial T}{\partial\mathbf{k}}\frac{\mathbf{k}}{T}=\frac{\frac{\partial T}{T}}{\frac{\partial\mathbf{k}}{\mathbf{k}}}=\frac{\partial\log(T)}{\partial\log(\mathbf{k})}$$

Chapter 4

Results

In this chapter, different applications of my glocal analysis are presented. The first five sections have in common that the robustness analyses are applied to oscillatory systems. In particular, I assessed the effects of the feedback loops on different properties of the systems. First, I used a combination of global and local methods to quantify the robustness of two different models of the cyanobacterial circadian clock. This study led to a comparison of both models that discriminated them [134]. Second, the glocal approach was used to study the relation between entrainment robustness on two models of the *Drosophila* circadian clock with different architectures [136]. Third, following the ideas of neutral space [45], a generic model of the mitotic cycle was used to apply an evolution algorithm to show that the addition of a feedback loop can occur without disturbance of the oscillatory properties [137]. This analysis was completed by the application of the two-stage sampling algorithm to study the robustness of the different architectures [135]. Fifth, I will present the work done in collaboration with Didier Gonze from the Université Libre de Bruxelles, where we showed that the addition of a positive feedback loop enhances the robustness to molecular noise of a negative feedback oscillator [138].

The last section is dedicated to synthetic biology: robustness was used as a design criterion to build a circuit that allows maintenance of the homeostasis of two cell populations. In collaboration with Miles Miller and under the supervision of Ron Weiss from MIT, we propose a design procedure for synthetic circuits and optimization directions for the designed systems [139].

4.1 Robustness Analysis of Two Models of the Cyanobacterial Circadian Oscillators

To illustrate the application of the glocal method, I focus on two recent models of the *in vitro* cyanobacterial circadian oscillator [87, 90]. I chose this study system for several reasons. First, it is an area of very active recent model development, [87, 90, 92, 89, 88], driven by recent insights into the molecular mechanisms of the oscillator [81]. Second, the behavior or function of circadian oscillators is well-characterized: it has an ample oscillation with a period of approximately 24 hours [78], and low sensitivity to non-periodic environmental perturbations. Third, *in vitro* and *in vivo* experiments show that the cyanobacterial circadian clock is robust to many perturbations [165, 166]. Fourth, good estimates for the *in vivo* abundance of all involved proteins and of the cell volume for the cyanobacteria are available. Finally, being protein-based, the clock shares many features with signal transduction pathways, an important field of application for robustness analysis [167, 36].

4.1.1 Two Oscillator models of the Cyanobacterial Clock

The first model [87] (Figure 4.1A and equations (4.1)) involves complex formation of KaiC with the other proteins, as well as cyclic phosphorylation and desphosphorylation of KaiC. In this model, KaiA first binds to KaiC (top reaction of Figure 4.1A). The resulting complex KaiAC catalyzes the phosphorylation of KaiC forming KaiAC*. A central element of this model is that KaiAC* then exerts a positive feedback on its own formation (red arrow in Figure 4.1A). In a subsequent step, KaiB binds to the complex KaiAC* and inhibits this autocatalysis. To complete the cycle, KaiA is released, followed by KaiB, and finally KaiC* is dephosphorylated. I will refer to this model as the 'autocatalytic model'. It contains 8 states variables and 7 reactions with 7 individual parameters. In the equation system below, square brackets denote concentrations, and names inscribed in these brackets denote (phospho)proteins or complexes thereof.

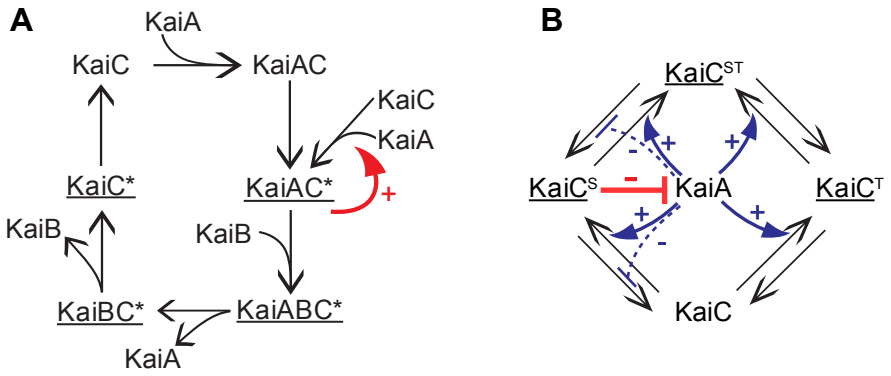

Figure 4.1: Two models of the *in vitro* cyanobacterial circadian cycle. (A) Autocatalytic model from Mehra et al. [87]. 'C*' stands for phosphorylated KaiC. The cycle proceeds clockwise, starting from the upper left. The sum of concentrations of the KaiC*-containing complexes (underlined) form the output of the model. The red arrow denotes the autocatalytic effect of KaiAC* on its synthesis. (B) Two phosphorylation sites model from Rust et al. [90]. There are three possible phosphorylated states for KaiC: KaiC^T, KaiC^S and KaiC^{ST}. The sum of concentrations of phosphorylated KaiC molecules (underlined) is the output of the system. KaiA catalyzes phosphorylation reactions (solid blue arrows) and inhibits some dephosphorylation reactions (dashed blue bars). KaiC^S (complexed with KaiB, not explicitly modeled) inhibits the action of KaiA (red bar).

$$
\begin{aligned}
\frac{\mathrm{d}[\text{KaiA}]}{\mathrm{d}t} &= \mathrm{k}_5[\text{KaiABC}^*] - \mathrm{k}_1[\text{KaiA}][\text{KaiC}] - \mathrm{k}_3[\text{KaiAC}^*][\text{KaiA}][\text{KaiC}] \\
\frac{\mathrm{d}[\text{KaiB}]}{\mathrm{d}t} &= \mathrm{k}_6[\text{KaiBC}^*] - \mathrm{k}_4[\text{KaiAC}^*][\text{KaiB}] \\
\frac{\mathrm{d}[\text{KaiC}]}{\mathrm{d}t} &= \mathrm{k}_7[\text{KaiC}^*] - \mathrm{k}_1[\text{KaiA}][\text{KaiC}] - \mathrm{k}_3[\text{KaiAC}^*][\text{KaiA}][\text{KaiC}] \\
\frac{\mathrm{d}[\text{KaiC}^*]}{\mathrm{d}t} &= \mathrm{k}_6[\text{KaiBC}^*] - \mathrm{k}_7[\text{KaiC}^*] \\
\frac{\mathrm{d}[\text{KaiAC}]}{\mathrm{d}t} &= \mathrm{k}_1[\text{KaiA}][\text{KaiC}] - \mathrm{k}_2[\text{KaiAC}] \\
\frac{\mathrm{d}[\text{KaiAC}^*]}{\mathrm{d}t} &= \mathrm{k}_2[\text{KaiAC}] + \mathrm{k}_3[\text{KaiAC}^*][\text{KaiA}][\text{KaiC}] - \mathrm{k}_4[\text{KaiAC}^*][\text{KaiB}] \\
\frac{\mathrm{d}[\text{KaiBC}^*]}{\mathrm{d}t} &= \mathrm{k}_5[\text{KaiABC}^*] - \mathrm{k}_6[\text{KaiBC}^*] \\
\frac{\mathrm{d}[\text{KaiABC}^*]}{\mathrm{d}t} &= \mathrm{k}_4[\text{KaiAC}^*][\text{KaiB}] - \mathrm{k}_5[\text{KaiABC}^*]
\end{aligned}
\tag{4.1}
$$

To initialize the iterative exploration of the parameter space, the parameter vector reported in [87] is used: $\mathbf{k} = [10^{-4}\ \text{mol}^{-1}\text{h}^{-1},\ 0.40\ \text{h}^{-1}, 0.45\ \text{M}^{-2}\text{h}^{-1},$

3.65 h^{-1}, 4.00 h^{-1}, 0.90 h^{-1}, 0.18 h^{-1}]. The total concentrations of the relevant molecules are $[\Sigma\text{KaiA}] = 3.0\ \mu M$, $[\Sigma\text{KaiB}] = 1.0\ \mu M$ and $[\Sigma\text{KaiC}] = 3.5\ \mu M$.

The second model [90] (Figure 4.1B and equations (4.2)) makes a distinction between the two phosphorylation sites S and T of KaiC [91], resulting in three possible phosphorylated states: KaiC^T, KaiC^S and KaiC^{ST} (see section 1.4.2). KaiA catalyzes the phosphorylation of both S and T sites and inhibits the dephosphorylation of KaiC^{ST} and KaiC^S. These actions of KaiA are inhibited by KaiC^S (red bar in Figure 4.1B). Although KaiC^S exerts its effects on KaiA jointly with KaiB [83], KaiB does not appear in the equations, because it is assumed to be at saturation level in this model. I will refer to this model as the 'two (phosphorylation) sites model'. It contains 4 states variables and 8 reactions with 12 parameters [90] and [KaiA] is expressed as a function of $[\text{KaiC}^S]$:

$$
\begin{aligned}
[\text{KaiA}]\left([\text{KaiC}^S]\right) &= [\text{KaiA}] = \max\left\{0, [\Sigma\text{KaiA}] - 2[\text{KaiC}^S]\right\} \\
[\text{KaiC}] &= [\Sigma\text{KaiC}] - [\text{KaiC}^T] - [\text{KaiC}^{ST}] - [\text{KaiC}^S] \\
\frac{\mathrm{d}[\text{KaiC}^T]}{\mathrm{d}t} &= \frac{\mathrm{k}_4[\text{KaiA}]}{\mathrm{k}_{12} + [\text{KaiA}]}[\text{KaiC}] + \frac{\mathrm{k}_9[\text{KaiA}]}{\mathrm{k}_{12} + [\text{KaiA}]}[\text{KaiC}^{ST}] \\
&\quad - \left(\mathrm{k}_1 + \frac{\mathrm{k}_8[\text{KaiA}]}{\mathrm{k}_{12} + [\text{KaiA}]}\right)[\text{KaiC}^T] - \frac{\mathrm{k}_5[\text{KaiA}]}{\mathrm{k}_{12} + [\text{KaiA}]}[\text{KaiC}^T] \\
\frac{\mathrm{d}[\text{KaiC}^{ST}]}{\mathrm{d}t} &= \frac{\mathrm{k}_5[\text{KaiA}]}{\mathrm{k}_{12} + [\text{KaiA}]}[\text{KaiC}^T] + \frac{\mathrm{k}_6[\text{KaiA}]}{\mathrm{k}_{12} + [\text{KaiA}]}[\text{KaiC}^S] \\
&\quad - \left(\mathrm{k}_2 + \frac{\mathrm{k}_{10}[\text{KaiA}]}{\mathrm{k}_{12} + [\text{KaiA}]}\right)[\text{KaiC}^{ST}] - \frac{\mathrm{k}_9[\text{KaiA}]}{\mathrm{k}_{12} + [\text{KaiA}]}[\text{KaiC}^{ST}] \\
\frac{\mathrm{d}[\text{KaiC}^T]}{\mathrm{d}t} &= \left(\mathrm{k}_2 + \frac{\mathrm{k}_{10}[\text{KaiA}]}{\mathrm{k}_{12} + [\text{KaiA}]}\right)[\text{KaiC}^{ST}] + \frac{\mathrm{k}_7[\text{KaiA}]}{\mathrm{k}_{12} + [\text{KaiA}]}[\text{KaiC}] \\
&\quad - \left(\mathrm{k}_3 + \frac{\mathrm{k}_{11}[\text{KaiA}]}{\mathrm{k}_{12} + [\text{KaiA}]}\right)[\text{KaiC}^S] - \frac{\mathrm{k}_6[\text{KaiA}]}{\mathrm{k}_{12} + [\text{KaiA}]}[\text{KaiC}^S]
\end{aligned}
\tag{4.2}
$$

To initialize the iterative exploration of parameter space, I use the parameter vector reported in [90]: $\mathbf{k}$ = [0.21 h^{-1}, 0.31 h^{-1}, 0.11 h^{-1}, 0.4791 h^{-1}, 0.2129 h^{-1}, 0.5057 h^{-1}, 0.0532 h^{-1}, 0.7985 h^{-1}, 0.173 h^{-1}, -0.3194 h^{-1}, -0.1331 h^{-1}, 0.43 M]. The total concentrations are $[\Sigma\text{KaiA}] = 1.3\mu M$ and $[\Sigma\text{KaiC}] = 3.4\mu M$.

Both models capture important empirical observations about the cyanobacterial circadian cycle: phosphorylation of KaiC with the help of KaiA [91],

inhibition of this effect by KaiB when bound to phosphorylated KaiC [91, 83], and finally dephosphorylation to complete the cycle [91]. However, the models are also fundamentally different in some key assumptions about the underlying mechanism. Because of these dramatic differences, biochemical data will play a decisive role in model discrimination. The robustness analysis carried out in this work is a first step towards such validation.

4.1.2 The Two-Sites Model Shows Greater Global Robustness

In applying my global approach to both models, I sampled parameter vectors covering a range of six orders of magnitude for each parameter, centered on published parameter values for both models (see above) using the PCA sampling method (see section 3.1.1). In order to avoid biased estimates the interval bounds should be within biophysical feasibility. Such an *a priori* range needs to be established, both for practical reasons, and for models that are unidentifiable [154, 155, 57]. For the systemic properties $\boldsymbol{\pi}$, I chose the period, the peak value and the amplitude. The bounds of the viable range $[\underline{\boldsymbol{\pi}}, \overline{\boldsymbol{\pi}}]$ are chosen [79] to be 10% below and above the respective values of the properties $\boldsymbol{\pi}$ of each model with the parameter vector defined above [87, 90]. With these constraints, I carried out the sampling procedure for ten PCA iterations. For each iteration, I sampled 10^5 parameter vectors uniformly, and used the viable parameters of the last four iterations to define the hyperbox for the Monte Carlo integration.

Figure 4.2A shows the (normalized) viable volumes R for the two models. These volumes can be interpreted as the average allowable variation per parameter that leaves the circadian oscillations intact. The two-sites model is vastly more robust than the autocatalytic model. Specifically, the value $R = 0.718$ for the autocatalytic model means that the parameters can vary over 0.7 orders of magnitude, or 5.2-fold. For the two-sites model, the value of $R = 1.60$ is more than twice that, correspond to a 39-fold allowable variation. The values shown are based on at least 5×10^4 parameter vectors and have sampling errors of less than one percent (see section 3.1.3). I also note that the estimated viable parameter volumes were highly reproducible among five independent applications of the iterative procedure. For example, the mean values of $R = 1.60 \pm 0.01$ (two-sites model) and $R = 0.718 \pm 0.006$ (autocatalytic model) have a coefficient of variation below one percent over these five iterations, which shows that the PCA-guided sampling approach gives highly reproducible results.

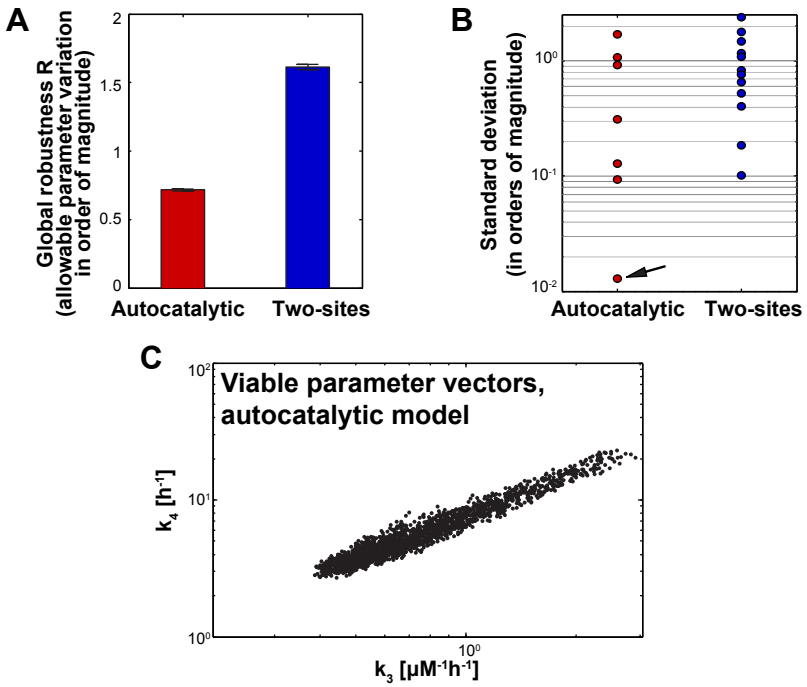

Figure 4.2: Results of the global robustness analyses for both models. (A) The two-sites model (right) has significantly greater nomalized viable volume than the autocatalytic model (left). Error bars ($< 1\%$) correspond to standard deviations over five independent estimates. (B) Standard deviations along the principal axes of viable parameters for the autocatalytic model and the two-sites model. Note the logarithmic scale. The autocatalytic model has a strongly constrained axis (arrow); amounts of variation along the other axes are overall smaller for the autocatalytic model. (C) Projection of the viable vectors of the autocatalytic model after the MC integration on the plane (k_3, k_4). These two parameters are strongly correlated resulting in the lowest standard deviation for the autocatalytic model (B).

What is responsible for the lower robustness of the autocatalytic model? One possibility is that strong associations exist between individual parameters in viable parameter sets, such that some parameters cannot vary independently from others. Such associations, if present, may also provide mechanistic insights into complex, high-dimensional circuits. Figure 4.2B shows the standard deviations of viable parameters along the principal axes of both models. With one exception, the amount of variation along most principal component axes is similar for both models. The exception (indicated by the arrow in the Figure 4.2B) is the lowest PCA axis for the autocatalytic model.

The high constraint on variation in this axis is caused by a strong positive correlation between the rate for the autocatalytic reaction, parameter k_3, and

the rate for the formation of the complex KaiABC*, k_4 (Figure 4.2C). This axis deviates by merely 13 degrees from the vector $\mathbf{k} = (0, 0, 1, -1, 0, 0, 0)$ defined by the model's parameters. Parameters k_3 and k_4 are highly correlated (Pearson's r = 0.97, significance of all statistical tests are summarized in Table 4.1). This strong association contributes to the lack of global robustness observed in the autocatalytic model. It means that a perturbation of parameter k_3 that would not be followed by a corresponding perturbation in parameter k_4 would prevent the model to preserve properties $\boldsymbol{\pi}$ of interest. Examining the structure of the equations for the autocatalytic model (Figure 4.2A), I find that the mechanistic cause for this association lies in the dynamics of KaiAC*: on one hand, if k_3 is too large, the concentration of KaiAC* increases too fast and the autocatalytic effect is too strong; on the other hand, if k_4 is too large, the concentration of KaiAC* is too low and the autocatalytic effect is too weak. The parameters k_3 and k_4 need to be delicately balanced to have the correct concentration of KaiAC* resulting in the appropriate feedback strength.

To assess whether this strong association is responsible for the smaller global robustness of the autocatalytic model, I collapsed the highly correlated parameters k_3 and k_4 into one. That is, I assumed that k_3 and k_4 are linearly dependent and can be considered as one single parameter. The reduced model with only six parameters yields a global robustness estimate of $R = 1.09$. This corresponds to an allowable 12-fold average variation of each parameter, and accounts partially for the lower robustness of the autocatalytic model.

4.1.3 The Two-Sites Model Shows Greater Local Robustness

Figure 4.3A shows the distribution of ρ_P, the quantifier of robustness to local parametric perturbations for both the autocatalytic model and the two-sites model. The median robustness of the autocatalytic model is lower by 29% (median $\rho_P = 0.179$ and $\rho_P = 0.231$ for the autocatalytic and two-sites model, respectively; see table 4.1 for significance).

My combination of global and local analysis allows us to ask whether individual chemical reactions (represented through their parameters) are particularly important for a model's robustness. To this end, I investigated whether there exist statistical associations between ρ_P and any of the model parameters. One striking such association stands out for the autocatalytic model (Figure 4.4A). Specifically, ρ_P is highly associated with k_7, (Spearman's r = -0.638), whereas all other parameters and ρ_P show only r < 0.11 (Spearman's partial correlation given k_7). A glance at the model equations (Eq. 4.1) shows that

Null hypothesis	Test type	r-value	p-value	n
Parameters k_3 and k_4 are correlated in the autocatalytic model	Pearson's	0.97	$< 10^{-323}$	1828
ρ_P is larger for two-sites model	Wilcoxon rank		3.32×10^{-91}	
ρ_P correlated with k_7 for autocatalytic model	Spearman's	-0.638	$< 10^{-323}$	1828
robustness to temperature changes is larger than ρ_P for autocatalytic model	Wilcoxon rank		1.95×10^{-246}	
robustness to temperature changes is larger than ρ_P for two-sites model	Wilcoxon rank		0.245	
robustness to temperature changes is larger in the two-sites model	Wilcoxon rank		2.28×10^{-4}	
ρ_C is larger for two-sites model	Wilcoxon rank		9.25×10^{-177}	
ρ_C correlated with k_7 for autocatalytic model	Spearman's	-0.718	2.81×10^{-289}	1828
ρ_N is larger for two-sites model	Wilcoxon rank		3.09×10^{-239}	
ρ_N correlated with k_1 for autocatalytic model	Spearman's	0.921	$< 10^{-323}$	1828
ρ_N correlated with k_2 for two-sites model	Spearman's	0.629	$< 10^{-323}$	604
ρ_N correlated with k_3 for two-sites model	Spearman's	0.414	$< 10^{-323}$	604
ρ_A is larger for two-sites model	Wilcoxon rank		4.31×10^{-10}	
ρ_S is larger for two-sites model	Wilcoxon rank		1.69×10^{-151}	
ρ_T is larger for two-sites model	Wilcoxon rank		1.48×10^{-238}	
ρ_T is correlated with the distance from the parameter with the highest ρ_T for autocatalytic model	Spearman's	-0.355	$< 10^{-323}$	1828
ρ_T is correlated with the distance from the parameter with the highest ρ_T for two-sites model	Spearman's	-0.196	$< 1.15 \times 10^{-6}$	604

Table 4.1: Statistical tests and their significance used to assess model discrimination and correlations.

the reaction associated with k_7 dephosphorylates KaiC* and thus triggers the initialization of a new autocatalytic cycle. If this initialization occurs too fast (at large k_7), synchronization of complex formation and absorption of perturbations is poor.

As an extension of this quantifier, properly correlated parametric perturbations are used to address the robustness to temperature changes (see section 3.2.1). First, I notice that the sensitivity of the period to parameters shows that for the autocatalytic model, α_4 is positive and for the two-sites model $\alpha_3, \alpha_6, \alpha_9, \alpha_{10}$ are positive, therefore temperature compensation is possible in both models. With the finite correlated perturbations, I find that the two-sites model has a median robustness only 4% greater than the autocatalytic model

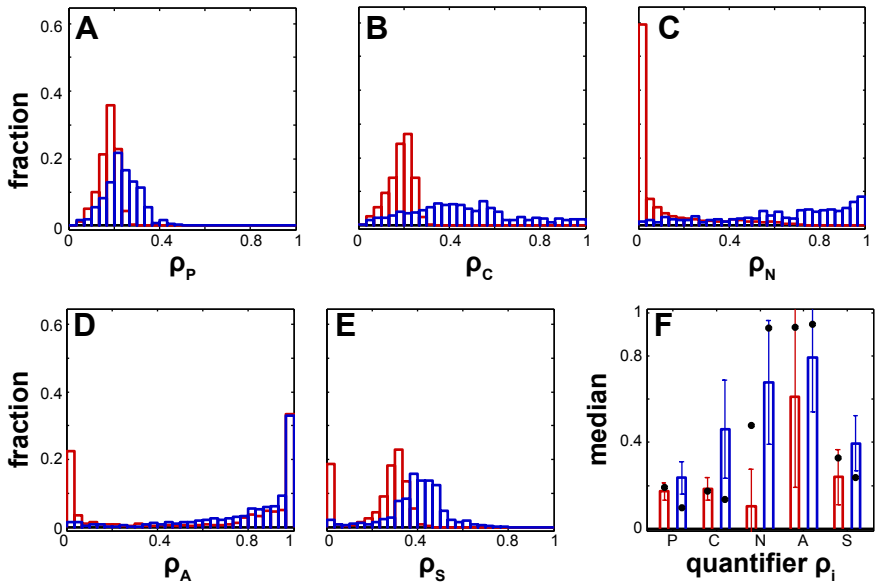

Figure 4.3: The two-sites model (blue) has greater local robustness than the autocatalytic model (red). Shown are the distributions of (A) robustness to local parameter perturbations ρ_P, (B) robustness to total concentration perturbations ρ_C, (C) robustness to molecular noise ρ_N, (D) attraction of the cycle ρ_A, and (E) sensitivity of the period ρ_S. In (F) median values are shown with their associated standard deviation (error bars) for both models and all five quantifiers. Black dots indicate local robustness values for the previously published parameter vector's [87, 90].

(Figure 4.5B). Individual analyses of both models show why this difference, yet significant ($p = 2.28 \times 10^{-4}$), is small compared to the difference in ρ_P. On the first hand, the autocatalytic model is more robust to such correlated perturbations than to uncorrelated perturbations (median of 0.230, and 0.179, respectively). The large difference between the two cases for the autocatalytic model (Figure 4.5A and 4.5B, red bars) can be explained by the strong association between k_3 and k_4 discussed above: correlated perturbations cannot be aligned with the most constrained direction of the viable parameter volume. On the other hand, the two-sites model, which does not have such highly associated parameters, does not show increased robustness to correlated parameter changes ($p = 0.245$).

I next turn to total concentration perturbations ρ_C (distribution shown in Figure 4.3B. Here, the two-sites model is on average 2.5-fold more robust than the autocatalytic model, with a median $\rho_C = 0.192$ and $\rho_C = 0.439$ for the autocatalytic and two-sites model, respectively. For instance, for 10% of viable parameter vectors in the two-sites model, more than 80% of perturbations leave

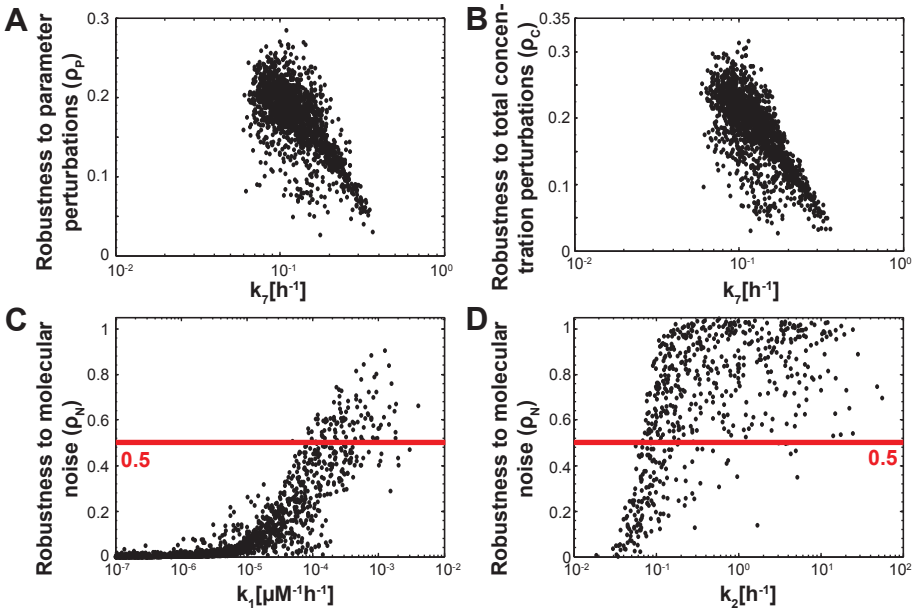

Figure 4.4: Correlations of the local robustness quantifiers with model parameter. (A) Parameter k_7 (horizontal axis) negatively affects robustness to parameter perturbations (vertical axis) in the autocatalytic model (Spearman's r = −0.638). (B) Parameter k_7 (horizontal axis) negatively affects robustness to concentration perturbations (vertical axis) in the autocatalytic model (Spearman's r = −0.718). (C) Score for robustness to molecular noise for the autocatalytic model plotted against k_1 and (D) the two-sites models plotted against k_2. In the autocatalytic model, k_1 has a Spearman's correlation coefficient with ρ_N of 0.921 and less that 6 percent of the parameter vectors have a score above 0.5. For the two-sites model, k_2 has a correlation coefficient of 0.629 with ρ_N and more than 80 percent of the parameter vectors have a score above 0.5.

the circadian oscillation intact. Exactly as for ρ_P, I find that in the autocatalytic model, k_7 strongly influences ρ_C (Figure 4.4B), with a Spearman's rank correlation between k_7 and ρ_C of −0.718, which underscores the importance of this dephosphorylation reaction.

I next assessed robustness ρ_N to molecular noise. To this end, I used Gillespie's algorithm [15] to simulate an oscillator with 2000-6000 molecules in a reaction volume of 3μl, numbers that are of the correct order of magnitude for the number of Kai proteins in a cyanobacterial cell [168] (the exact numbers of molecules for the simulations are 5420 for KaiA, 1807 for KaiB, and 6323 for KaiC for the autocatalytic model and 2349 molecules of KaiA and 6142 molecules of KaiC for the two-sites model). When changing the cellular volume, the results stay qualitatively the same (results not shown). Here again, the

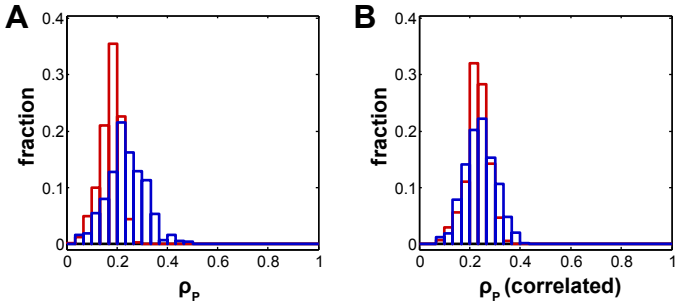

Figure 4.5: Robustness to temperature change. (A) distribution of the scores for the robustness to parameter perturbations (autocatalytic model in red and two-sites model in blue), similar as Figure 4.3B. (B) distribution of the scores for the robustness to temperature changes. The results are obtained with the same algorithm as the one for ρ_P but the random variates are correlated such that for a particular perturbation all parameters are either increased or decreased. In this case, the median robustness for the two-sites model is only 4 percent larger than the median of the autocatalytic model.

two-sites model is significantly more robust, with a median (mean) value of ρ_N that is 45 (6.5) times larger (Figures 4.3C and 4.3F). For example, for the autocatalytic model, fewer than 6% of viable parameter vectors show $\rho_N > 0.5$ (Figure 4.4C), whereas more than 80% of the parameters show $\rho_N > 0.5$ in two-sites model, where noise also affects only a small region of the viable parameter volume (Figure 4.4D).

Figure 4.4C plots, for the autocatalytic model, ρ_N against the model parameter k_1 that is most highly correlated with it (Spearman r = 0.921). The Figure shows that $\rho_N < 0.5$ for more than half the range of viable parameters. All other parameters show a partial Spearman rank correlation with ρ_N (controlling for k_1) lower than r < 0.35. Parameter k_1 governs the rate of KaiAC complex formation. Its importance can be explained by the disproportionate effect of k_1 decrease as it is already low. For example, when k_1 is equal to 4×10^{-5}, this complex forms at an average rate of 0.76h^{-1} with the given copy numbers. If k_1 decreases modestly to 10^{-5}, this rate decreases to 0.19h^{-1} or one complex formation every five hours. Because this reaction starts the cycle, the fluctuations in its rate can spread and strongly affect the period.

In the two-sites model, the parameters most highly correlated with ρ_N are k_2 and k_3 (Spearman r = 0.629, respectively, $p < 10^{-323}$, n = 604). All other parameters show a partial Spearman rank correlation with ρ_N (controlling for k_2 and k_3) lower than r < 0.12. Figure 4.4D shows a scatterplot of ρ_N against model parameter k_2 over the entire range of viable parameters that

spans four orders of magnitude. With a few exceptions, ρ_N is smaller than 0.5 only for one quarter of this range. Parameters k_2 and k_3 represent the rates of the dephosphorylation reactions KaiCST →KaiC^{S} and KaiC^{S} →KaiC, respectively. KaiC^{S} is a critical element of the negative feedback loop (red bar in Figure 4.1B) that inhibits the action of KaiA. This observation explains the importance of the reactions that form and destroy KaiC^{S}. Low values of k_2 combined with finite numbers of reacting molecules lead to greater noise in the formation of KaiC^{S} from KaiCST. Its concentration will thus fluctuate to a greater extent, and these fluctuations can then be further amplified by feedback.

I next turn to the attraction of the cycle ρ_A, whose distribution is shown in Figure 4.3D. The two-sites model has a significantly higher median $\rho_A = 0.891$ compared to $\rho_A = 0.846$ for the autocatalytic model. An analogous difference holds for period sensitivity (Figure 4.3E), where ρ_S is on average 65 percent greater in the two-sites model.

I had noted previously that k_3 and k_4 are strongly and negatively associated with global robustness. When analyzing their association with period sensitivity, I find that they also have a strong and opposite impact on the period ($\alpha_3 = \partial \log(T)/\partial \log(k_3)$ is negative, whereas α_4 is positive). The reason is the same as discussed in the results for global robustness, namely that the autocatalytic feature that is so central to this model requires a delicate balance of two reactions producing and destroying KaiAC*. This feature also explains the higher robustness to temperature compensation as discussed above.

A remaining question regards the relationship between robustness quantifiers $\boldsymbol{\rho}$ and the center of the set of viable parameters. Naively, one might assume that robustness might be highest in this center, and that the value of any robustness quantifier decreases from this center. However, this is not generally the case (Figure 4.6A-E). Specifically, only 3 and 2 out of the five robustness quantifiers show this expected distance dependency for the autocatalytic and two-sites models, respectively, and none of these associations exceed r = 0.25.

Average local robustness ρ_T is also not higher at the center (Figure 4.6F). The same holds if ρ_T is calculated through multiplication rather than averaging (not shown). In addition, there exist regions of parameter space that have higher average robustness ρ_T (Figure 4.6G) than the center, and there is a negative association between a parameter vector's ρ_T and its distance to the parameter vector with the highest ρ_T. This association is higher for the

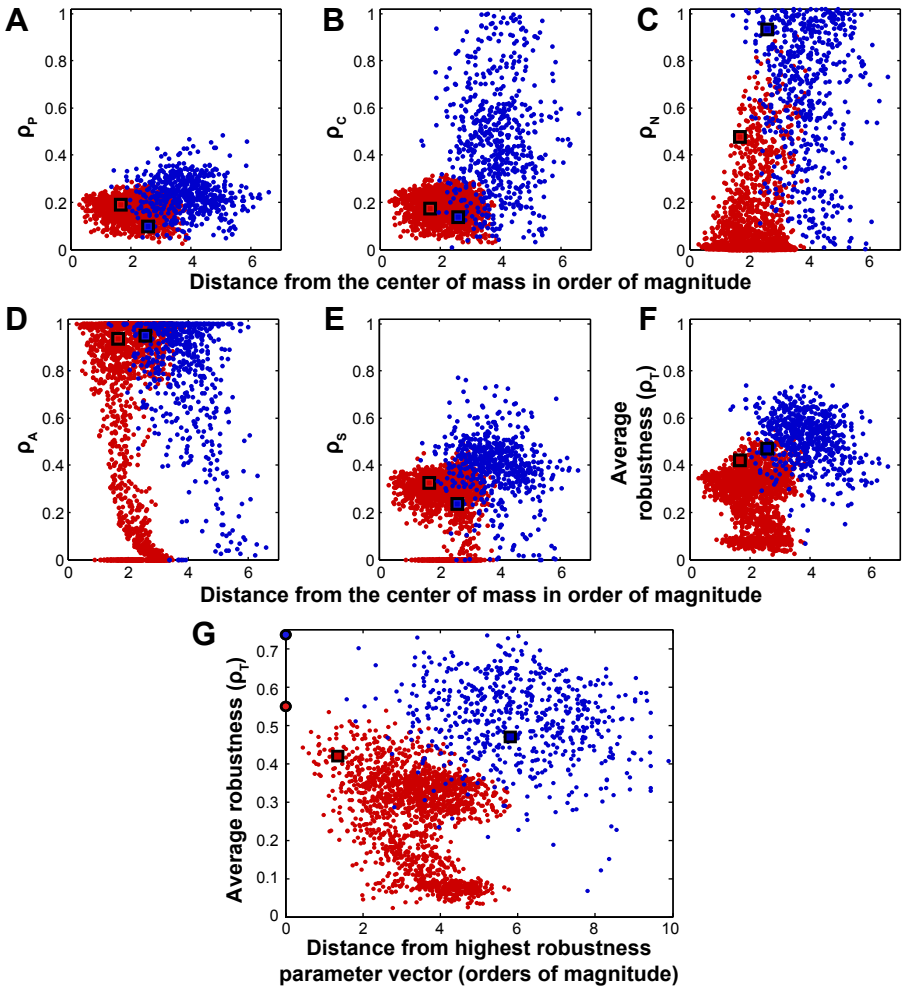

Figure 4.6: Distribution of the average local robustness ρ_T for the two models. Local robustness is not maximal in the center of the viable parameter set. Shown are local robustness quantifiers (vertical axes) plotted against the distance of viable parameter vectors (horizontal axes) from the center of mass of the entire set of viable parameter vectors, for the autocatalytic model (red) and the two-sites model (blue): (A) robustness ρ_P to parameter perturbations, (B) robustness ρ_C to total concentration perturbations, (C) robustness ρ_N to molecular noise, (D) attraction of the cycle ρ_A, (E) sensitivity of the period ρ_S, (F) total robustness, ρ_T, defined as the arithmetic mean over all five robustness quantifiers. Maxima for all local robustness quantifiers are not found near the center of mass, and there is only a weak correlation between a parameter vectors distance from this center and local robustness. For each viable parameter vector $\mathbf{k}$, the figure shows its distance (horizontal axis) from the viable parameter vector with the highest average local robustness ρ_T plotted against the ρ_T of $\mathbf{k}$ (vertical axis). Large circles correspond to the two parameter vectors with the highest ρ_T for each model. The greater the distance of $\mathbf{k}$ to the most robust parameter vector, the lower its ρ_T. This negative association is stronger for the autocatalytic model. The two squares in each panel correspond to previously published parameter vectors for each model [87, 90].

autocatalytic model (see table 4.1). These observations further underscore the higher robustness of the two-sites model. To summarize: first, not surprising, the average ρ_T of all five quantifiers indicates much greater robustness for the two-sites model. Second, the spatial distributions of ρ_T show that in the two-sites model, robustness decreases more slowly with distance from the points of highest average local robustness reflecting a larger volume with high average robustness.

4.1.4 Connectivity in the Parameter Space

Following the study of the spatial distribution of robustness, I address the question whether the viable region of parameter space forms a connected set. Such connectedness would facilitate the evolution of oscillators with high robustness through gradual changes of individual parameters. For any high dimensional model whose governing equations cannot be solved analytically, this is perhaps the most difficult problem in global analysis, for the following reason. The sample of viable parameters that our approach generates, albeit large, is finite, and comes from a multidimensional parameter space with uncountably many elements. In this parameter space, the set of all viable parameters may be connected, or, alternatively, it may be fragmented into many small islands of viable subsets. No sampling approach can prove which of these extremes (or a spectrum of possibilities in between) is the case. However, sampling, along with the continuous properties of the ODE system, can provide a hint as to which of these scenarios is closer to the truth. To this end I define a graph whose nodes are parameter vectors in the viable set, and where an edge connects two nodes if a straight line exists that preserves the oscillatory behavior for all points along the line. To determine whether two parameter vectors, say $\underline{\mathbf{k}}$ and $\overline{\mathbf{k}}$, are neighbors in this graph, I sampled a convex combination $\mathbf{k}^{(i)}$ of M uniformly distributed points in the logarithmic domain of points between $\underline{\mathbf{k}}$ and $\overline{\mathbf{k}}$:

$$\log_{10}(\mathbf{k}^{(i)}) = \left(1 - \frac{i}{M+1}\right) \log_{10}(\underline{\mathbf{k}}) + \frac{i}{M+1} \log_{10}(\overline{\mathbf{k}}) \tag{4.3}$$

I assessed whether all of these M points preserved proper oscillatory behavior, i.e. $\boldsymbol{\pi}(\mathbf{k}^{(i)}) \in [\underline{\boldsymbol{\pi}}, \overline{\boldsymbol{\pi}}] \ \forall \ i = 1, \ldots, M$, which suggests that the straight line connecting $\underline{\mathbf{k}}$ and $\overline{\mathbf{k}}$ lies in its entirety in the viable set. Figure 4.7 shows a projection of the structure of the entire graph into the two-dimensional space defined by parameters k_5 and k_6 for the autocatalytic model, and into the two-dimensional space defined by parameters k_1 and k_2 for the two-sites model.

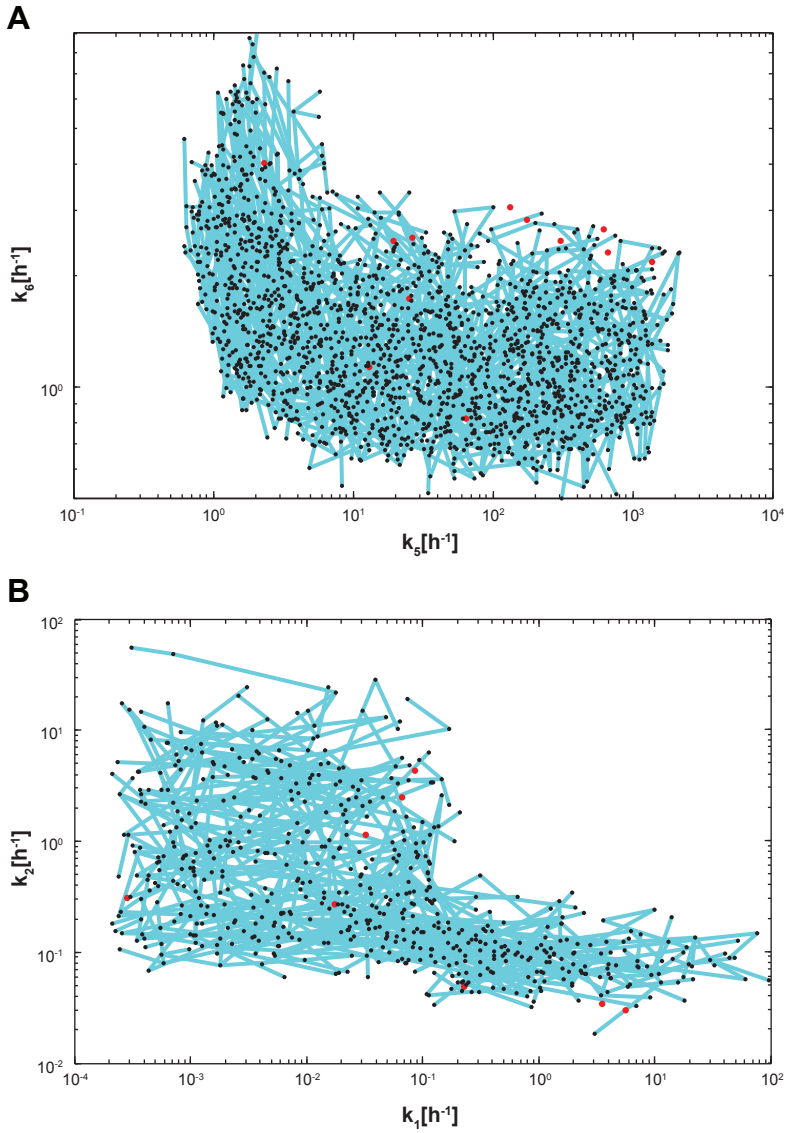

Figure 4.7: Viable parameter sets form large connected regions in parameter space. (A) autocatalytic model (projection on k_5 and k_6), (B) two-sites model. Pairs of viable parameter vectors (black dots) are connected by blue lines, if they are likely to be part of the same connected region of parameter space, as determined by numerical analysis explained in the text. Parameter vectors that cannot be connected to other parameter vectors are shown as red dots. The graph is shown as a projection on to the axes formed by k_5 and k_6 for (A), and as a projection onto the axes formed by k_1 and k_2 in (B), because these projections best illustrate that the viable region is not convex.

For both models, the graph consists of one giant component comprising the vast majority of parameter vectors, and few isolated nodes. Specifically, in the autocatalytic model, only 0.7% (12 of 1828) of parameter vectors are not in this connected component. In the two-sites model 1.3% (8 out of 604) of parameter vectors are not in this component. The isolated parameter vectors lie close to the boundary of the viable parameter volume. For these analyses, I chose to sample 10 points per order of magnitude change along each straight line connecting any two parameter vectors ($M = 10 \cdot \| \log_{10}(\underline{\mathbf{k}}) - \log_{10}(\overline{\mathbf{k}}) \|$), which corresponds to a 25% difference in parameter values between two successive sampling points. Increasing the density of the sampling did not affect these results qualitatively.

To summarize, even if the connectivity question cannot be answered rigorously, I show that there probably exists a giant component for both models. Therefore, the volume formed by these connected parameter vectors likely forms a 'neutral volume' [45] in which circadian oscillations with a given period and amplitude are preserved.

4.1.5 Conclusion

In my application of the glocal analysis to two circadian oscillator models, I found that the two-sites model shows greater global robustness than the autocatalytic model, with 39-fold and 5-fold allowable parameter variation, respectively, along each parameter dimension on average. Similarly, the two-sites model has higher local quantifier values for robustness to parameter changes, molecular noise, transient state perturbation, and period sensitivity. Based on these considerations alone, the architecture of the two-sites model is superior to the one of the autocatalytic model. If robustness is advantageous, and if this oscillatory mechanism is realizable biochemically [79, 165], it should be the preferred architecture. This observation is consistent with recent experiments that provide strong evidence in favor of ordered phosphorylation in the cyanobacterial clock [91, 169]. In contrast, the autocatalytic mechanism [87], obtained by interpreting experimental results of [168], whereas phosphorylated KaiC facilitates KaiA-KaiC association and subsequent KaiC phosphorylation, was not confirmed by recent experiments [91, 169, 90].

The glocal combination of global and local robustness analysis shows which chemical reactions in these models are of particular importance for robustness (or a lack thereof). For example, the rates of two central reactions of the autocatalytic loop in the autocatalytic model need to be delicately balanced, a prop-

erty that partially accounts for its lack of global robustness. Put differently, the central feature of this model is partly responsible for its low robustness. In the two-sites model, my local analysis shows that the rates of the reactions that form and destroy $KaiC^S$ are of particular importance for its robustness. For low values of these parameters, the concentration of $KaiC^S$ fluctuates to a greater extent. The resulting fluctuations are then amplified by the feedback loop central to this model.

4.2 Entrainment as a Selection Criteria for Circadian Cycles Robustness Analysis

The results of the previous section were based on *in vitro* experiments of the cyanobacterial circadian clock, which is a particular system as it is protein-based. It also lacks a good understanding of the entrainment mechanism, an essential feature of circadian clocks. In this section, I will present a research that complements the previous results. I will use two models of the *Drosophila* circadian clock and study their robustness properties using the glocal method focusing on the entrainment aspect. This work was done in collaboration with Pierre Sacré from the group of Prof. Rodolphe Sepulchre in the University of Liège, Belgium.

Identically to the previous section, the global approach identifies the viable region of the high-dimensional parameter space where both models display the experimentally observed behavior under entrainment (period of the cycle and phase of the peaks). In this case, the local analysis will be based on the phase response curve (PRC) that specifically analyzes the entrainment properties of the system. PRC analysis has proven to be a useful tool to study the input-output properties of oscillators [170]. It tabulates the phase shift at steady state oscillations that results from a particular input perturbation as a function of the phase at which this input is applied.

As a case study, we investigate two particular models of circadian clocks in *Drosophila* that have common features to many eukaryotic circadian clocks (see section 1.4.3). As described in the introduction (section 1.4.3), the first published model of the circadian rhythm in *Drosophila* comprises only the protein PER that forms a single feedback loop [104]. The protein is going through two successive phosphorylation steps before entering the nucleus and inhibiting its own expression. These intermediate stages induce a delay in the feedback, a necessary condition for oscillations [72]. A second model proposes an additional feedback loop through the action of the Timeless (TIM) protein. TIM acts in parallel to PER but its degradation is enhanced by light allowing the system to be entrained [105]. Further experimental studies have found the existence of another feedback loop with the Cycle and Clock proteins [171]. Other recently published models propose several new components [107], but we will focus on the first two systems which are generic models of moderate complexity for circadian clocks and are composed of either one [104] or two feedback loops [105] (figure 4.8).

With this work, we want to address two questions. First, we noticed that most of the studies on robustness of circadian cycles have been performed on the autonomous oscillations, i.e. without any external entrainment [48]. They are therefore based on the properties of the system in a dark environment, whereas the main feature of circadian clocks is their ability to be entrained by the daily light/dark rhythm. In this research, we want to overcome this shortcoming and try to understand how the oscillator architecture influences the robustness of a system in relation to its entrainment properties. Second, it is currently believed that the specific structure of the biological network is responsible for their dynamical behavior and their robust performance [151]. However, the actual relations between structure and robustness are still not clear. For example the purpose of the multiple feedback loops present in many circadian networks remains controversial [102]. One tentative explanation for the emergence of these, often parallel, control mechanisms is redundancy: Udea *et al.* [172] showed that adding a feedback loop in a circadian clock enhances the robustness of the system to point mutations. With this research on two models with respectively one and two feedback loops, we hope to bring a new argument for the advantage of additional feedback loops.

4.2.1 Two Models of the Drosophila Clock

For the one-loop model, we use the model and its parameters as published by Goldbeter [104]. The two-loop model published by Leloup and Goldbeter [105] was selected as a comparison. As no entrainment was implemented for the one-loop model, we changed this model by adding a light-modulation of the degradation of the PER protein as it is the case for the TIM protein in the two-loop model (see figure 4.8 and equations (B.1) in appendix). The parameters are the same as the ones used in the original paper. For this latter model (equations (B.2)), we performed two sampling procedures: one with the original constraint on the symmetry of the parameters in the two loops and one where this restriction was released (see appendix, section B). To this end, we obtain three models: one loop, two symmetric loops and two asymmetric loops. For all models, we use three different operating conditions that form a hierarchy: first, the sampling is done (1) under normal entrainment, then refined (2) by studying three cycles after interruption of the entrainment and finally (3) in dark conditions (free-run).

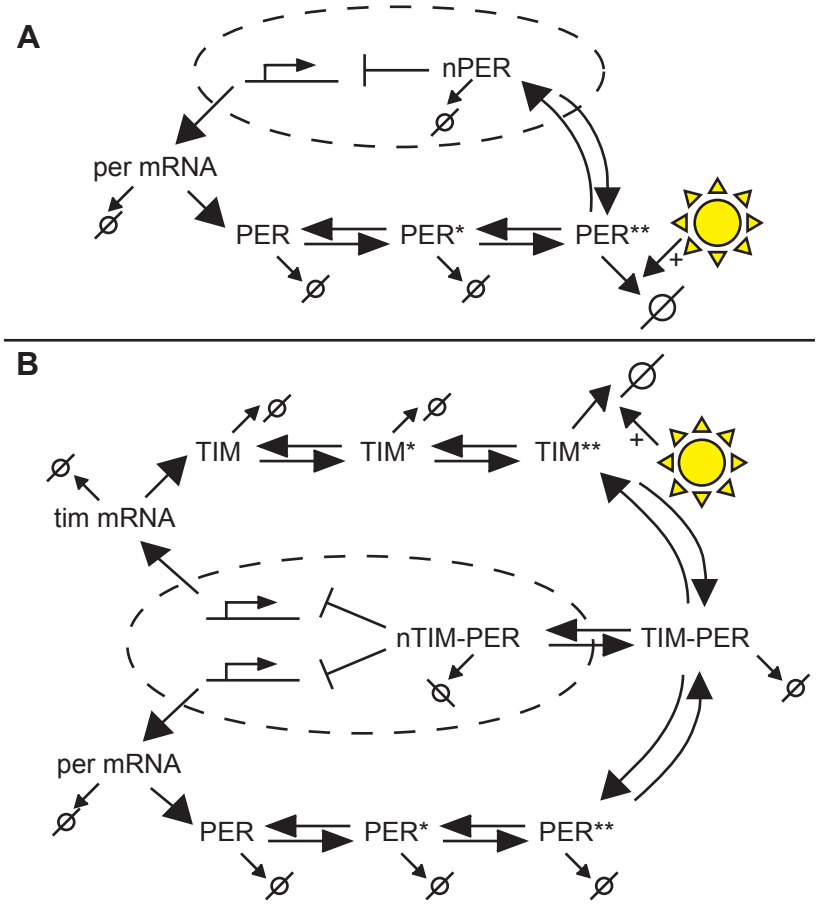

Figure 4.8: The one-loop (a) and the two-loop (b) models of the *Drosophila* circadian clock used for this study, adapted from [104] and, respectively, [105]. PER (and TIM) have two successive phosphorylation steps (indicated as *). PER** or the TIM-PER complex enters the nucleus (dashed circle) and inhibits gene expression (nPER and nTIM-PER stands for nuclear PER or nuclear complex). The light periodically enhances the degradation of PER** or, respectively, TIM**.

4.2.2 Higher Global Robustness of the Two-Loop Model

To sample the parameter space for both models, we use the PCA sampling method. The criteria chosen for a parameter set to be viable are consistent with the experimental findings on the *Drosophila* clock. More specifically, we select parameter sets for which the models, under entrainment (light acts on the system from hour 0 to 12, i.e. ZT0 to ZT12), show the following criteria [173]:

- the oscillations are stable with a period of 24 hours (with a small margin of 0.05h to account for numerical errors) with an entrainment of 24 hours period;
- the relative amplitudes $\left(\frac{\max-\min}{\max}\right)$ of *per* mRNA and the nuclear complex (nPER or nTIM-PER) concentrations are above 60%;
- the *per* mRNA concentration peaks during the early night (between ZT12 and ZT19 hours);
- the nuclear component concentration peaks in the late night (between ZT18 and ZT3 hours) and is with a delay of 4.5 to 10 hours after the peak of *per* mRNA;
- the *tim* mRNA and *per* mRNA concentrations peak within less than two hours difference (for two-loop models).

The sampling procedure is performed over a large range of four orders of magnitude around the original parameter set. With this broad sampling, we suggest that the computed characteristics are inherent in the structure and not in the parameterization of the models. The boundaries are a necessity: as some reactions are bidirectional, the effective rate at equilibrium is the ratio of both forward and backwards reactions leaving individual parameters unconstrained.

The results of the global analysis show that the model with two loops is more resilient to parameter perturbations: the region of the parameter space where the model is properly entrained by an external signal is larger. The model with one loop has on average around two and a half orders of magnitude (2.43 ± 0.02) of possible variations per parameter to fulfill the defined criteria. The symmetric and asymmetric two-loop models have respectively 7.4% and 14.0% more freedom (normalized volume of 2.61 ± 0.03 and 2.77 ± 0.01) reflecting a higher global parameter robustness. The error is the standard deviation of three different simulations.

We further refine our analysis by checking how the oscillations behave when the entrainment is released. This is performed in two stages. First we check if the system is able to maintain proper oscillations for three cycles in the dark (without entrainment). We use the same criteria as above but with a range 10% larger for the phase of the peaks and a range of [21.6h, 26.4h] for the period interval. The hierarchy of results for the three models remains the same, but interestingly the one-loop model shows a 5.3% decrease (to a value of 2.30 ± 0.02) whereas both two-loop models have only 2.8% and 2.6%

less freedom for the symmetric and, respectively, asymmetric models (values of 2.53 ± 0.05 and 2.70 ± 0.01).

We pursue the refinement by selecting the parameter sets that have a proper free-run behavior as observed experimentally. For this stage the absolute phase criteria are not considered and only the ones for the period (10% around 24 hours), the relative amplitude of some components (more than 60%) and the relative phases (between peaks of mRNA and nuclear component concentrations) remain. For this last analysis, the difference is even stronger with the average parameter variation dropping to 2.12 ± 0.03 for the one-loop model (-12.0% compared to the case with entrainment). On the contrary, the symmetric and asymmetric two-loop models show both a decrease of 5.1% to a value of 2.46 ± 0.05 and 2.63 ± 0.02, respectively.

Breaking the symmetry of the parameters of the two loops increases only slightly the viable parameter space. The little difference could be explained by the low constraints on the phosphorylation steps. In the asymmetric model, there are two times more parameters for phosphorylation (for the PER and the TIM loops) and being the least constrained parameters, they increase the average viable volume, when changing independently. More significant is the difference in the decrease of the normalized viable volume due to the refinement (relaxation and free-run operating conditions). This reduction is two times larger for the one-loop model than for the model with two loops. As both two-loop models have the same decrease we can argue that the advantage of the two-loop models in terms of robustness for an entrained system comes from the structure and not from an artifact due to the number of parameters.

4.2.3 Local Analysis Based on the PRC

The local analysis is based on the phase response curve. The PRC measures the positive (or negative) time shift in the phase (usually in hours) that results from an input perturbation given at a specific phase of the cycle. Experimental PRCs of *Drosophila* circadian clock (a diurnal organism) exhibit delay phase shifts for light pulses in the early subjective night, advanced phase shifts in the late subjective night and little phase shifts during the subjective day, a region called the dead zone [174] (figure 4.9). This specific PRC profile allows the system to be entrained at the correct phase with a periodic light signal. We use the PRC to define qualitative measures of the entrainment and use it as a discriminative criterion for model selection. More specifically, we focus on the following aspects: positions of the extrema and occurrence of a dead zone.

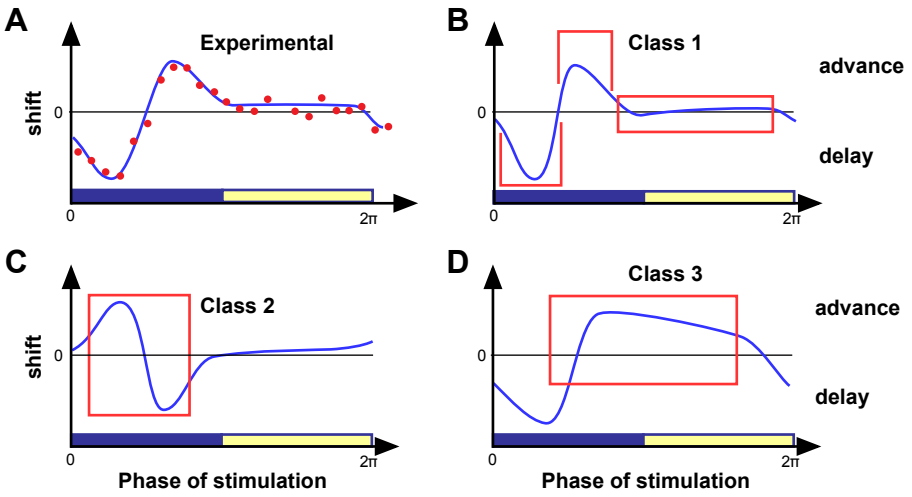

Figure 4.9: Qualitative classification of PRCs. (A) Experimental data (red dots) and interpolated PRC curve (blue line) for the *Drosophila* (adapted from Hall and Rosbach [174]). (B) PRCs of class 1 exhibits a minimum followed by a maximum and the dead zone. (C) PRCs of class 2 see their exterma inverted. (D) PRCs having no dead zone are classified in the third group.

First we sample again the parameter space for sets that fulfill the free-run criteria. The measure of the viable volume shows an even stronger advantage for the two-loop models: the symmetric and asymmetric two-loop models have a normalized viable volume of 2.57 ± 0.03 and, respectively, 2.72 ± 0.01 whereas the value of the one-loop model is of 2.24 ± 0.04 only.

According to the analysis of the PRC for the different viable points in the parameter space, we define three classes of PRCs (Fig. 4.9) based on the position of the maximum and minimum and the existence of a dead zone (at least 9.6 circadian hours at a value inside a range 10 times smaller than $[\min(PRC), \max(PRC)]$). In the first class, parameter sets exhibit PRCs with a minimum followed by a maximum and then a dead zone (in accordance with experimental PRCs), the second class has PRCs with an inversion in the order of the minimum and maximum and, finally, the parameter sets of the third class exhibit PRCs without any dead zone.

With this classification the normalized volume of parameter sets with PRCs of class 1 is reduced by 15.3% (value of 1.90 ± 0.06) in the case of the one-loop model. On the contrary, for the two-loop models, the volume where the parameter sets show a PRC comparable to the experimental ones remains higher: the viable volume decreases by 9.5% for the symmetric two-loop model and by 9.2% for the asymmetric one (values of 2.33 ± 0.06 and, respectively,

2.47 ± 0.04). This local analysis emphasizes the advantage of an additional feedback loop for the entrainment properties: a higher region in the parameter space is consistent with the experimental PRC.

A closer look at the distribution of the three classes along the different parameters shows a clear bias toward low values of the parameters controlling the concentration of the nuclear component. More specifically, for all three models, the degradation rate of the component in the nucleus (nPER or nTIM-PER) is significantly lower for parameters in class 1 (Wilcoxon's rank sum test with $p < 0.05$, see Fig. 4.10). Except for the asymmetric two-loop model, the translocation rates (in or out of the nucleus) are also significantly lower in class 1 (p-value < 0.05). These differences are stronger for the one-loop model. Other parameters show no significant difference with respect to the three classes.

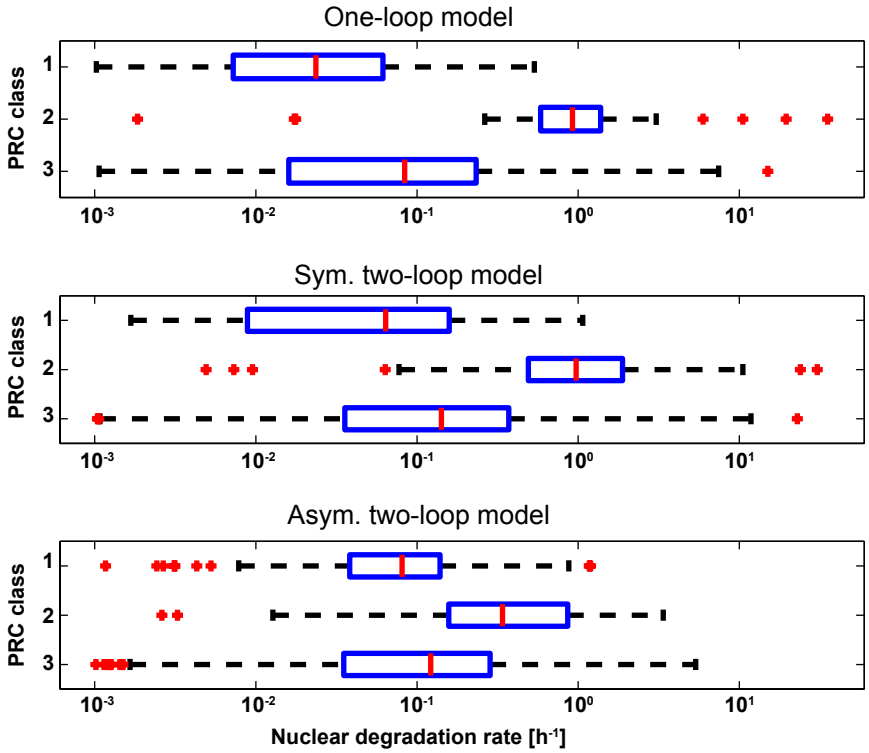

Figure 4.10: Boxplot of the nuclear degradation rate for the three classes of PRCs for the viable parameter sets in the different models. In all models, the average of the distribution for the class 1 is significantly lower than other classes however this difference is more important for the one-loop model.

4.2.4 Conclusion

Taking into account entrainment for the analysis of robustness is a logical step when studying the circadian cycle. Our work suggests that the additional loop enhances the robustness of the entrained system. First, the global analysis gives two advantages of an additional feedback loop: a larger region of the parameter space shows viable results and a larger fraction of it maintains regular oscillations after the entrainment is stopped. This is a critical property for a circadian clock: evolutionary selection pressure acts on this particular phenotype. We can also see that the asymmetry introduced between the two loops does not change the results through the refinement: the robustness comes from the architecture (two-loops) and not the details of the equations.

Second, with the classification based on the PRC, we can investigate what are the critical parameters that influence the entrainment properties. Our global sampling helps to understand the interactions between specific rates and PRC profiles that are consistent with experimental data. The results show that the rate of the reactions controlling the concentration of the inhibitory component (nPER or nTIM-PER) in the nucleus (translocation and degradation) can discriminate between the different PRC classes.

Interestingly, this part of the models, that can be considered as the most sensitive one, is strongly simplified in comparison to the most recent models [171, 107]. In fact, the inhibition by the TIM-PER complex is mediated through another complex, Cycle-Clock, which is also controlled by positive and negative feedbacks. An interesting perspective would be to study how these additional feedback loops influence the robustness and the entrainment properties and if they can overcome the weaknesses found by our research on the simpler models.

4.3 Evolution of Feedback Loops in Oscillatory Systems

Building on the results of the connected viable space for the two models of the cyanobacterial clock, and following the work with models including multiple feedback loops, the logical question that I want to address is: could different architectures be reached with small steps in the parameter space while maintaining viability of the systemic properties? Even though feedback loops are considered key motifs of biochemical systems such as signaling pathways, homeostatic regulatory circuits and oscillators [2], it is not clear from an evolutionary point of view, how multiple loops evolved in complex systems like the early embryonic cell cycle [114] or the circadian clocks [75]. *In silico* approaches have been used to understand how a system with a transcriptional repression pattern can evolve oscillations [175, 176], and if such systems can show oscillations by tuning their kinetic parameters [175, 176, 177]. On the scale of the network topology (neglecting kinetic parameters), oscillations can be conserved while modifying the structure of a network [151].

All these studies do not address the question whether new feedback loops can be created in an oscillatory system without disrupting existing oscillatory behavior. Even if evolution occurs in discrete steps, it seems *a priori* very unlikely that a new loop can be created at random with precisely the correct parameters to maintain the existing period and sufficiently high amplitude of oscillation. The emergence of a new feedback loop would more probably occur in multiple small steps which facilitate adjustment of kinetic parameters to maintain core oscillatory properties. In this work, I aim to answer this precise question: *"could a system evolve from a simple model to a more complex one with a continuous transition in the parameter space?"* I will show that evolution of this kind is possible for simple models that have been used to model the mitotic cell cycle [114]. I propose two models based on early models from the mitotic cycle in embryogenesis [72, 73]. These models have one feedback loop, either positive or negative. I evolve these models toward a system with both positive and negative feedback loops while conserving their systemic properties.

4.3.1 Random Walk for Evolution of Models

I first describe an algorithm to study the emergence of new regulatory motifs in a model. Starting from a model, say $\mathbf{M_1}$, with a nominal parameter vector $\mathbf{k}^{(\mathbf{M_1})}$, the goal is to evolve through small changes of the parameters toward a predefined second model, say $\mathbf{M_2}$ that is similar to $\mathbf{M_1}$, but has

an additional motif. During evolution, some viable properties of the system have to be conserved. For example, in a homeostatic mechanism, the target concentration of a molecule may have to stay in a specific interval; in this case evolution of a new motif could allow faster or more reactive control [51]. For oscillatory systems, the period and the amplitude have to be conserved while better robustness can appear through new motifs [102]. In practice, the parameters of the new motif in model $\mathbf{M_2}$ are set to zero in order to mimic the nominal parameter vector of $\mathbf{M_1}$. The aim of the evolution process is to change progressively these parameters, and if necessary other parameters of the model, in order to reach their nominal values in model $\mathbf{M_2}$. I allow a maximum of 10% variation in the parameter space at each step to mimic the fact that most mutations may affect biochemical parameters only to a small extent.

The algorithm starts with a random walk in the logarithmic domain [178] containing a drift term:

$$\log_{10}(\mathbf{k}^{(j+1)}) = \log_{10}(\mathbf{k}^{(j)}) + \alpha\boldsymbol{\epsilon} + \beta\frac{\boldsymbol{\Delta}^{(j)}}{\|\boldsymbol{\Delta}^{(j)}\|} \tag{4.4}$$

where the drift $\boldsymbol{\Delta}^{(j)} = \log_{10}(\mathbf{k}^{(\mathbf{M_2})}) - \log_{10}(\mathbf{k}^{(j)})$ is the difference vector between the j-th parameter vector and the nominal vector for model $\mathbf{M_2}$ in the logarithmic domain. The random vector $\boldsymbol{\epsilon}$ is normally distributed with independent components. The value $\alpha = 0.041$ is chosen such that the standard deviation of parameter variation is 10% of the previous parameter value; we set β to a value of 1/3.

This random walk finds viable points in the parameter space, i.e. parameter vectors for which a model shows predefined viable properties. When the random walk has reached the vicinity of the nominal vector for $\mathbf{M_2}$, the second stage of the algorithm is to shorten the path from $\mathbf{M_1}$ to $\mathbf{M_2}$ by reducing the number of points by linear interpolation between distant points along the path. As I *a priori* choose the maximal length for the line segments, I do not recover the shortest possible path. During this process, I check that the viable properties are conserved along the line connecting the two intermediate points. If the whole path consists of viable points and the connections between them are also viable, the properties are considered to be conserved along the evolution process. I apply this algorithm to a generic model of the mitotic cycle.

4.3.2 Generic Models of the Mitotic Cycle and Constraints

The mitotic cycle has been one of the first biological systems to be modeled with feedback loops. The first two published models were based either on a

positive feedback [179] or a negative one [180]. Further models were published including both kinds of loops (see [73, 114] and ref. therein). The models for my case study are inspired by models in these papers, because they are simple yet biologically realistic. Specifically, I propose three models with different feedback architectures, shown in figure 4.11. All my models are based on the expression of a protein R, its phosphorylation and its degradation. For the model with positive feedback ($\mathbf{M_P}$, figure 4.11B), the phosphorylated state of the protein (R_P) acts as a kinase for a secondary protein Z ($Z \rightleftarrows Z_P$). The positive feedback loop is formed by Z_P that increases the rate of the reaction $R \rightarrow R_P$. For the model with negative feedback ($\mathbf{M_N}$, figure 4.11C), an intermediate step is needed to introduce more delay: R_P acts as a kinase for an intermediate protein X ($X \rightleftarrows X_P$) and X_P phosphorylates a third protein Y ($Y \rightleftarrows Y_P$). The phosphorylated state of this protein, Y_P, increases the degradation rate of R, therefore acting as a negative feedback. For the more complex model ($\mathbf{M_{PN}}$, figure 4.11A), both loops are present: R_P influences the phosphorylation of X and Z. To translate the models into differential equations, we assume that the reactions involving R and R_P follow mass-action kinetics:

$$\begin{aligned} \frac{\mathrm{d}[R]}{\mathrm{d}t} &= k_1 - p([Z_P])[R] \\ \frac{\mathrm{d}[R_P]}{\mathrm{d}t} &= p([Z_P])[R] - n([Y_P])[R_P] \end{aligned} \tag{4.5}$$

The functions $p([Z_P])$ and $n([Y_P])$ reflect the positive and negative feedbacks contained in model $\mathbf{M_{PN}}$:

$$\begin{aligned} p([Z_P]) &= k_2 + k_{11}[Z_P] \\ n([Y_P]) &= k_3 + k_{12}[Y_P] \end{aligned} \tag{4.6}$$

In models $\mathbf{M_P}$ and $\mathbf{M_N}$, k_{12}, resp. k_{11}, are set to zero such that only one feedback is effective. The phosphorylation of X, Y, and Z are governed by Michaelis-Menten kinetics:

$$\begin{aligned} \frac{\mathrm{d}[X_P]}{\mathrm{d}t} &= k_4[R_P]\frac{([X_T]-[X_P])}{k_{10}+([X_T]-[X_P])} - k_5\frac{[X_P]}{k_{10}+[X_P]} \\ \frac{\mathrm{d}[Y_P]}{\mathrm{d}t} &= k_6[X_P]\frac{([Y_T]-[Y_P])}{k_{10}+([Y_T]-[Y_P])} - k_7\frac{[Y_P]}{k_{10}+[Y_P]} \\ \frac{\mathrm{d}[Z_P]}{\mathrm{d}t} &= k_8[R_P]\frac{([Z_T]-[Z_P])}{k_{10}+([Z_T]-[Z_P])} - k_9\frac{[Z_P]}{k_{10}+[Z_P]} \end{aligned} \tag{4.7}$$

The terms $[X_T]$, $[Y_T]$ and $[Z_T]$ denote the total concentration of proteins X, Y and Z, respectively. We choose them to be equal to 1, and note that the

absolute value is irrelevant, because a proper scaling of the parameters allows changing time and concentration independently. In order to simplify the notation, we will not write the units for time, concentration and parameters.

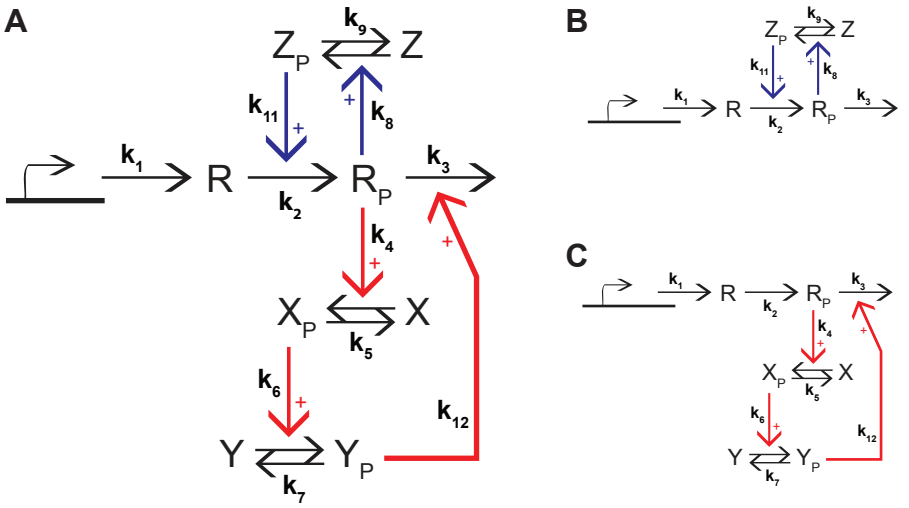

Figure 4.11: Reaction diagrams of the three models based on models of the mitotic cycle [179, 180, 114]. (A) In the complete model, $\mathbf{M_{PN}}$, the protein R is produced at a constant rate k_1 and phosphorylated at a rate k_2 forming R_p. The phosphorylated protein Z_P modulates this phosphorylation rate by means of a positive feedback loop (blue diagram in the figure). In addition, R_P is degraded with a rate k_3 that depends on the phosphorylated protein Y_P by means of a negative feedback loop (red diagram in the figure). (B) $\mathbf{M_P}$ model with only the positive feedback loop. (C) $\mathbf{M_N}$ model with only the negative feedback loop.

With the above-defined random walk in the 12-dimensional parameter space of my models, I can identify new viable parameter vectors. The chosen viable properties of my models are related to the oscillations of the concentration of R_P. In particular, I want the values of the period, the peak value and the amplitude of these oscillations to remain within predetermined intervals. A parameter vector is considered viable if the concentration of R_P oscillates with a period in the (arbitrary) interval [0.9, 1.1], a peak value contained in the range [0.5, 1.0] and an amplitude that is at least 40% of the peak value. I chose these criteria to reflect an important feature of biological oscillations, namely that they avoid very small amplitudes. Period and the peak values are readily adjusted with a proper scaling of the parameters. For models $\mathbf{M_P}$ and $\mathbf{M_N}$, the nominal parameter vectors were taken from [73] and rescaled in order to have a period of 1 and a peak value of 0.66. For the model $\mathbf{M_{PN}}$, the nominal parameters are chosen to be of the same order of magnitude as for

models $\mathbf{M_P}$ and $\mathbf{M_N}$ (in particular they were obtained as the average of both vectors and then rescaled to fulfill the same viable properties).

4.3.3 Parameters Adjust Through Evolution

For most of the simulations ($> 85\%$) my algorithm is able to connect one of the one-loop models to the two-loop model (refer to Figure 4.12A-B). Usually about a few thousands of points are tested during the random walk, around half of which are viable. The path is then reduced to about one hundred points connected with viable segments.

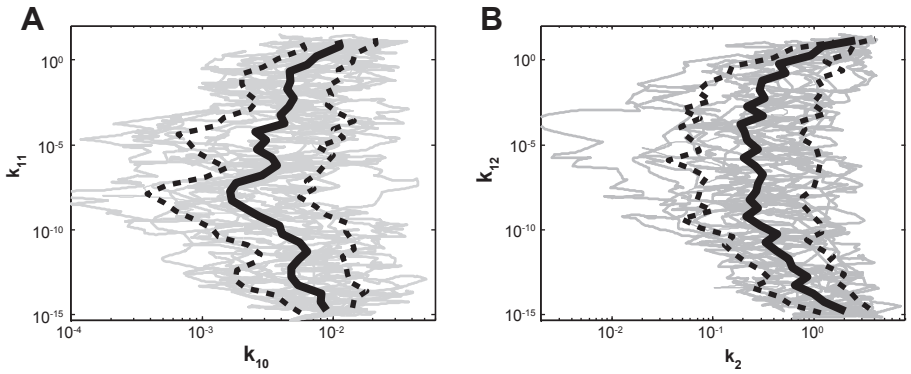

Figure 4.12: Paths for the model evolutions. (A) 15 different paths from model $\mathbf{M_N}$ to model $\mathbf{M_{PN}}$ projected on the plane (k_{10}, k_{11}). Parameter k_{11} is increased during evolution strengthening the positive feedback. (B) 15 different paths from model $\mathbf{M_P}$ to model $\mathbf{M_{PN}}$ projected on the plane (k_2, k_{12}). Parameter k_{12} is increased during evolution strengthening the negative feedback. In both plots, light gray curves correspond to paths, the black line is the average path, and dashed black lines are the standard deviations along the average path.

The addition of a positive feedback loop to the model with only a negative feedback loop is also possible, but the straight line connecting the two parameter vectors for model $\mathbf{M_N}$ and $\mathbf{M_{PN}}$ is not viable. The line crosses a Hopf-bifurcation, where the amplitude decreases and then oscillations disappear. Therefore to connect both models, the random walks follow a bent trajectory. The most significant adaptation is seen for the Michaelis constant, k_{10} (Fig. 4.12A). When k_{10} decreases, the transitions ($X \rightleftarrows X_P$) and ($Y \rightleftarrows Y_P$) become switch-like which strengthens the nonlinearity and may support oscillatory behavior.

The addition of a new negative feedback loop to the model with only one positive feedback loop is also possible. In this case also, the straight line in

the parameter space is not viable because the period increases beyond the allowed maximal value along the path. The random walks become biased toward smaller values of the phosphorylation rate of R, k_2 (Fig. 4.12B). This can be explained by the fact that k_2 has a positive impact on the period. Decreasing it also decreases the period.

4.3.4 Conclusion

This work is a first step to understand the emergence of feedback loops in oscillatory systems. Using a Monte Carlo approach, I have shown that a simple model can evolve toward a more complex one with small adaptations of its kinetic parameters. I found evolutionary paths in the parameter space that conserve the viable properties of the system. In the considered oscillatory models, such conservation means that period and amplitude of oscillation have to stay within predetermined intervals. For my two case studies, the addition of a second loop is possible only if multiple parameters are changed simultaneously during this process.

In large systems, the high-dimensionality prohibits classical qualitative bifurcation analyzes. Moreover, if quantitative constraints on system function have to be fulfilled, the problem cannot be solved analytically. A random sampling approach based on brute force is limited by the computational cost, which is very high for large ranges of parameters in high dimensions. In such cases, the random walk approach scales better and is a promising tool to understand the evolution of more complex systems, such as the mammalian circadian clock.

4.4 Global Analysis of the Generic Mitotic Cycle Model

The results of the previous section showed that the generic model for the mitotic cycle can oscillate with parameter values in a broad range. In particular, feedback strength of one or the other loop can be decreased to zero while oscillations remain. Yet, as both feedback cannot *a priori* be null at the same time, the viable parameter space should be strongly non-convex. It is therefore a ideal case study for the two-stage sampling method (section 3.1.2). In collaboration with E. Zamora, we characterized the non-convex viable space of this model. In particular, we are interested in its geometry in relation with the oscillator's robustness and the connectivity of the system as described in the section 4.1.4.

4.4.1 Model and Constraints

Our system has 12 parameters (see equations (4.5) to (4.7)). As usual, we worked in the logarithmic domain to explore broad ranges of parameters, but we still constrain the individual parameters as follows

$$\begin{aligned} k_i &\in \left[10^{-4}, 10^{2}\right], \ i = 1, 2, \ldots, 10, \\ k_i &\in \left[10^{-7}, 10^{2}\right], \ i = 11, 12. \end{aligned} \tag{4.8}$$

Together, these ranges define the 12-dimensional parameter subspace $\mathcal{K}$ used for the sampling.

For the Adaptive Metropolis sampling, we use the cost function

$$E(\mathbf{k}) = \begin{cases} \left[(\pi_T(\mathbf{k}) - 1)/0.1\right]^2, & \text{if } R_P \text{ oscillates,} \\ \infty, & \text{otherwise,} \end{cases} \tag{4.9}$$

where $\pi_T(\mathbf{k})$ is the period of the oscillations of R_P for a parameter vector $\mathbf{k}$. The minimum of this cost function is attained by parameter vectors for which $\pi_T(\mathbf{k}) = 1$.

Finally, we use the viability condition $E(\mathbf{k}) \leq 1$, meaning that a parameter vector $\mathbf{k}$ is viable if it makes R_P oscillate with a period in the interval $[0.9, 1.1]$ as in the previous section. Note that in this work, we have no explicit restriction on the amplitude or peak values, yet we consider that the oscillation should have a least a relative amplitude ($\frac{max-min}{max}$) of 10^{-5}.

4.4.2 Sampling of the Non-Convex Viable Region

First, we carried out the Adaptive Metropolis sampling that roughly sampled the viable space. The Ellipsoid-Based sampling completed the work and yielded a large amount of points in the viable region of the parameter space. In a next step, we performed a Monte Carlo integration and obtain 3595 parameter vectors uniformly distributed in the viable region. The detailed results and the performance of the algorithm was assessed in [135]: our two-stage sampling method carries out a 13 times more accurate estimation of the viable volume, and obtains 400 times more uniformly distributed viable points than a brute-force approach using the same number of sampling points randomly distributed in $\mathcal{K}$.

Now I will focus on the relations between the shape of the parameter region and the oscillatory function. Specifically, we noted that the viable region in the (k_{11}, k_{12}) plane (Figure 4.13A) is composed of two approximately rectangular or bar-like regions that, together, form a non-convex shape resembling an inverted L. Parts of these regions define topologies in which the effect of one of the feedback loops is very weak. More precisely, the left part of the horizontal bar corresponds to viable parameter points for which k_{11} is small and k_{12} is large. In this region, the effect of the positive feedback loop is weak. Conversely, the bottom part of the vertical bar consists of viable parameter points for which k_{11} is high and k_{12} low. It corresponds to architectures where the positive feedback loop is dominant. These regions cover the evolutionary paths that we found in the previous section. If we project the viable parameter vectors in the other dimensions, we noticed that some parameters are hardly constrained. In particular, when k_{11} is low, i.e. the positive feedback loop inactive, the parameters k_8 and k_9 of the feedback loop are spread over the whole admissible region (Figure 4.13C). The reverse is also true with k_4 to k_7 being unconstrained when k_{12} is very low (Figure 4.13D). These observations confirm that the sampling algorithm has properly covered the viable region, even if it is non-convex.

4.4.3 Classification Based on the Feedback Loop Importance

This specific geometry clearly leads to a classification of the viable vectors depending if the negative or the positive feedbacks are predominant. For every parameter point, we asked whether a feedback loop was essential by removing the positive (negative) loop. To do so, we set all the parameters involved in

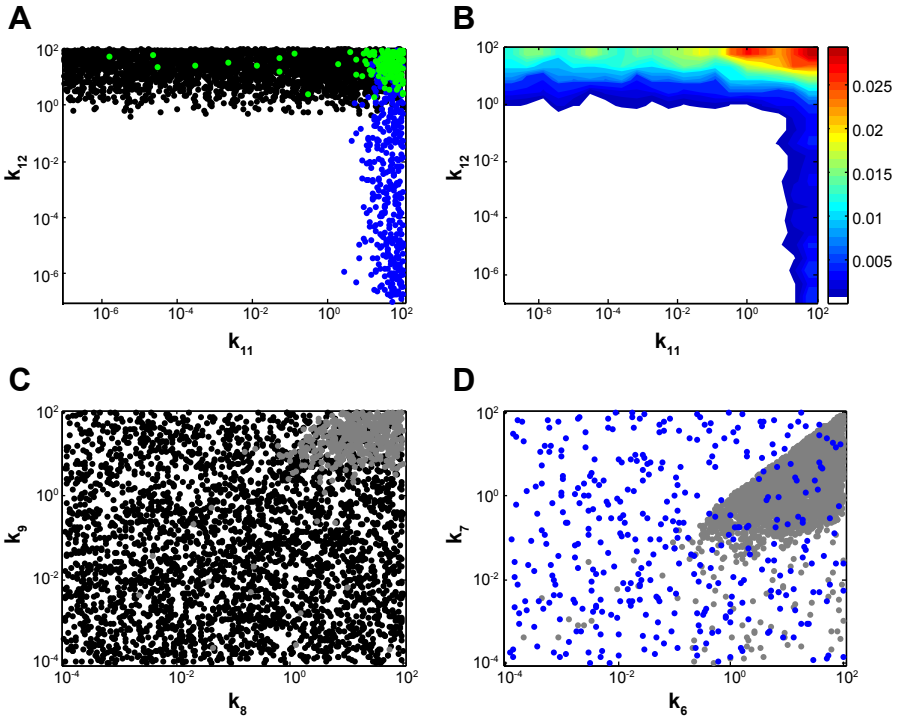

Figure 4.13: Viable parameter vectors for the generic model of the mitotic cycle (results of the MC integration). (A) Projection in the (k_{11}, k_{12}) plane show that the viable region is non-convex because at least one of the feedback should be effective. (B) Density plot of the viable vector projected in the (k_{11}, k_{12}) plane shows that the density of the viable parameter vectors for which the negative feedback is effective (top region) is higher with a peak when both feedback (top right corner) are effective. Warm colors represent a higher density. (C) Projection in the plane (k_8, k_9) of the points with essential negative feedback (in red; black points are other viable vectors) shows that the parameters of the positive loop are not constrained when the feedback is not essential. (D) Projection in the plane (k_6, k_7) of the points with essential positive feedback (in red; black points are other viable vectors) show the same phenomenon of unconstrained parameter values for the negative loop.

the positive (negative) loop equal to zero and determined whether the system lost viability. When the system lost its viability, we checked if, for this specific parameter point, the role of the positive (negative) loop was just to increase the activation (degradation) rate of R_P without being involved in the creation of sustained oscillations. To do so, we removed again the positive (negative) loop and increased the value of k_2 (k_3) which controls the activation (degradation) rate. It is a way to assess if the effect of one of the feedback loop is essential or marginal for oscillations. To summarize, we classified each of the viable vectors we found into one of the following classes:

- Essential negative feedback loop: The oscillator remains viable after removing the positive loop, and eventually increasing the activation rate of R_P.
- Essential positive feedback loop: The oscillator remains viable after removing the negative loop, and eventually increasing the degradation rate of R_P.
- Essential positive and negative feedback loops: No loop can be removed or compensated by a higher activation or degradation rates without destroying the oscillations.

We found that oscillator architectures for which the negative feedback loop is essential occupy the vast majority (86%) of the viable space we sampled (see figure 4.15A). In contrast, significantly fewer parameter vectors lead to viable oscillations based on an essential positive loop (10%), or on a combination of essential positive and negative feedback loops (4%). Unsurprisingly, both feedback loops cannot be removed at the same time.

With this classification, we can quantify more rigorously the parameter space with respect to the importance of the different feedback loops. As already observed, if a single loop is essential, only the parameters mainly responsible for this loop will be constrained (see figure 4.13C-D). These are parameters k_8, k_9, k_{11} for the positive loop, and parameters k_4, k_5, k_6, k_7, k_{12} for the negative loop (Figure 4.11). To illustrate these constraints, we used a boxplot similar to figure 4.10, in figure 4.14A-B, black coloring indicates to what extent parameters involved in the negative loop are constrained if this loop is essential, blue coloring indicates these constraints if only the positive loop is essential, and green coloring indicates these constraints if both loops are essential. Clearly, parameters involved in the positive loop can vary to a lesser extent if this loop is essential than when it is not essential (Fig. 4.14A). Analogous observations can be made for parameters involved in the negative loop (Fig. 4.14B).

A comparison of Fig. 4.14A-B also shows that parameters involved in the negative and positive feedback loop are constrained to different extents. Specifically, negative loop parameters can vary over broader intervals when the negative loop is essential than positive loop parameters can when the positive loop is essential. Specifically, the five negative loop parameters can vary in an interval of two orders of magnitude when the negative loop is essential, whereas the three positive loop parameters vary in an interval of just one order of magnitude when the positive loop is essential. In addition, the parameters that are

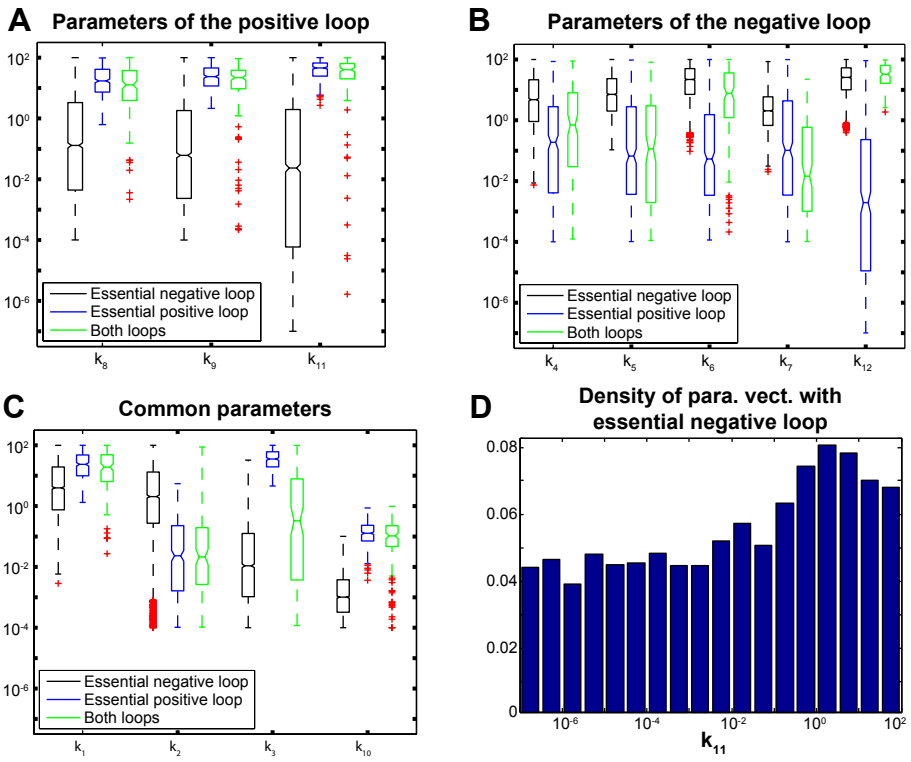

Figure 4.14: Distribution of the viable parameter vectors. (A-C) Boxplot of the different parameters for viable vectors in the three classes: systems with an essential negative loop (in black), with an essential positive loop (in blue), and where both loops are essential (in green). (A) Boxplot for the parameters involved in the positive feedback loops. (B) Boxplot for the parameters involved in the negative feedback loops. (C) Boxplot for the parameters forming the core of the system. (D) Distribution of the viable vectors for systems with an essential negative loop along the parameter k_{11}.

not part of any loop (k_1, k_2, k_3, k_{10}) are again more constrained in oscillators with essential positive feedback loop than in oscillators with a negative feedback loop (Fig. 4.14C). These observations are also confirmed by the principal component analysis of the parameter vectors of the different classes.

Taken together, these observations imply that oscillator architectures based on a negative loop fill more of the viable space, and allow individual parameters to vary more broadly than architectures based on positive feedback loops. In other words, in our generic model, the oscillator based on an essential negative feedback loop is more robust to parameter variations than oscillators with essential positive loops or with both essential positive and negative loops. We should note that this statement takes into account only the oscillatory period and other properties such as amplitude may depend on the architecture as

observed by Tsai *et al.* [117].

In addition, we found that adding a positive (not necessarily essential) loop to oscillators based on a negative feedback loop further increases robustness and the allowable range of parameter variation. Figure 4.13B already showed a first hint of this observation: the highest density of viable parameter points occurs in regions of parameter space where both k_{11} and k_{12} are high therefore both feedback loops are active. With the classification, we can specifically observe the architectures where only the negative feedback loop is essential. For this class the mean value of the parameter k_{11} which controls the strength of the positive feedback loop, is significantly higher (p-value= $4.5 \cdot 10^{-18}$ t-test) than the center of the interval in which k_{11} is defined. In other words, randomly sampled architectures with an essential negative feedback loop preferentially occur in regions of parameter space where a positive loop is also active. The value of the parameter k_{11} is also positively correlated (Pearson's $r = 0.88$) with the density of viable parameter points (Fig. 4.14D). Thus, a higher strength of the positive feedback loop increases the number of kinetic parameters combinations that gives rise to viable oscillatory behavior. Taken together, these observations suggest that an added positive (but not necessarily essential) feedback loop gives a negative-loop-based oscillator access to a larger set of viable parameter vectors.

4.4.4 Connectivity of the Viable Parameter Space

We next turned to the connectivity of the oscillator's viable space. We have already shown (section 4.3) that a positive (or negative) feedback-based oscillator can evolve to an oscillator with both feedback loops being active. With the dense sampling obtained in this work, we can now apply to this system the same approach as in section 4.1.4. To study connectivity, we first randomly chose 333 parameter points of all the three classes we defined. In this set of points, all three architectures with essential positive feedback, essential negative feedback, or both essential, are represented. Two nodes are connected by an edge if the entire straight line between the nodes does not leave the viable space, as indicated by a numerical interpolation procedure (Eq. (4.3)). Such an edge reflects the existence of a straight evolutionary path from one to the other node (parameter vector) that does not leave the viable space. Based on this information, we defined a graph whose nodes are the viable parameter points,

and whose edges are given by the adjacency matrix [181]

$$A_{i,j} = \begin{cases} 1, & \text{if } i \text{ and } j \text{ are viable points connected by a viable straight line,} \\ 0, & \text{otherwise,} \end{cases} \tag{4.10}$$

where $i, j = 1, 2, \ldots, 333$.

We found that the graph thus defined has one giant connected component that comprises 95% of all nodes, similarly as the cyanobacterial clock models (section 4.1.4). The connected component contains nodes associated with all three basic architectures, but these three kinds of nodes are not equally likely to be connected to each other (Fig. 4.15B). Specifically, we analyzed the graph topology and found that nodes (viable vectors) corresponding to oscillators with essential positive feedback loops are only connected to themselves, and to nodes with essential positive and negative feedback loops. Similarly, nodes that define topologies with essential negative feedback loops are only connected to themselves and to nodes with essential positive and negative feedback loops. This partitioning of the adjacency matrix is well represented in the figure 4.15A. These properties mean that evolutionary trajectories that connect oscillators with an essential positive feedback loop to oscillators with essential negative feedback loop, need to pass through configurations for which both loops are essential. The different topologies can be connected with evolutionary trajectories that maintain systemic properties but the trajectories have to pass through a region of the parameter space where both loops are essential. This observation also confirms previous analyses: most of the viable space forms a non-convex connected body.

4.4.5 Conclusion

Our two-stage sampling method is able to properly cover this non-convex region in a 12-dimension space. Detailed analysis of the viable parameter space for the generic oscillator indicated that the viable region is a non-convex connected body in which three classes of parameter vectors exist. Based on the previous work, we defined three classes of parameter vectors that correspond to oscillators where the negative feedback loop, the positive feedback loop, or both loops are essential for oscillations.

The dense sampling allows us to assess their difference with statistics of their parameter ranges. With this method, we found that oscillators with an essential negative feedback loop provide more robust fixed period oscillations than those based on an essential positive loop, and that the addition of

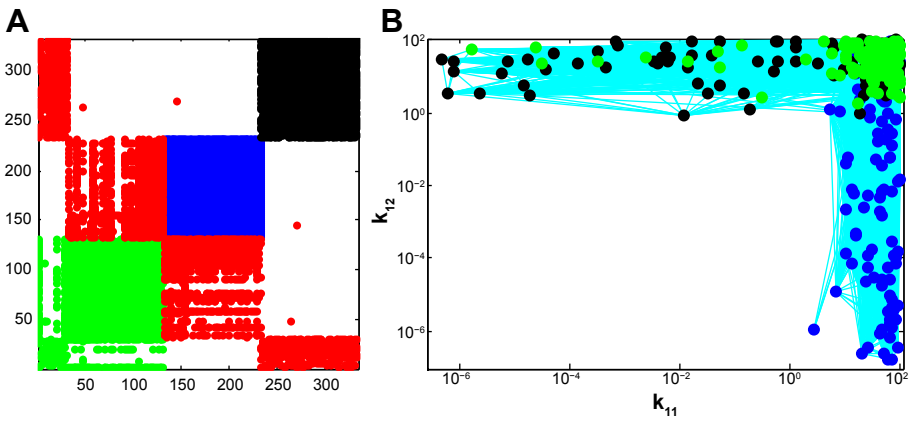

Figure 4.15: Connectivity of the viable parameter space. (A) Representation of the adjacency matrix for the connectivity of the different viable vectors. Indexing of the parameter vectors is such that the different classes based on the essentiality of the feedback loops are grouped together (the first 131 columns are vectors where the two feedback are essential, the next 101 columns are vectors where the positive feedback is essential and finally the last columns are vectors where the negative feedback is essential). Each point represents the existence a direct viable path between the vectors: green ones are for links between the group of vectors with positive and negative essential loops, blue ones for parameter vectors with essential positive feedback, black ones for parameter vectors with essential negative feedback and, finally, red ones are links between vectors of different classes. (B) Viable connections (in light blue) between the viable parameter vectors of the different classes (black dots for the essential negative, blue ones for the essential positive and green ones for both loops being essential) cover the viable space in a non-convex way.

a nonessential positive feedback loop to an oscillator with an essential negative feedback loop increases the robustness of fixed period oscillations (similar results were found in [117]). These results are consistent with the evidence from circadian oscillators: they rely on positive and negative feedback loops [90, 107, 60, 114], even though negative feedback alone is sufficient for fixed period oscillations [104, 180, 105]. Moreover, we showed that biological oscillators evolve toward more complex systems with multiple feedback loops as it is the only way to change topology without disrupting oscillations.

4.5 Robustness to Molecular Noise of Oscillatory Models with the Different Architectures

This work pursues the previous analysis on the generic mitotic oscillation with a local robustness assessment. In collaboration with D. Gonze from Univeristé libre de Bruxelles, we used the same model of the mitotic cycle as in section 4.3 and study its robustness to molecular noise. After an analysis in the phase space to show the effect of an additional positive feedback loop, we use the glocal approach to assess the higher robustness of the dual-feedback model.

Recently, Tsai *et al.* [117], using several prototypical models, performed a series of simulations showing that positive feedbacks lead to a greater tunability of the frequency and to an increase of the region of the parameter space which leads to limit-cycle oscillations. Hasty *et al.* [182] proposed a theoretical model based on interlocked positive and negative feedback loops and showed that such design, when coupled to another genetic oscillator, is capable of entrainment and of amplified oscillations. Recently, guided by the predictions of computational models, Stricker *et al.* [123] designed and constructed experimentally an artificial oscillator based on interlinked positive and negative feedback loops. This study confirmed that the positive feedback loop provides the system with a greater tunability of its frequency and leads to an increase of the robustness of the oscillations in the sense that it functions over a large number of conditions (temperature, IPTG, carbon source, etc).

In the present work, our aim is to check if the positive feedback loop may also lead to a higher robustness of the oscillations with respect to molecular noise. Several works already showed that oscillators based on positive and negative regulatory elements make oscillations more resistant to fluctuations [142, 161], but no comparative study showed how the addition of a positive feedback to an oscillator affects its robustness. We consider here two minimal models for the cell cycle. The first one is only based on a negative feedback similar to the one in figure 4.11C. The second one has the same architecture, but incorporates an additional positive feedback in the negative loop. We performed stochastic simulations using the Gillespie algorithm [15] and we quantify the robustness of the oscillations using the auto-correlation function and the distribution of the periods. We show that the positive feedback loop increases the robustness of the oscillations independently of the parameter values, and we provide a possible explanation for this observation.

4.5.1 Minimal Models for the Mitotic Cycle

We consider here a minimal model proposed by Goldbeter for the frog embryonic cell cycle [180] (Fig. 4.16A). The oscillator involves the activation of a cyclin-dependent kinase (CDK1) by Cyclin B, and the CDK1-induced degradation of Cyclin B by an ubiquitin ligase which is part of the ubiquitin-mediated proteolysis system. Once activated, CDK1 triggers the entry into mitosis.

In an extension of the model, Goldbeter included an additional positive feedback loop, mediated by the CDC25 phosphatase [183]. In this work, the positive feedback was modeled with an additional variable, standing for the active fraction of CDC25. The latter is activated by CDK1 and, once active, CDC25 activates CDK1. Here we simplify this model by considering a direct feedback of CDK1 on itself (Fig. 4.16B). This can be seen as an auto-catalytic process.

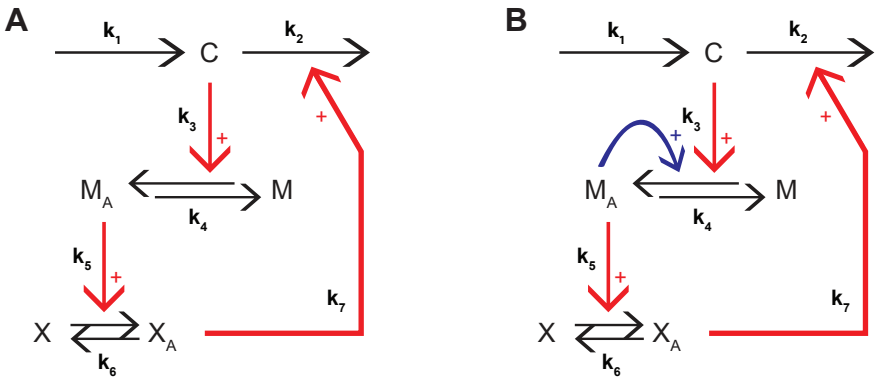

Figure 4.16: Scheme of the models of the mitotic cycle. (A) 3-variable model [180] with the feedback loop in red. (B) 3-variable model including a positive feedback loop (auto-catalysis, in blue), adapted from [183]. Variables C, M_A, and X_A denote the Cyclin B, the active form of CDK1 kinase, and the active cyclin protease, respectively. The variables M and X refer to their inactive form.

The time evolution of the three variables is governed by the following system of kinetic equations for both models [180, 183]:

$$\begin{aligned}
\frac{\mathrm{d}[C]}{\mathrm{d}t} &= k_1 - k_7[X_A]\frac{[C]}{K_d + [C]} - k_2[C] \\
\frac{\mathrm{d}[M_A]}{\mathrm{d}t} &= k_3(a + b[M_A])\frac{[C]}{K_C + [C]}\,\frac{([M_T] - [M_A])}{K_1 + ([M_T] - [M_A])} - k_4\frac{[M_A]}{K_2 + [M_A]} \\
\frac{\mathrm{d}[X_A]}{\mathrm{d}t} &= k_5[M_A]\frac{([X_T] - [X_A])}{K_3 + ([X_T] - [X_A])} - k_6\frac{[X_A]}{K_4 + [X_A]}
\end{aligned} \tag{4.11}$$

In these equations, the variables denote the concentration of Cyclin B ($[C]$), of active CDK1 kinase ($[M_A]$), and of active cyclin protease ($[X_A]$). Note that in the original version [180], M_A and X_A were the fraction of active CDK and protease but, in order to facilitate the conversion to the stochastic version of the model, we write here all the variables in terms of concentration (in nM) and consider that the total amount of M and X are $[M_T] = 1nM$ and $[X_T] = 1nM$. The positive feedback is effective when $b > 0$. In the following we will compare the case where $b = 0$ and $b = 1$. It is interesting to underline that, in this version of the model, the positive feedback can thus be added continuously through a progressive increase of one parameter (b). To take into account the fluctuations arising from the limited number of molecules, we need to resort to stochastic simulations using the Gillespie algorithm [15] (see section 1.2.2). We set $\Omega = 1000$, which lead to a number of molecules of few hundreds, a value in agreement with the estimation of the actual number of cell cycle molecules in a cell [184].

4.5.2 Phase Space Analysis

To understand the differences of dynamics between the models, it is insightful to examine the dynamics in the phase space. The deterministic and stochastic limit cycles associated with the oscillations are given in figure 4.17A-B (red closed curves). To get a deeper understanding of the dynamics, it is useful to draw the nullclines of the system (thin lines). These curves have been obtained by bifurcation analysis of the reduced model defined by the second and third equation in 4.11 with C=constant. The main difference between the two models is the appearance of an S-shaped curve in the reduced model with auto-catalysis. This S-shaped curve is associated with bistability which results in periodical switches between the two plateaus (panel B). Two time scales thus appear: a slow motion when the system moves along the upper and lower branches of steady states and a rapid jump from one steady state to the other. Typical stochastic time series obtained by simulating our models with the Gillespie algorithm are shown in figures 4.17C (for $b = 0$) and 4.17D (for $b = 1$). In presence of noise, the oscillations still persist but their amplitude and period show some variability. We can already notice that the model with auto-catalysis appears more robust than the model without auto-catalysis.

To quantify the effect of noise, we computed the auto-correlation function [41] and the period distribution (Fig. 4.18). Since the entry into mitosis is controlled by the active CDK1, we computed these two functions using variable

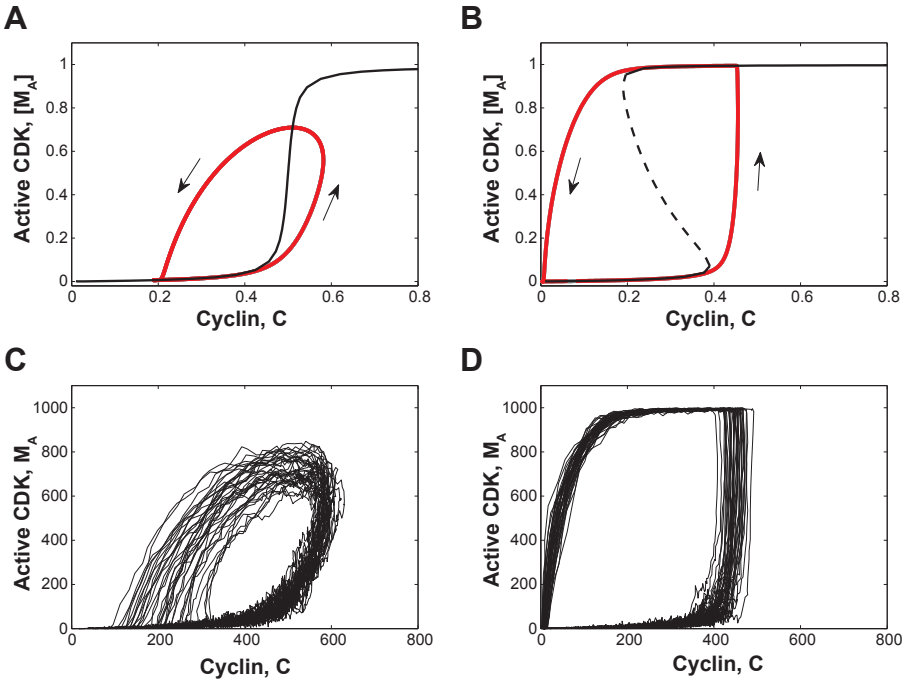

Figure 4.17: Deterministic vs. stochastic limit cycles for both models of the mitotic cycle. (A,C) Model without auto-catalysis ($b = 0$). (B,D) Model with auto-catalysis ($b = 1$). (A,B) The thin line corresponds to the steady state of M_A when C is taken as a parameter. The red curve is the deterministic limit cycle. (C,D) Stochastic trajectories obtained for $\Omega = 1000$. Parameter values are: $k_1 = 0.025$ nM/min, $k_2 = 0.01$ min^{-1}, $k_3 = 3.0$ min^{-1}, $k_4 = 1.5$ min^{-1}, $k_5 = 1.0$ min^{-1}, $k_6 = 0.5$ min^{-1}, $k_7 = 0.25$ nM/min, $K_d = 0.02$ nM, $K_1 = K_2 = K_3 = K_4 = 0.005$ nM, $K_C = 0.5$ nM, $M_T = X_T = 1$ nM, a=1nM. In panels A and B, the concentrations of C and M_A are in nM.

M_A. The periods (or, rather, the peak-to-peak intervals) were determined as the time interval separating two successive upward crossings of the mean level of variable M_A, an arbitrary value which can be seen as the threshold above which mitosis is triggered. We then use the half-life of the decorrelation and the standard deviation of the period as quantifiers of the robustness [142, 41]. Comparing the auto-correlation function and the period distribution, it is now obvious that the model with auto-catalysis ($b = 1$) is more robust than the model without auto-catalysis ($b = 0$). Indeed, for the model without auto-catalysis (Fig. 4.18A vs. Fig. 4.18B), the auto-correlation decreases more rapidly and the variability of the period is greater (Fig. 4.18C vs. Fig. 4.18D), reflecting a higher sensitivity to noise. Note that the two measures used here rather focus on the robustness of the period of the oscillations. We could have quantified the variation of the amplitude of the oscillations, but from a

biological point of view we can hypothesize that mitosis is triggered when a threshold in the concentration of CDK1 is reached and that small variations of the amplitude would not affect the dynamics of cell cycle.

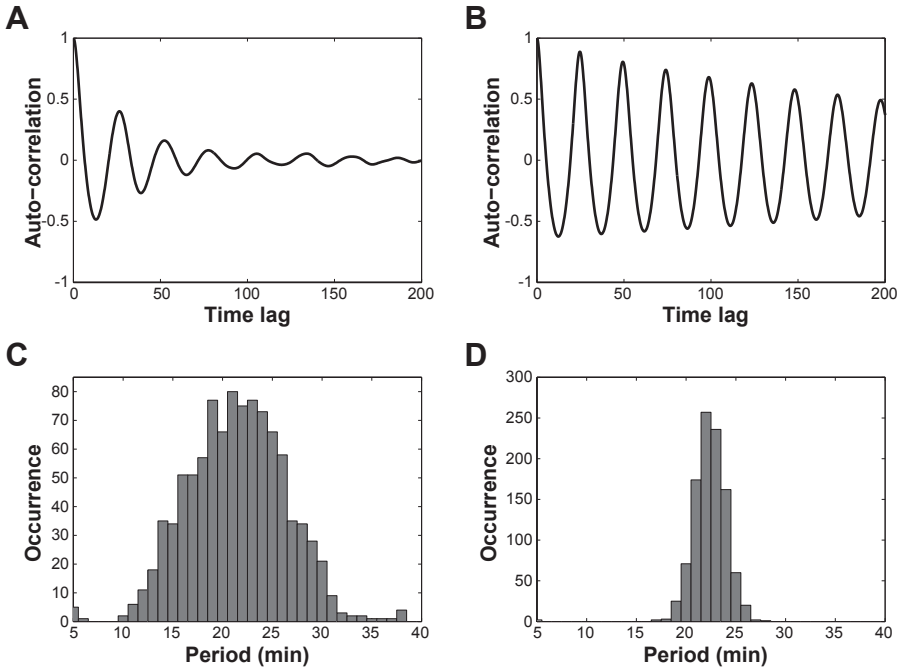

Figure 4.18: Quantification of the robustness of the stochastic oscillations (obtained for $\Omega = 1000$). (A,C) Model without auto-catalysis ($b = 0$). (B,D) Model with auto-catalysis ($b = 1$). (A,B) Auto-correlation function. (C,D) Period distribution. These results have been obtained for the time series of M over a time period of 10000 min. Parameter values are as in Fig. 4.17.

4.5.3 Global Analysis

So far we have compared the two models for one parameter set only. As discussed in the chapter 2, such analyses could be biased and a global approach should be used to check if our observations are parameter-independent and therefore due to the differences in architecture. Thus, with the PCA sampling method, we generated about 100 parameter vectors for each model fulfilling the following systemic properties: a limit-cycle oscillations with a period within the range $[30, 40]$ minutes and a minimum amplitude of 0.6 for $[M_A]$. In order to avoid extreme and irrealistic values of some parameters, the sampling is restricted to a region of four orders of magnitude along each parameter, centered on the published parameter vector (see legend of Fig. 4.17).

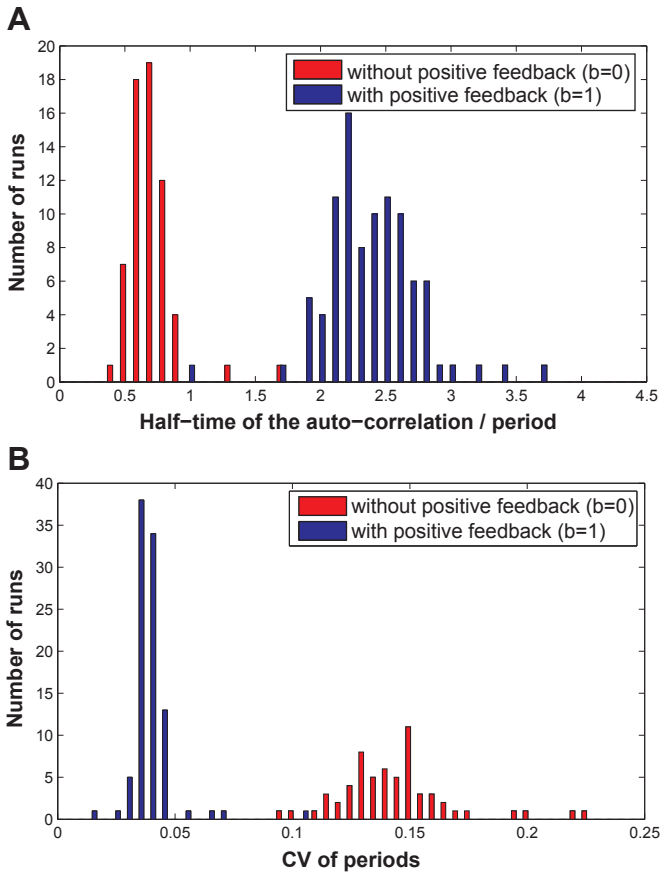

Figure 4.19: Robustness of the stochastic oscillations for various parameter sets. (A) Distribution of the auto-correlation half-time and (B) distribution of the CV (coefficient of variation) values of the period for the model without ($b = 0$) and with ($b = 1$) the positive feedback loop. For each model about 100 parameter sets have been generated as described in the text. One run has been performed for each parameter set and for each run the time series analysis has been done for variable M_A over a time period of 10000 min.

For both models, we performed stochastic simulations for each parameter set and systematically calculated the half-life of the autocorrelation and the standard deviation of the period distribution. The distributions of these two quantifiers are given in figure 4.19. A Wilcoxon rank sum test returned $p-$values of 2.01×10^{-26} for the auto-correlation and 1.9×10^{-26} for the period distribution, ensuring that we have two distinct distributions. This data thus confirms that the model with auto-catalysis is more robust than the model without auto-catalysis, regardless of the parameter values.

4.5.4 Conclusion

While a negative-feedback circuit is necessary and sufficient to have limit-cycle oscillations, the role of positive feedbacks is not clear. Thus questions such as understanding the role and advantage of additional positive feedback loops observed in most natural cellular oscillators arise. Besides frequency tunability [117] and oscillations amplification [182], another possible role is illustrated here: positive feedback loops may increase the robustness of oscillator with respect to molecular noise.

This minimal model of the mitotic cycle can be seen as a prototypical cascade model and is therefore useful to investigate questions about design. First, with our phase space analysis, we observed that the positive feedback induces bistability and hysteresis which affect speed and attraction of the limit cycle. As already noticed in other works [41, 161], the spreading of stochastic trajectories along the deterministic cycle is correlated with its attraction properties. These two observations may explain the increase of robustness observed in models based on interlocked positive and negative feedback loops. Second, our glocal approach shows that the increase of robustness due to the additional loop is general and not an artifact from a specific parameter vector. The present study suggests that positive feedback loops, also occurring in circadian clocks [75, 107, 60], may also play a role in the robustness of the oscillations with respect to molecular noise.

4.6 Design of a Robust Synthetic Circuit

This work was done in collaboration with M. Miller and R. Weiss from MIT. It presents the design, optimization and analysis of a large scale synthetic gene regulatory network that controls the population dynamics of engineered stem cells and adult cells in a multicellular environment. Robustness was a major concern while building the circuit as reliability of such a system is essential for it to be functional when implemented *in vivo*. We used principles of the glocal analysis to assess the robustness of the different systems. We also used the global sampling and correlation analysis to give direction for the optimization of the parameters.

The general property of balancing growth, death, and differentiation of multiple cell-types within a multicellular community can be defined as tissue homeostasis. Previous work has demonstrated that population control of bacteria and yeast [128], mammalian cell proliferation [185], and stem cell differentiation [186] can be manipulated artificially using appropriate interactions with natural control mechanisms. Based on these results and various recent accomplishments in synthetic biology [118, 182, 121, 122, 119], we designed a synthetic gene network that copes with natural mechanisms of cell-fate regulation and independently directs stem cell differentiation, proliferation, or quiescence. Our engineered circuit may be employed to regulate tissue homeostasis both *in vitro* where the cell culture is removed from natural cues (e.g. tissue engineering), and *in vivo* when natural systems fail. An example of such deficient system is found in Type I diabetes, where natural populations of insulin-producing β-cells are destroyed due to autoimmune defects. As a possible therapy for diabetes, stem cell and β-cell transplantations have been studied, but initial results suggest that the transplanted cells become either tumorigenic or depleted within a few months [187, 188]. The system we propose may be a new treatment as it regulates stem cell proliferation and differentiation into insulin producing β-cells in order to maintain a steady level of β-cells despite constant destruction of the β-cells by the immune system.

To manage the complexity of the design process, we used a modular approach: we create the systems iteratively such that each additional module contributes to the overall system performance. Each module includes transcriptional regulatory elements and fulfills desired input/output (I/O) properties. Having to face large and complex circuits, we used a wide panel of complementary models to simulate the systems at a different scale. We start by analyzing the high level population dynamics with ordinary differential equations (ODE),

but logic abstraction has limits in biological systems. Stochastic effects, feedback control, and module interdependence have significant consequences at multiple levels: cellular and molecular. To capture many of these effects we use a Langevin approach [10] with Hill terms for protein expression as an intermediate level of abstraction. In this part, we computationally modeled the time evolution of each cellular component, within a multicellular simulation environment. This model is a compromise between a realistic simulation of the molecular noise and computational efficiency, and therefore allows sampling of the parameter space within a reasonable computational time. For the most detailed level of modeling, we used a Gillespie simulation [15] where all binding, transcription, and degradation events are explicit. This last stage allows making adjustments at finer levels of granularity, tuning the performance of an individual element. At each level of analysis and optimization, we used global sampling of the parameter space and correlation analysis to optimize each individual part of the system. With our modular approach, the local optimization results in an increase of performance and robustness of the system as a whole.

4.6.1 System Design

Stem cell differentiation is a multistage process that typically takes up to several weeks to complete in mammalian cells [189]: for example, differentiation of human embryonic stem cells into insulin producing cells occurs under administration of exogenous growth factors and involves first endoderm induction, followed by pancreatic specialization, and finally expansion and maturation [190]. We model this process with four populations. The first one are stem cells (population size S) that grow at a constant birth rate k_b. When stem cells maturate, they go through two intermediate populations of endodermic and pancreatic cells (E and P respectively) before finally giving rise to β-cells (B) which are in turn killed at a constant rate k_k. We describe the sequential maturation of S into E, P, and β-cells by first-order reactions with rates k_{c1}, k_{c2} and k_d. The simplest possible tissue homeostasis system would only need a mechanism that causes cells to start maturation and eventually differentiate in β-cells:

$$\begin{aligned}
\frac{\mathrm{d}S}{\mathrm{d}t} &= k_b S(t) - k_{c1} S(t) \\
\frac{\mathrm{d}E}{\mathrm{d}t} &= k_{c1} S(t) - k_{c2} E(t) \\
\frac{\mathrm{d}P}{\mathrm{d}t} &= k_{c2} E(t) - k_d P(t) \\
\frac{\mathrm{d}B}{\mathrm{d}t} &= k_d P(t) - k_k B(t)
\end{aligned} \tag{4.12}$$

For this trivial system, non-zero equilibrium exists only if $k_b = k_{c1}$, for any sized equilibrium population $S_0 > 0$. The equilibrium stem cells populations S_0 and the equilibrium β-cell population ($B_0 = S_0 \cdot k_{c1}/k_k$) are sensitive to small deviations of k_{c1}/k_b, which results in uncontrolled proliferation or depletion of stem cells [187, 188].

System 1 – Two Feedback Control Differentiation

In the following systems, we design feedback control through the implementation of two artificial cell-cell communication components. The "Stem Cell Population Control" (SPC) module allows differentiation only when the population density of self-renewing cells lies above some threshold. We also designed the SPC to suppress proliferation through the expression of a growth arrest factor (GAF), currently under development in the lab. The "β-Cell Population" (BCP) module produces high output and inhibits differentiation when the density of β-cells reaches a threshold (Fig. 4.20A). Consequently, system 1 allows differentiation only when there is a high density of stem cells (the SPC module output is high) and a low density of β-cells (the BPC module output is low) as described in the following equations:

$$\begin{aligned}
\frac{\mathrm{d}S}{\mathrm{d}t} &= k_b S(t) \cdot \frac{K_S^n}{K_S^n + S(t)^n} - k_{c1} S(t) \cdot \frac{S(t)^n}{K_S^n + S(t)^n} \cdot \frac{K_B^n}{K_B^n + B(t)^n} \\
\frac{\mathrm{d}E}{\mathrm{d}t} &= k_{c1} S(t) \cdot \frac{S(t)^n}{K_S^n + S(t)^n} \frac{K_B^n}{K_B^n + B(t)^n} - k_{c2} E(t) \\
\frac{\mathrm{d}P}{\mathrm{d}t} &= k_{c2} E(t) - k_d P(t) \\
\frac{\mathrm{d}B}{\mathrm{d}t} &= k_d P(t) - k_k B(t)
\end{aligned} \tag{4.13}$$

For the feedback, we use a highly cooperative Hill functions ($n \geq 4$), where K_S and K_B represent the SPC and BPC module thresholds, respectively. If the

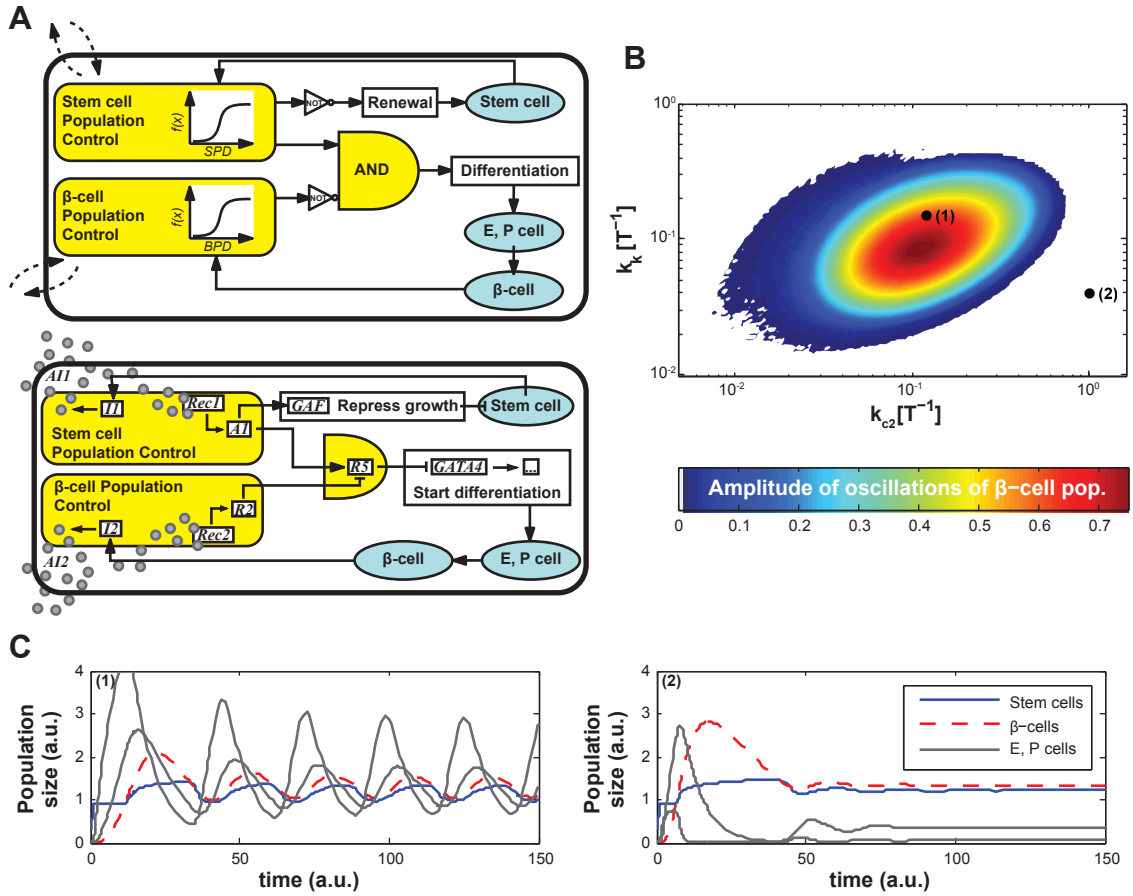

Figure 4.20: Implementation of system 1 and results. (A) Circuit diagram: two Population Control modules sense the density of stem and β-cells. The AND gate integrates the output of the modules to induce differentiation. Gray circles represent signaling molecules that diffuses from one cell to another. (B) Projection in the (k_{c2}, k_k) plan of the region of the parameter space where oscillations of the populations are occurring for the system 1 (Eq. 4.13). Other parameters are $k_b = 1.5$, $k_{c1} = 5$, $k_d = 0.1$ and $n = 16$. (C) Two examples of population evolution: (1) shows sustained oscillations (points 1 in figure B, $k_{c2} = 0.18$ and $k_k = 0.15$); (2) shows an asymptotically stable steady state (point 2 in figure B, $k_{c2} = 1$ and $k_k = 0.04$); other parameters are $k_b = 1.5$, $k_{c1} = 5$, $k_d = 0.1$ and $n = 16$.

differentiation process is long (e.g., 20 days [190]), k_{c1}, k_{c2} and k_d are low and the system can show undesirable oscillations for a strongly nonlinear feedback ($n > 8$) (Fig. 4.20D-E). Even if we engineer feedback within intermediate maturing populations (e.g. E), there realistically remains at least a two day delay.

System 2 – Feedback Based on a Toggle Switch

To minimize delay, we introduce a commitment module that decouples feedback control from the slow differentiation process. We design commitment to occur through a one-way toggle switch that in turn activates the differentiation module. We engineer the feedback control to be immediately downstream of the commitment toggle switch rather than following the full differentiation process (Fig. 4.21A). The state of the one-way switch defines whether or not the cell has irreversibly committed to differentiate, and this status feeds back into what we now term the "Uncommitted Population Control" (UPC) and "Committed Population Control" (CPC) modules. The output of the CPC module is based the number of cells at any stage of the differentiation process. Consequently, we gain a faster feedback response in exchange for assuming that a relatively constant fraction of cells successfully differentiates upon commitment. In the equations for the committed system, we correct the feedback such that the rate for the first stage of differentiation ($S \rightarrow E$ in (4.13)) is

$$k_{c1} S(t) \cdot \frac{S(t)^n}{K_S^n + S(t)^n} \cdot \frac{K_C^n}{K_C^n + (E(t) + P(t) + B(t))^n} \tag{4.14}$$

Compared to the previous system, the population sizes stabilize quickly to an equilibrium point in system 2 (Fig. 4.22). We further tested different initial conditions and parameter vectors. We found, for a given parameter set, that all trajectories coming for the different initial conditions converge to a unique equilibrium point.

In further analysis, we simplified our model to a two-population system: as E, P and B populations are identical with respect to feedback, we merge them into the committed population C. The equations of this reduced system are:

$$\begin{aligned} \frac{\mathrm{d}S}{\mathrm{d}t} &= k_b S(t) \cdot \frac{K_S^n}{K_S^n + S(t)^n} - k_d S(t) \cdot \frac{S(t)^n}{K_S^n + S(t)^n} \cdot \frac{K_C^n}{K_C^n + C(t)^n} \\ \frac{\mathrm{d}C}{\mathrm{d}t} &= k_d S(t) \cdot \frac{S(t)^n}{K_S^n + S(t)^n} \cdot \frac{K_C^n}{K_C^n + C(t)^n} - k'_k C(t) \end{aligned} \tag{4.15}$$

In this system of ODEs, the actual β-cell population B is a fraction of this committed population C (at equilibrium $B_0 = \frac{k_d k_{c2}}{k_{c2} k_d + k_{c2} k_k + k_d k_k} C_0$) and the killing rate should be consequently corrected to $k'_k = \frac{k_k k_d k_{c2}}{k_{c2} k_d + k_{c2} k_k + k_d k_k}$. We note that a two dimensional system may not fully restore the diversity of a four dimensional system, as for example chaotic behavior is not possible. In the working range of our particular system with feedback, this two population

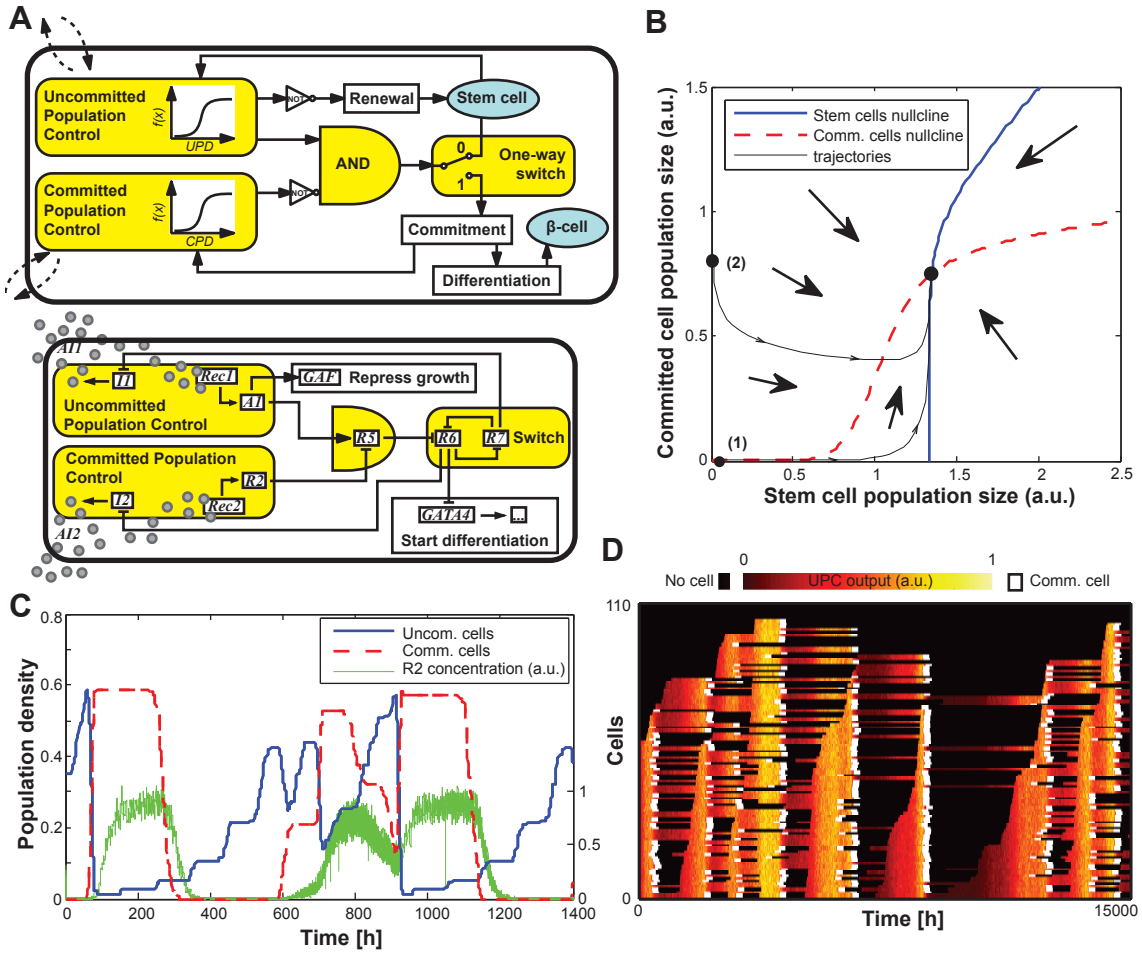

Figure 4.21: Implementation and results for system 2. (A) Circuit diagram: two Population Control modules sense the density of stem and committed cells. The AND gate integrates the output of the modules to induce commitment trough the switch state. Gray circles represent signaling molecules that diffuses from one cell to another. (B) Phase space diagram of the system 2 ($k_b = 1$, $k_d = 0.1$, $k_k = 0.15$ and $n = 8$): there is a unique non-trivial asymptotically stable equilibrium point to which all trajectories converge. Black lines correspond to two different trajectories: (1) start with low populations of stem and β-cells whereas (2) has a large population of β-cells. (C) Stochastic trajectories for the system 2 (stem cells in blue, committed cells in red) for a simulation starting with a small stem cell population. The parameters are chosen to emphasize the issues with sudden commitment. In green is plotted the output of the Committed Population Control module ($R2$) in some stem cells (right axis, a.u.). A high value inhibits commitment in individual stem cells. (D) Heat map with the concentration of $A1$, the output component of the Uncommitted Population Control module. Each line corresponds to a single cell which is uncommitted when colored in red-yellow or committed when white. As soon as commitment is possible (yellow), most of the stem cells will commit suddenly.

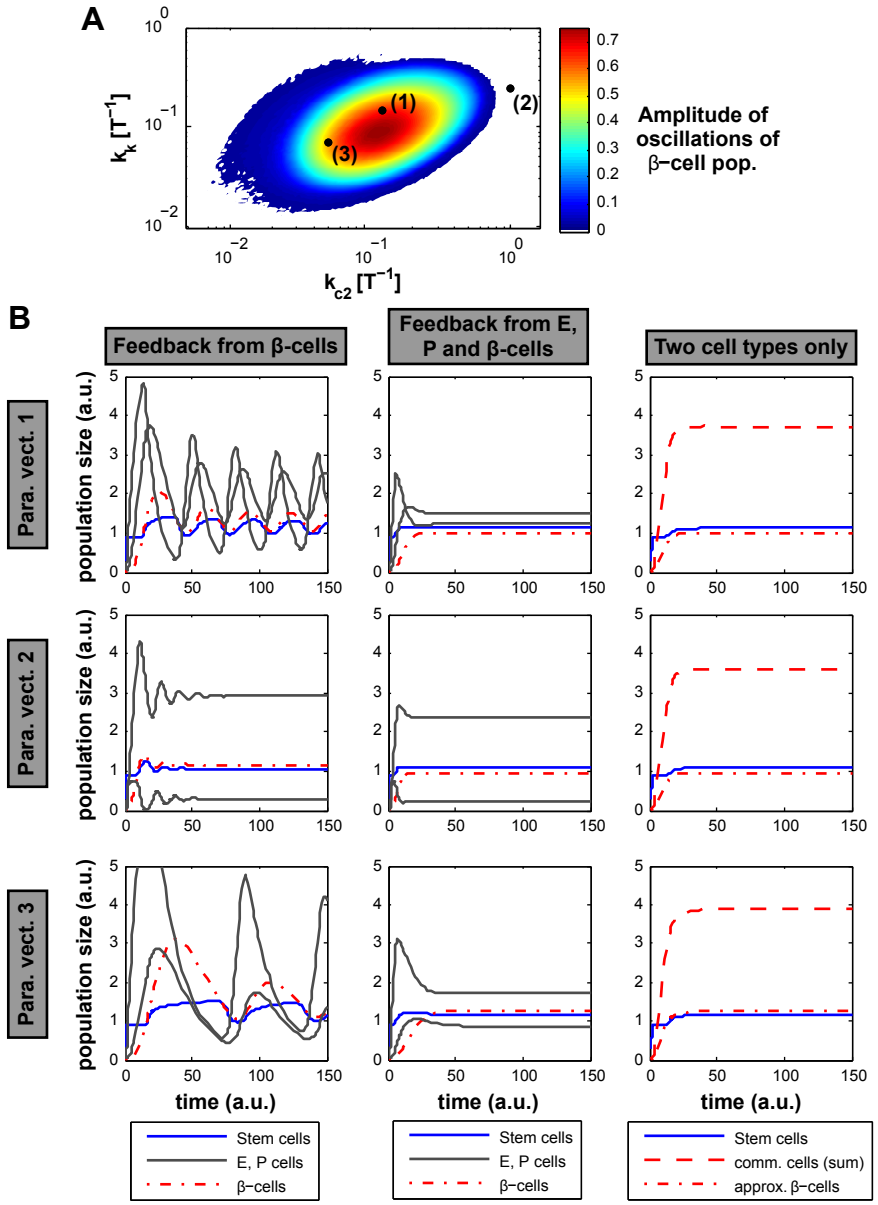

Figure 4.22: Effect of the feedback coming from all committed cells on the four population system. (A) Projection in the (k_{c2}, k_k) plan of the region of the parameter space where oscillations of the populations are occurring for the system 1. Other parameters are $k_b = 1.5$, $k_{c1} = 5$, $k_d = 0.1$ and $n = 16$. (B) Three examples of trajectories with feedback from the β-cells (left column) and all committed cells, equivalent to system 2 (middle column): (1) show sustained oscillations (points 1 in panel A, $k_{c2} = 0.18$ and $k_k = 0.15$); (2) shows a asymptotically stable steady state (point 2 in panel A, $k_{c2} = 1$ and $k_k = 0.04$); (3) shows sustained oscillations with a longer period (point panel A, $k_{c2} = 0.05$ and $k_k = 0.07$); other parameters are $k_b = 1.5$, $k_{c1} = 5$, $k_d = 0.1$ and $n = 16$. The last column shows an equivalent two population system with stem cells (blue line) and committed cells (red lines). The β-cell population size is extrapolated.

model shows a qualitatively similar behavior and may be a correct substitute for the scope of our analysis (Fig. 4.22).

In the following, we focus on maintaining a constant population of committed rather than differentiated cells. With this simplification, we can prove that the system has a non-trivial asymptotically stable equilibrium point when $k_b > k_d$ (Fig. 4.21). Moreover, at this equilibrium, we have that $S \geq K_S$ and $B \geq K_B$, provided that the parameters satisfy $\frac{k_d}{k'_k} \geq 4\frac{K_B}{K_S}$ [139].

The combination of the signaling feedback and toggle switch ostensibly provides a mechanism of measuring overall population density. However, these systems generally involve low molecular count and small population size, especially when considering physiologically localized signal diffusion. Consequent stochastic effects can significantly impact on system performance. To better understand these issues, we performed simulations that account for limited population size and molecular noise due to gene expression using stochastic differential equations [10] (see section C.1 in appendix). These simulations reveal that homogeneity within the stem cell population can present a significant problem. More specifically, if the committed population is low, the signal for commitment is strong in all stem cells and many simultaneously commit, resulting in a homeostasis failure (Fig. 4.21C,D). We can optimize this system to have high diffusion rate and a rapid state switch of the toggle. With such fast feedback, system 2 is able to maintain homeostasis with moderate fluctuations in some situations, for example, with a high initial committed cell population. But in practice it may not be possible to implement such a fast response. Moreover, significant perturbations to the system, for example resulting from injury or elevated autoimmune response, are likely to create situations where system 2 cannot properly control homeostasis.

System 3 – Addition of an Oscillator

Heterogeneity is necessary among individual cells to facilitate a proportionate and homeostatic system response to population-wide cues. Therefore, we design a mechanism to balance commitment and desynchronize single-cell responses to feedback signals. For this system, we incorporate an asynchronous oscillator into the design as a generator of intrinsic heterogeneity (figure 4.23A). This module interacts with the system such that a cell's commitment to differentiation can only occur when that cell's oscillator peaks. As oscillations in individual cells grow out of phase from each other due to stochastic effects,

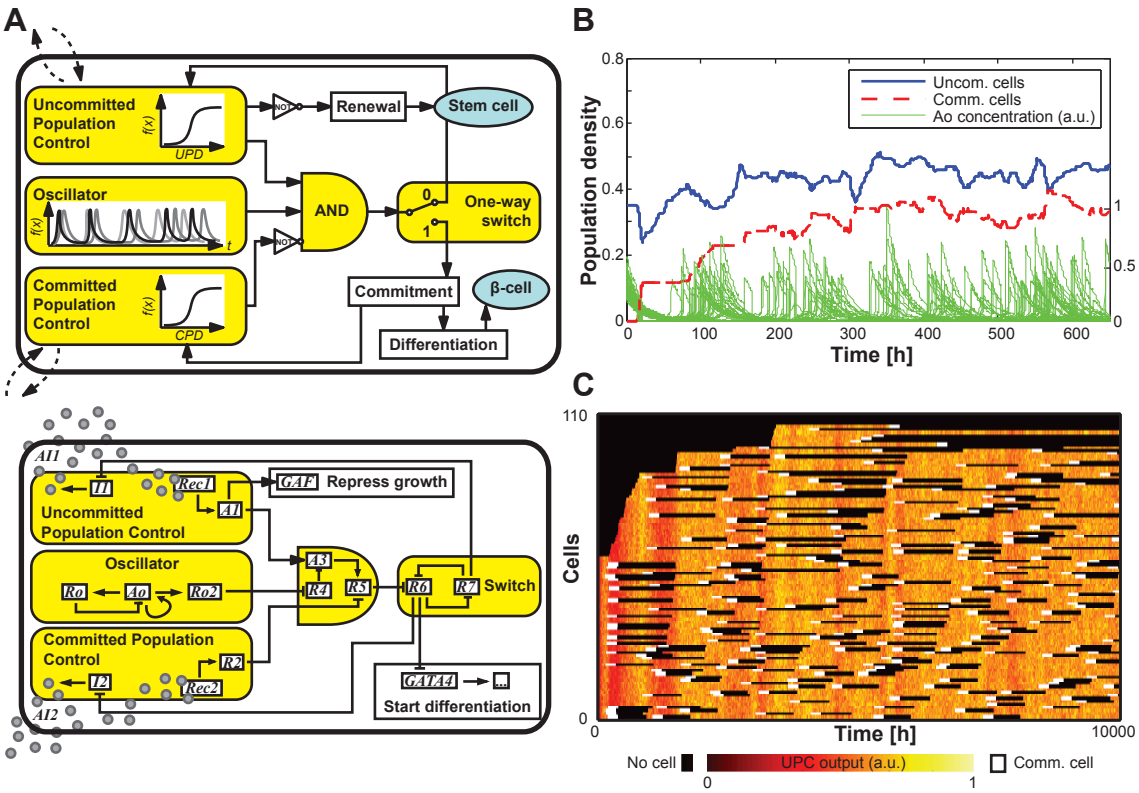

Figure 4.23: Implementation and results for system 3. (A) Circuit diagram: two Population Control modules sense the density of stem and committed cells. An oscillator is added in comparison to system 2. The AND gate integrates the output of the population control modules and the oscillator to induce commitment trough the switch state. Gray circles represent signaling molecules that diffuses from one cell to another. (B) Time trajectories for the system 3 (stem cells in blue, committed cells in red) for a simulation starting with a small stem cell population (see table C.5 in appendix for simulation parameters). In green is plotted the main component of the oscillator (*Ao*) in some stem cells (right axis, arbitrary units). A high value allows commitment in individual cells. (C) Heat map with the concentration of *A*1, the output component of the Uncommitted Population Control module. Due to the oscillator, just a fraction commit when *A*1 is high.

coupling the asynchronous oscillator to cell-fate decisions prevents all cells in a population from responding simultaneously to the same commitment signals.

We use the simplest possible oscillator, made of a component *Ao* that activates itself and regulates the expression of a repressor *Ro* that inhibits *Ao*. The readout of the module is implemented with a second repressor *Ro*2, and cells can only commit when *Ro*2 peaks. Simulations indicate that with

the oscillator, our tissue homeostasis system functions as desired despite the fact that feedback signaling cues to commit remain synchronized even after homeostasis is established (Fig. 4.23B-C).

System 4 – Addition of a Throttle

To address the problem of population-wide commitment, we also explore another solution based on quick lateral inhibition acting as a throttle on the commitment process (Fig. 4.24A). Through this fast lateral feedback, a cell starting to commit blocks the commitment of adjacent cells. The throttle approach necessitates a third signaling molecule, $AI3$, that diffuses like $AI1$ and $AI2$. When the toggle switches, an activator At controls transient $AI3$ release. In adjacent cells, $AI3$, through activation of $Rec3$, inhibits the production of the component $R5$ that induces the switching of the toggle. The rest of the circuit remains similar to previous systems. Simulations indicate that when the populations reach their steady state value, the throttle effectively prevents the simultaneous commitment of too many cells and therefore maintains homeostasis (Fig. 4.24B-C).

4.6.2 Analysis and Optimization

Having working modules as the ones we presented, the next stage is to link them as described in the section 4.6.1. Yet, achieving a large scale functional network is not trivial and *in silico* analysis may be necessary. In this section, we analyze and optimize at four scales of increasing details: multicellular systems, cellular networks, individual modules and biochemical reactions. At the highest multicellular level, using the theory of population dynamics, we can find the necessary conditions for homeostasis. Second, we perform simulations at the cellular level and integrate a set of logic modules that are embedded within each cell, where each module exhibits a specific output as a function of input values. This allows us to optimize the modules independently and find the best regime for each system. Third, we focus on the specific dynamics of the control module and determine the critical parameters that need to be adjusted. Our intent is that the optimization of the module's function achieves the high-level system objective. Finally, we optimize the population control module at the biochemical reaction level, searching for association and dissociation rates that improve module performance.

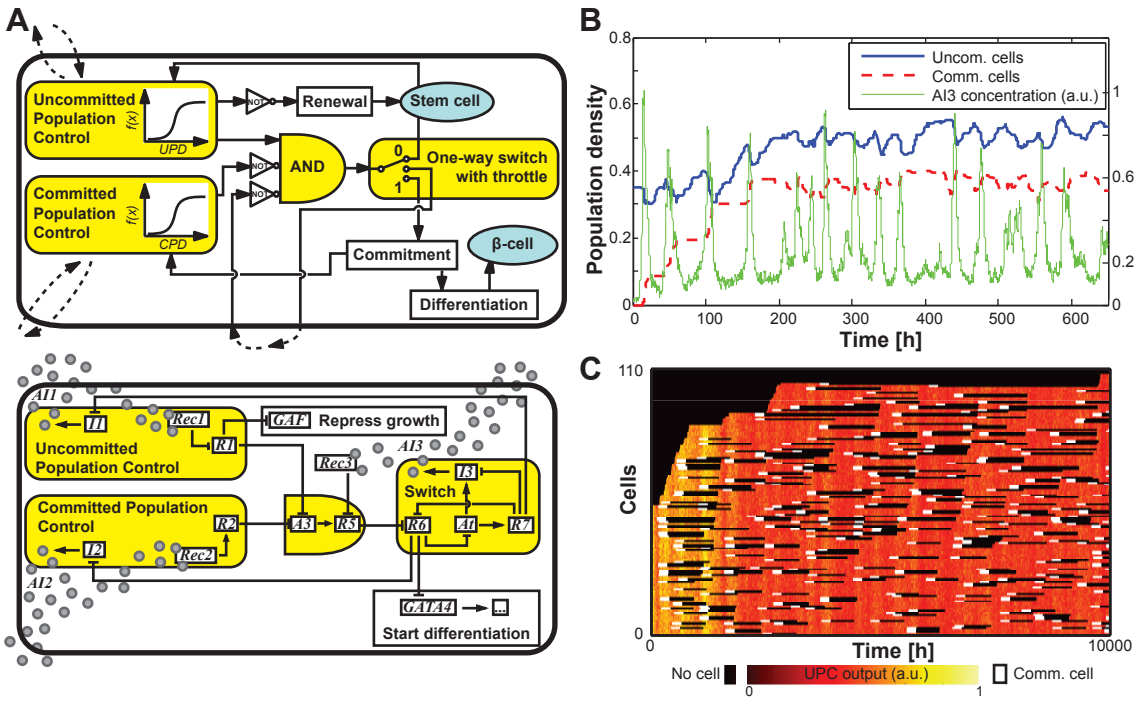

Figure 4.24: Implementation and results for system 4. (A) Circuit diagram: two Population Control modules sense the density of stem and committed cells. The AND gate integrates the output of the population control to induce commitment trough the switch state. During the commitment process, the throttle is activated and feeds in the AND gate of adjacent cells. Gray circles represent signaling molecules that diffuses from one cell to another. (B) Time trajectories for the system 4 (stem cells in blue, committed cells in red) for a simulation starting with a small stem cell population (see table C.5 in appendix for simulation parameters). In green is plotted the throttle signaling component ($AI3$) in the external medium (right axis, arbitrary units). A high value prevents commitment through the population. (C) Heat map with the concentration of $R1$, the output component of the Uncommitted Population Control module. Due to the throttle, just a fraction commit when $R1$ is low.

Spatial Distribution of the Different Populations

We start the analysis at the highest level by observing how the two populations are distributed in space and if the different mechanisms of regulation induce different patterning of the system. For this analysis, we changed the simulations to assign a specific position to each cell in a 3D grid (see Appendix, section C.1.8). For all cells, we calculated the ratio of committed to uncommitted neighbors at various distances. We use normalized Z-scores as indicators of how biased the distribution of committed/uncommitted neighbors is at a given

distance for a given reference cell-type (see section C.2.1 in appendix for the details of the algorithm).

This analysis reveals how competing mechanisms for tissue homeostasis work. In all systems, committed cells express $AI2$ in order to inhibit further commitment in the population. Results indicate that neighbors of committed cells are biased to be uncommitted at close distances and to be committed further away (Fig. 4.25). The lateral inhibition of commitment by already committed cells helps explain this phenomenon. When we incorporate the oscillator in system 3, the trend of lateral inhibition mostly disappears for immediately adjacent cells (Fig. 4.25B). In this case, committed cells tend to form small clusters as neighboring cells are more likely to have spawned from the same parent cell and are more likely to have in-phase relaxation oscillators. System 4 implements a third QS signal ($AI3$) that inhibits commitment locally, further strengthening lateral inhibition compared to system 2 (Fig. 4.25C). If we decrease molecular noise in the Langevin simulations by increasing the 'cell volume' which is related to the number of molecules in each cell (see section C.1 in appendix), the oscillators stay in-phase for longer, and this effect amplifies the spatial clustering produced in system 3 (Fig. 4.25E). Increased cell volume also leads to spatial clustering in system 2, mainly due to relatively simultaneous commitment of large portions of the population in this case (Fig. 4.25D).

Robustness to Variation of the Killing Rate k_k

Another analysis at the population level is how the system reacts when confronted to variation of the average survival time of the committed population ($1/k_k$) as this parameter can fluctuate *in vivo*. Control of growth and commitment in the undifferentiated population are designed to be robust to external factors. To test this aspect, we simulate the systems with different values of the killing rate k_k and measure the S/N value (signal to noise ratio defined as the relative variance of the committed population density, see (C.1) in Appendix). The results are plotted in figure 4.26A-B as the ratio of division rate over killing rate (k_b/k_k). In general, systems 3 and 4 are not affected by a different lifetime and they both perform significantly better than system 2. We also analyze the effect of the parameter k_k on the population size. When the ratio of committed cell-death rate to uncommitted cell-growth rate is close to one, equilibrium populations remain near the desired homeostatic levels (Fig. 4.26B). At death/growth ratios below one, however, when survival time is shorter than the division time, the equilibrium committed population levels drop because population renewal cannot compensate losses (Fig. 4.26B).

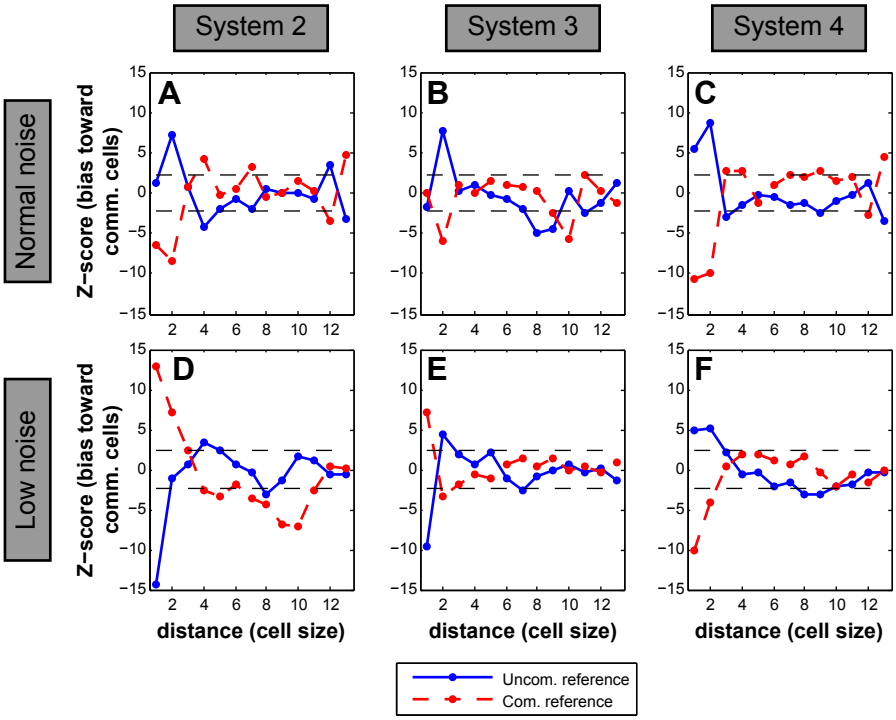

Figure 4.25: Spatial distribution of the cells in the three systems. The bias toward a higher concentration of committed cell (positive value of the Z-score) is plotted at a given distance (blue line for the neighborhood of uncommitted cells, red line for the neighborhood of committed cells). Values outside the dashed lines are significant. System 2 (A) shows a bias for cell alternation on short distances: uncommitted cells are close to uncommitted cells. On the contrary, system 3 (B) has a less bias, especially in the neighborhood of committed cells. Finally system 4 (C) has the most significant bias. (D-F) Same simulation results as A-C with lower molecular noise which emphasizes the differences between the three systems.

The relation between committed population density and killing rate could be compared to the results of the ODE model with two populations (figure 4.27). We determined three properties for the sensitivity of both uncommitted and committed populations to variations of the β-cell killing rate (k_k). First, the population of uncommitted cells is well controlled and remains almost constant for all ratios of division rate over killing rate (k_b/k_k). Second, for high ratios, the population of committed cells follows a power law with an exponent close to $1/n$ where n is the Hill coefficient in the feedback function. Third, for low ratios, on the contrary, the population decreases linearly with the killing rate. For an ODE model with a Hill coefficient $n = 16$, the uncommitted population is very robust to variations of the killing rate. For high ratios, the

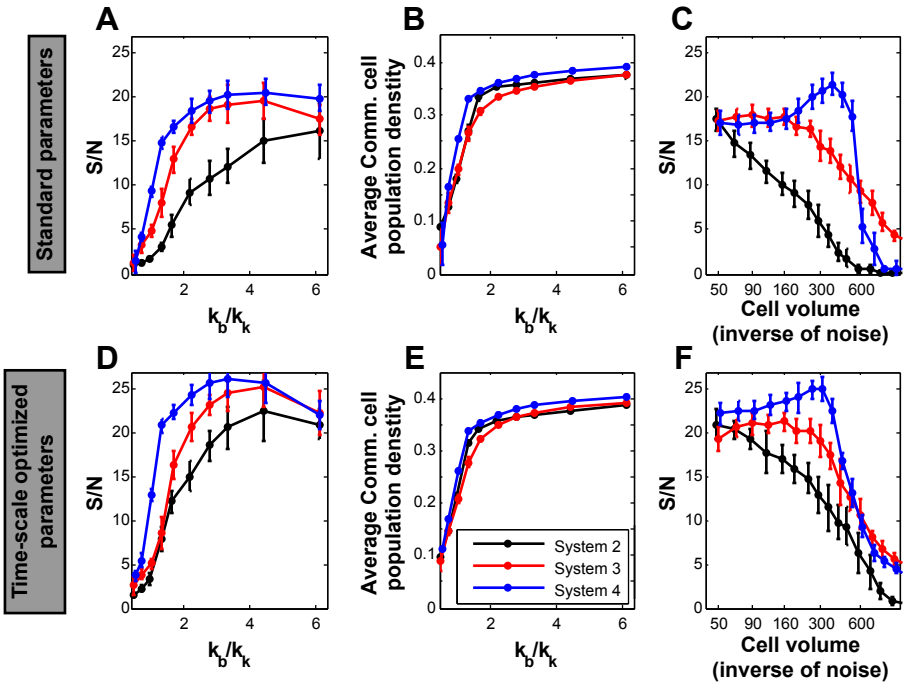

Figure 4.26: Population level properties of systems 2, 3 and 4. (A) Signal to noise value (S/N) for different ratio of stem cell division rate (k_b) and β-cell killing rate (k_k). Both systems 3 and 4 are able to maintain a better performance for short survival time. For value lower than 1, all systems show a strong decrease of performance. (B) Committed cell population density for different ratio of stem cell division rate (k_b) and β-cell killing rate (k_k). All systems have the same average population that decreases for large killing rate. (C) Signal to noise value (S/N) for different cell volume Ω (corresponds to the number of molecules in each cell). The S/N value of system 2 is constantly decreasing, whereas systems 3 and 4 are able to maintain a better performance up to a volume of $\Omega = 400$. For all systems, performance decreases significantly for very low molecular noise ($\Omega > 1000$). (D-F) Same simulations with the time-scaled optimized parameter values. S/N values are significantly improved for all three models by about 5 units.

committed population is also robust due to its low exponent of 0.07 (close the theoretical value $1/n = 0.0625$ [139]), but follows a linear dependence (exponent of 1.00) for high killing rates. These results are confirmed by a theoretical analysis of the system [139]. But more interestingly, the results of the stochastic simulations with the Langevin models are qualitatively similar (figure 4.27B,D). The fits of system 2 – which is the closest to the ODE model – have power laws with exponents 0.06 and 1.14 for respectively high and low ratios. These values are very close to both the theoretical analysis and

the ODE simulations of the simplified model, showing the consistency of our analysis with multiple level of modeling. Note that systems 3 and 4 show small differences for low ratio k_b/k_k, but the qualitative behavior is similar.

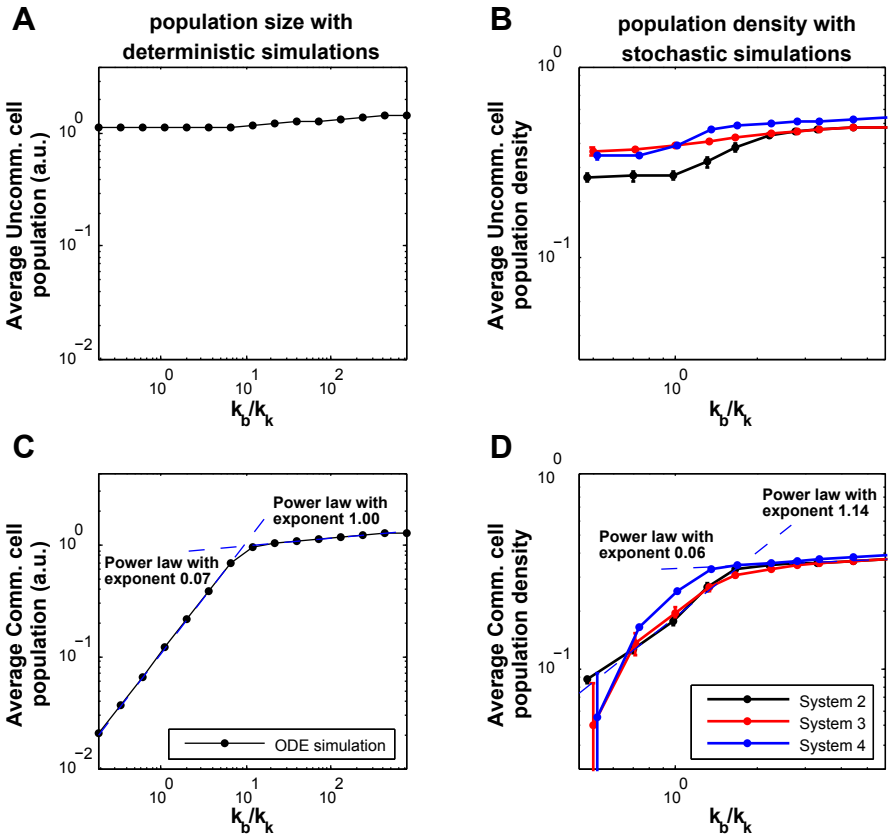

Figure 4.27: Population density for different ratio of division over killing rates. Deterministic simulation with a two-population model with $n = 16$ (A,C) and stochastic simulations of the systems 2, 3 and 4 (B,D) show qualitatively similar results. (A-B) the population of uncommitted cells remains constant with a small decrease for low rate ratio. (C-D) the population of committed cells follows a power law with an exponent near 1 for low ratio and close to $1/n = 0.625$ for large ratio. Power laws in D are fitted on the results of system 2, the closest system to the ODE model.

Robustness to Variations of Molecular Noise Amplitude

The last population analysis we performed is how molecular noise affects the system. Our systems rely on molecular noise to generate diversity: for example, near the transition region for the UPC module, stochastic effects will often cause cells exposed to similar environmental conditions to respond

very differently (i.e. they commit or remain uncommitted). In this analysis, we change the amplitude of molecular noise in the three systems and clearly see that the performance of system 2 decreases constantly with the number of molecules in the system (Fig. 4.26C). For small values of Ω, systems 3 and 4 show the same performance as system 2, mainly because the large noise present in protein expression overwhelms the control mechanisms and cells commit almost randomly. But for intermediate values, the oscillator and the throttle allows the systems to maintain better homeostasis for a larger range of conditions than system 2. For high Ω values, all systems show a strong decrease of performance that can be explained in system 2 by the lack of molecular noise that induces a coordinate response and gives rise to a sudden commitment. In system 3, the lack of desynchronization of the individual oscillators impedes the S/N value. And finally in system 4, performance is decreased as the throttle cannot prevent differentiation because its effect is too weak.

Analysis and Optimization of the Time Scale of the Different Modules

We will now give directions for optimization of the systems. Our modules have relatively well defined response in isolation, for example the toggle switch is engineered to be bistable and can be implemented using two mutually inhibiting repressors [119]. Assuming an input-output function that meets our design specifications for each module, we study the effect of the time scales of the modules independently. This reflects how fast a module will integrate the incoming signal and change its outcoming signal according to the new input.

We regroup the different parameters according to their module and change the rate constants (see appendix, section C.1 and table C.5) for each component with the following factor: TS_{QS} stands for the time-scale of the quorum signaling molecules (including diffusion), TS_{QM} for the time-scale of the quorum sensing module ($A1$, $R2$, ...), other time-scales are specific to the components $R5$, $R6$, $R7$ and At. In this reduced parameter space, we performed a random sampling over one order of magnitude for each time-scale parameter and evaluated the S/N value. Figure 4.28 shows the distribution and range of parameters over which we sampled, along with the impact of variation in individual module time-scales on system performance, as defined by S/N. Although informative, the first-order correlations as shown by scatter plots in Fig. 4.28 can be insufficient descriptors of parametric sensitivity for complex nonlinear systems. We use the Random-Sampling High Dimensional Model Representation (RS-HDMR) algorithm [149] to understand both the individual and cooperative nonlinear effects of time-scale modulation on S/N (Fig. 4.29).

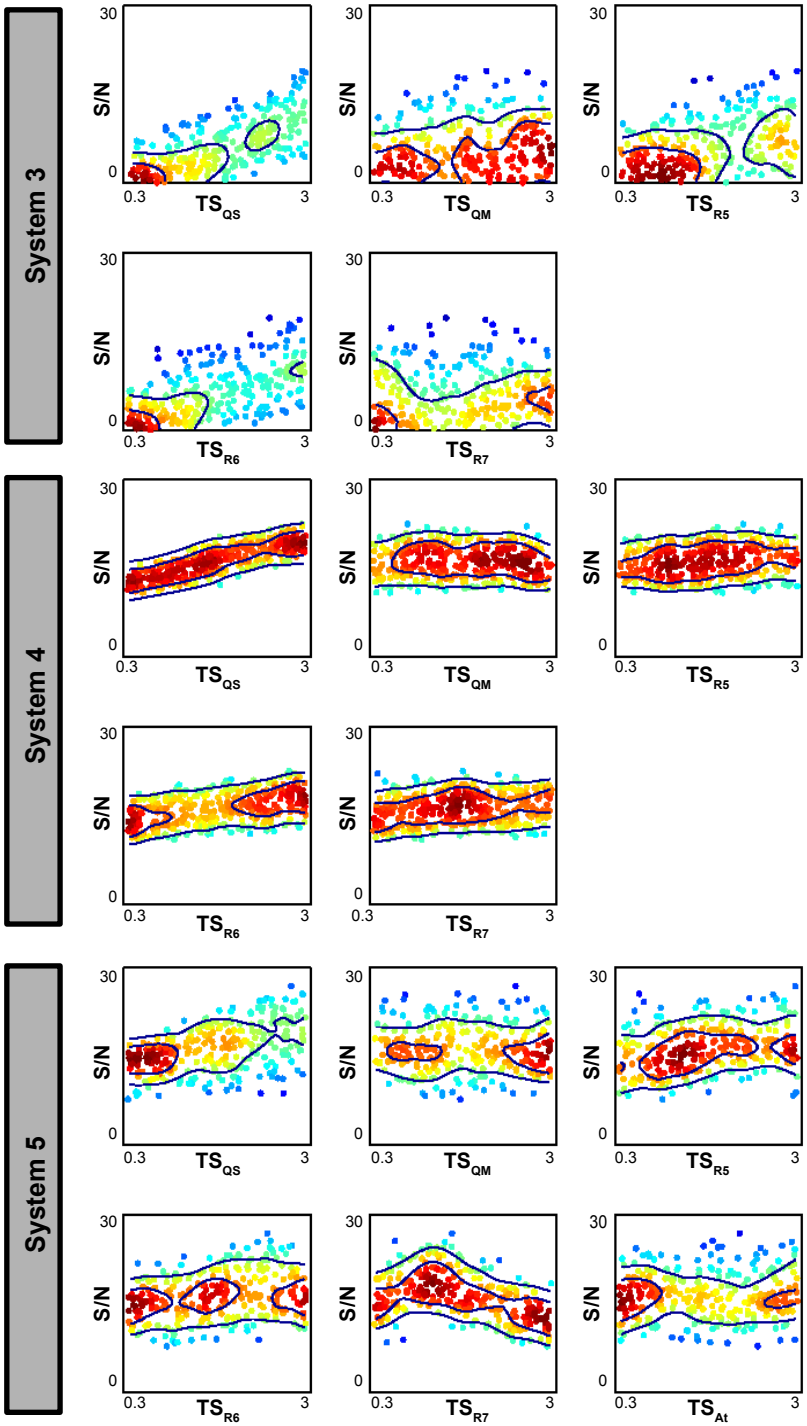

Figure 4.28: Parametric sampling distribution for modular time-scale analysis. Time-scale parameters were randomly varied across one order of magnitude for the time-scale of each module or component to produce roughly 360 parameter sets for each system (3, 4, and 5). Simulations of each time-scale parameter vector yielded a corresponding S/N value, which is plotted here as a function of the individual time-scale parameters. Each point represents an individual parameter vector. Warmer colors indicate higher point density; contour lines also indicate point density. TS_{QS} stands the time-scale of the quorum signaling molecules (including diffusion), TS_{QM} for the time-scale of the quorum sensing module ($A1$, $R2$, ...), other time-scales are specific to the components $R5$, $R6$, $R7$ and At.

For system 2, the first-order RS-HDMR component functions clearly show that a fast diffusion and a rapid toggle switch (through $R6$ dynamics) are necessary for a good performance of the system (Fig. 4.29A). Second-order RS-HDMR component functions indicate cooperative interactions among parameters that may be interpreted as 'inter-modular coupling' (Fig. 4.29B). In system 2, for example, having slow R7 dynamics can increase the beneficial impact of fast diffusion. For system 3, the only significant correlations between performance and time scales are found for the diffusion and, to a lesser extent, the toggle switch dynamics (Fig. 4.29A). In contrast to system 2, RS-HDMR detected no significant second-order component functions. One interpretation of these results is that the oscillator decouples the modules from each other, minimizing cooperative interactions between diffusion and the toggle switch by creating a buffer between the two. The performance dependency on time-scale parameters is more complex for system 4 compared to the other two systems. First-order relationships are less pronounced in system 4, and second order functions show greater significance (Fig. 4.29A-B). In particular, the cooperative interaction of a slow R7 dynamics combined with fast R5 dynamics produces a strong synergistic improvement in S/N. This is understandable as the lateral inhibition could not be effective if the switch is instantaneous. Total sensitivity indices, S_i^T, represent the summed weight of first- and second-order RS-HDMR component functions for each parameter (Fig. 4.29D). For system 2, observed S/N is most sensitive to changes in diffusion. In system 3, the dependency on diffusion is even stronger, whereas for system 4, dependencies are spread.

In general, performance of systems 3 and 4 is significantly better than system 2 (Fig. 4.29C): mean S/N over the entire ensemble of time-scale parameter sets was 16 for both systems 3 and 4, compared to 7 for system 2. Furthermore, systems 3 and 4 are more robust to time-scale variation, with a CV for S/N of 16% and, respectively, 24%. This compares favorably to the CV observed in system 2 (64%).

With the results of the time-scale analysis, we selected individually optimized values for systems 2 to 4 (see table 4.2). Using these values, the performance of all three systems can be improved significantly: the result for the two global analyses (variation of the killing rate k_k and the cell volume Ω) with optimized parameters are plotted in figure 4.26D-F. On average, the optimization allows a gain of 5 units for the S/N value for all systems in all conditions. The major qualitative difference is the ability of the time-scale optimized system 4 to cope with low molecular noise ($\Omega > 1000$) in a comparable way to system 3 (Fig. 4.26F). Note that such optimizations necessitate some

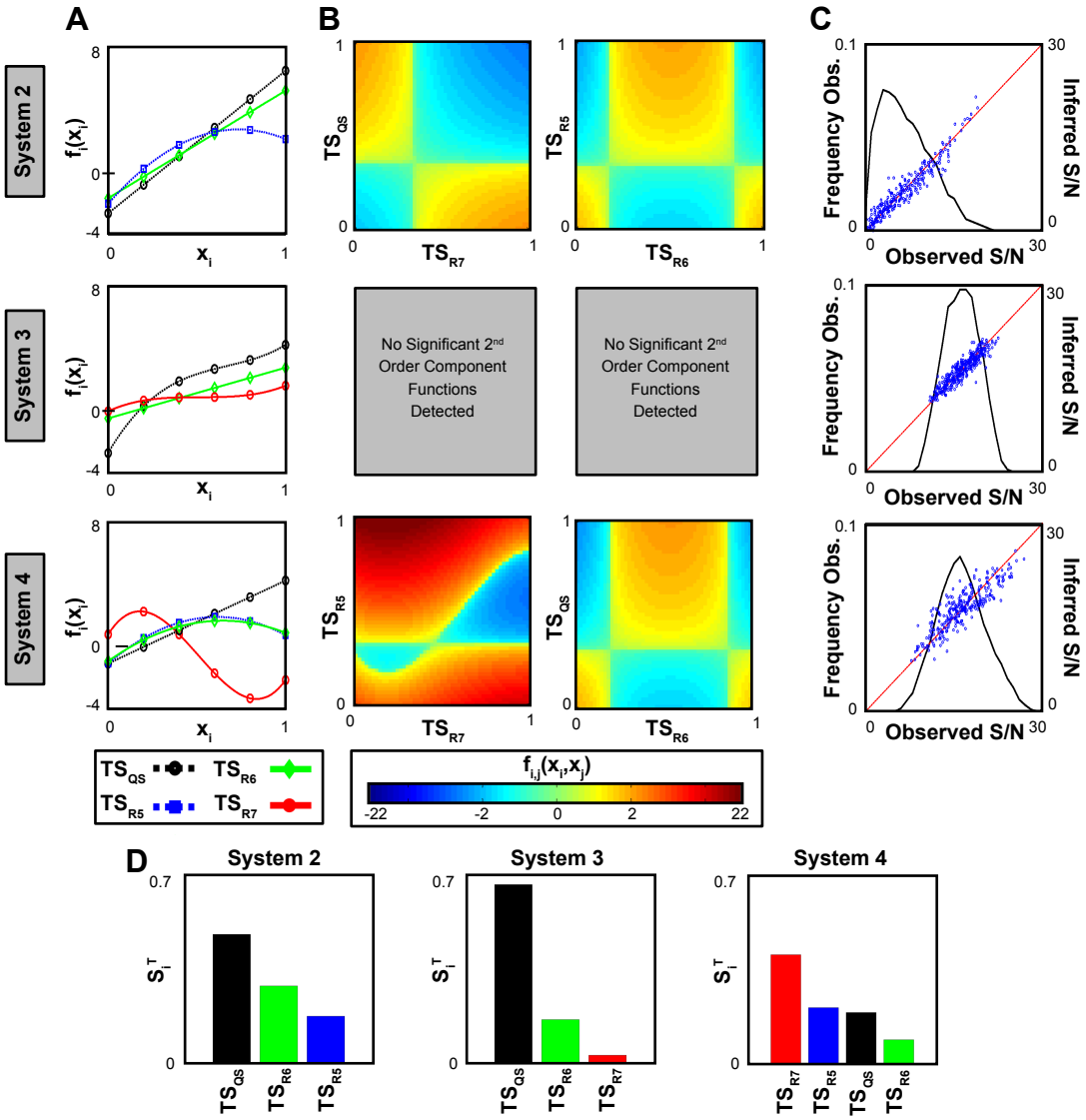

Figure 4.29: Optimization results using RS-HDMR algorithm for systems 2, 3 and 4 at the module levels (time-scales). (A-B) RS-HDMR was used to analyze the sensitivity of systems 2, 3, and 4 to changes in the reaction time-scales of their module components. First- (A) and second-order (B) RS-HDMR component functions describe the relationship between model parameters, which in this case are time-scales linearly normalized to [0,1], and the corresponding S/N observed in the overall system. (C) The distribution of S/N observed in response to time-scale parameter sampling (black) and the RS-HDMR inference accuracy of that variation (blue). (D) Total sensitivity indices (S_i^T) of the module time-scales observed for each system.

	TS_{QS}	TS_{QM}	TS_{R5}	TS_{R6}	TS_{R7}	TS_{At}
System 2	2.31	1.14	1.29	1.67	1.24	–
System 3	2.14	0.91	1.08	1.84	1.11	–
System 4	1.95	0.88	1.38	1.54	0.72	1.01

Table 4.2: Time-scale optimal values for the different modules and systems. These values are not the ones yielding the best S/N score, but are an average of the best 10% parameter sets. We choose this to maintain some robustness to small variations and have parameters that better represent a more realistic optimization than a fine-tuning.

interaction to be faster (for example, $TS_{QS} \geqslant 2$) which may not be possible to implement.

Optimization of the Oscillator and Throttle Module

Most modules work near a steady state and therefore are robust to small fluctuation and also less sensitive to parameter changes. But the oscillator and throttle work transiently with the consequence that the dependence of the systems on their parameter values is stronger. To understand how fluctuations of individual parameters affect the efficiency of the control and which values should be carefully chosen we performed parametric sensitivity analysis again. To obtain the results, we sampled the parameters involved in the oscillator or the throttle (see table C.5 in appendix). For systems 3 and 4, respectively, figures C.1A and C.2A show the distribution and range of parameters over which we sampled, along with the impact of variation in individual module time-scales on system performance, as defined by S/N. We did not test the system 2 as it only contains elements that work at their steady state.

First- and second-order RS-HDMR component functions, along with scatter plots depicting first-order correlations, can be found in supplementary figures C.1B-C. RS-HDMR analysis indicates that for the oscillator, fluctuations in H_{Ro} have the greatest impact on system behavior, accounting for roughly 20% of the total observed variance in S/N (Fig. 4.30A and C.1B-C). H_{Ao-A} and H_{Ro2} are also significant components, especially for the second-order functions. These three parameters account for the activation of the components *Ro*, *Ao* and *Ro*2, respectively, depending on the concentration of *Ao*. Such relation is understandable as these parameters control the durations of the peaks: a larger H_{Ao-A} results in a smaller auto-feedback effect of *Ao* and therefore smaller and shorter peaks. For H_{Ro}, a larger value means that *Ro* will be

less active and therefore the pulse of Ao (and further $Ro2$) will be longer and the control mechanism less efficient. The other parameters correlated with the S/N values (k_p^{Ro2}, k_d^{Ro2} and H_{R4}) are related to the expression of $Ro2$ and the output of the module.

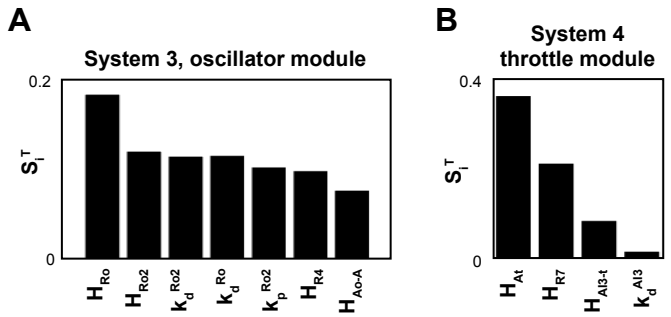

Figure 4.30: Optimization results using RS-HDMR for the additional modules in systems 3 and 4. Effect of variation in the parameters values are summurized in total sensitivity indices (S_i^T) for the oscillator in systems 3 (A) and the throttle in system 4 (B).

In the analysis of the throttle (Fig. 4.30B and C.2B-C), the parameters H_{At} and H_{AI3-t} have strong correlations with the performance of the system. These are the values for the repression of the component At depending on the concentration of $R6$ and, respectively, the activation of $AI3$ depending on the concentration of At. We also see, especially for the second order correlations, an influence of parameter H_{R7} representing the activation threshold of $R7$ expression depending on At concentration. All these parameters control the length of the pulse of $AI3$ to some extent. These results give the directions to design the oscillator and the throttle and, thanks to our modular system, these elements can be optimized and controlled prior to be connected to the whole system.

Optimization of the Quorum Sensing Module

In the Langevin models used for the previous optimization stages, we used a simplified model for gene regulation and protein expression: we hypothesized Hill-type responses, but such responses depend on many biophysical constants and are not straightforward to obtain. An optimization at the molecular level is necessary such that each element shows the desired sigmoidal response. We implement a model of system 3 using the Gillespie algorithm to explicitly simulate all binding and transcription events. Note that the results of this part

have been obtained by M. Miller and the details of the implementation of the Gillespie algorithm can be found in [139].

We focus on the UPC module with the objective to obtain a step-like response to population density, such that the module output is either low or high depending on whether the population falls below or above a given threshold, respectively. Fig. 4.31A-B demonstrates how excess UPC output below the QS threshold (first row) or insufficient output above the threshold (second row) in suboptimal systems can lead to overactive commitment or proliferation, respectively. We again employed RS-HDMR analysis and a genetic algorithm (GA) to optimize the module and understand how to control its behavior. To sample and optimize parameters rapidly, we used a ODE "two-compartment" model of the UPC module rather than the discrete stochastic model [139]. We use positive feedback on the $Rec1/AI1$ expression in our system to engineer a step-like "forward response" of UPC output to increasing cell density. However, this feedback could generate hysteresis: a high UPC output may be maintained as the population density decreases below the threshold level [191]. This hysteresis in UPC output can lead to sub-optimal or even non-functional tissue homeostasis performance (Fig. 4.31C-D, first line) and therefore the objective function should also account for the "reverse response". Eventually, our GA optimization was able to generate an ensemble of UPC networks with positive feedback that exhibit both step-like and non-hysteretic behavior (Fig. 4.31C-D, second line). The effect of optimizing the UPC module on the full-system behavior demonstrates how the subnetwork optimization is critical for the overall system performance.

Rather than converging to unique solutions, the GA optimizations produced diverse ensembles of parameters, each with system behavior closely fitting the objective (Fig. 4.31E). We performed RS-HDMR analysis of the UPC subnetwork to understand how rate constants affect hysteresis. To estimate global sensitivity while limiting ourselves to systems with desired input-output, we examined local parameter "neighborhoods" around each GA-generated vector of optimized parameters from Fig. 4.31E. Our sensitivity analysis suggests that systems displaying similar UPC behavior can have drastically different responses to similar changes in rate-constants: each parameter neighborhood that we analyzed has a distinct signature of parametric sensitivity (Fig. 4.31F). We clustered parametric neighborhoods based on these signatures. Despite differences in individual sensitivities, the clustered sensitivity analysis revealed that the majority of signatures fall into two main clusters, each with distinctive features. For example, in one cluster (red on the dendrogram) the decay rate of the receptor protein Rec1 (Fig. 4.23) significantly affects hysteresis, while

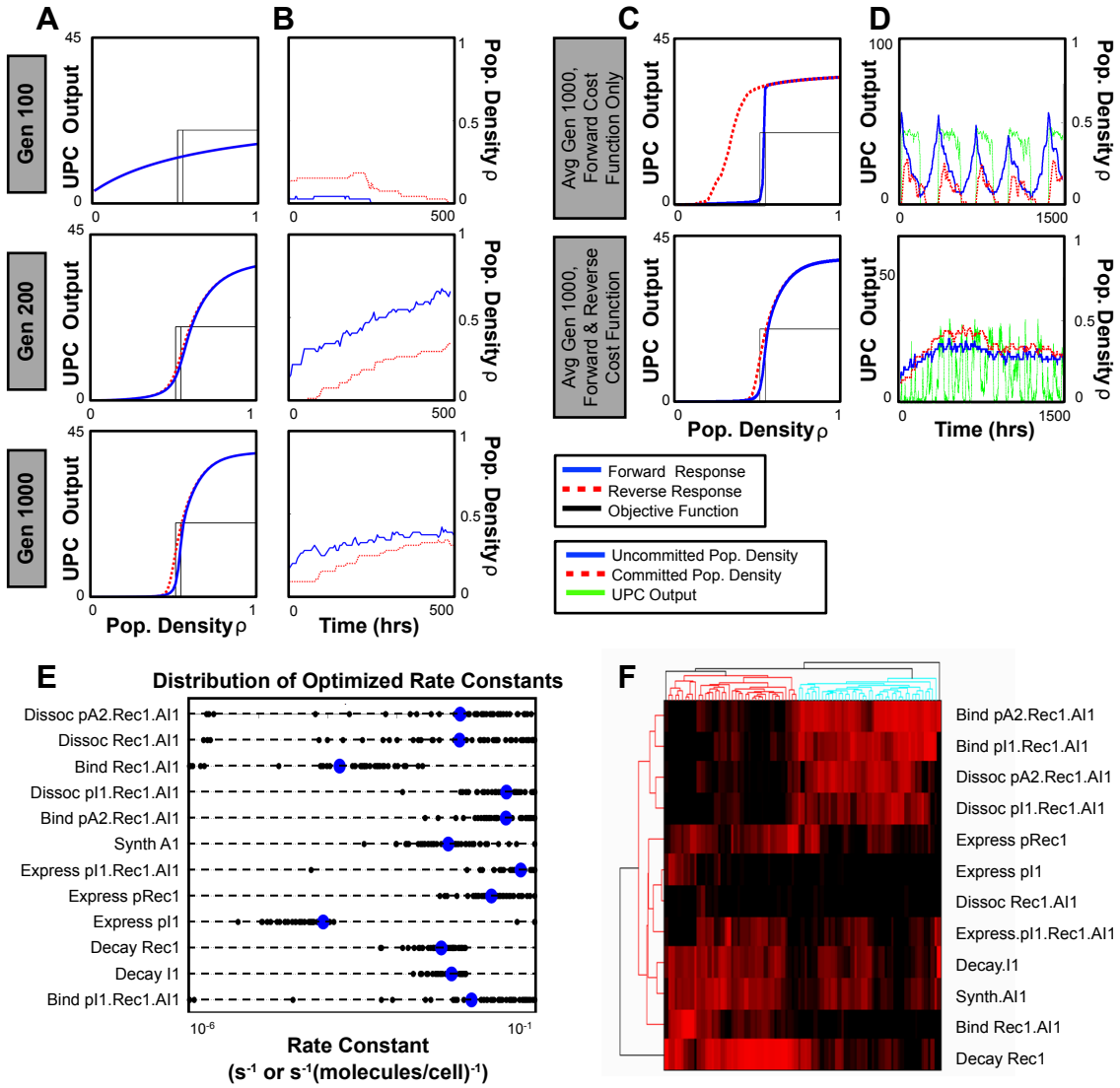

Figure 4.31: Parametric optimization of the "Uncommitted Population Control" subnetwork (Gillespie simulations). (A) GA optimization progress for three specific generations, using an ODE model of the UPC module. The objective function for the GA is a three-component step-function, with zero UPC activity below a defined threshold, an ignored transition region, and high activity above the transition region. (B) Behavior of the full system 3, implemented in the Gillespie framework, corresponding to optimization progress in A. (C) Average UPC module transfer curves when the reverse response is either excluded or included in the subnetwork GA optimization. (D) Full system behavior corresponding by row to the subnetwork optimization results in C. (E) Distribution of rate constants for the optimized parameter vectors determined by 75 independent GA runs of 1000 generations each, using both forward and reverse response objective functions. (F) Clustered sensitivity analysis of the UPC Module. Each column corresponds to a "parameter sensitivity signature" for each of the 75 local parameter neighborhoods that we sampled; rows correspond to the analyzed parameters of the UPC module. The first-order sensitivity values shown in the heat map range from 0.0 (black) to 0.5 (red).

the binding and dissociation rates of AI1-bound Rec1 complex (Rec1.AI1) have little influence. The opposite is true for the other cluster (cyan on the dendrogram in Fig. 4.31F).

4.6.3 Conclusion

In this work, we first use a model of the cell populations based on ODE. Even with this high abstraction level, useful conclusions could be drawn. Not surprisingly, feedbacks are a necessity to robustly maintain a constant population, but interestingly delays in the feedbacks should be avoided. To avoid population-wide oscillations, the key concept is to introduce a commitment process upstream of differentiation which feeds the population control module. Cells have to decide on their fate quickly and inform other cells as soon as their decision is irreversible. Moreover, the ODE analysis also teaches us that some constraints on the population dynamics should be fulfilled to ensure a asymptotically stable equilibrium, high cooperativity in the sensing module is an example of such constraints. The scope of the population model is limited and we then used a more detailed Langevin model that takes into account cells as individuals. The major difference to the ODE model is the implementation of the cells that in this case share identical synthetic genetic circuits but make individual cell-fate decision based on intercellular communication. The results of these simulations show that heterogenic phenotype in an isogenic population is a necessity to maintain homeostasis. Although intrinsic noise generates some heterogeneity, the implementation of an additional module to break asymmetry enhances performance and robustness. With the oscillator and the throttle, we propose two mechanisms that fulfill this task.

Another critical aspect for all biological systems is the fact that precise parameter values may be unknown and could vary *in vivo*. To be implemented, a synthetic circuit should show robustness to such fluctuations. Therefore having a glocal approach to know the region of the parameter space where the system is functional and what are the most sensitive parameters is critical to build a genetic network and optimize it. We use the RS-HDMR algorithm to obtain first order, and second order correlations at the three levels of optimization. First, we took advantage of the modular construction to find the best relative dynamics of the modules: in general, the critical component is the speed of intercellular communication. We also notice that slow toggle switching is beneficial for the system 4 but not the others. In general, we notice that a fast dynamics coupled to the ultrasensitivity of the systems amplifies the effect

of the molecular noise: it converts small fluctuations into a persistent phenotypic switch. Second, we specifically analyzed the oscillator and the throttle modules. Here again, thanks to modularity, optimizing each module individually for its designed task benefits the whole system. The RS-HDMR algorithm proved to be a useful tool: compared to traditional correlations analyses, it uses polynomial functions to fit the statistical dependencies and also provides second order relations. Finally, we focus on the population control module at the highest level of detail. For this last analysis, we use a Gillespie simulation that accounts explicitly for the binding, transcription and translation events. We show that obtaining Hill functions for protein expression is not trivial and the results of our GA results give which biophysics parameter should be changed in order to achieve such response. To complete the RS-HDMR analysis, we use clustered sensitivity analysis to find out which parameters can strongly influence the objective functions and are therefore good manipulation targets.

The strength of our design method is to take advantage of the different levels of modeling knowing the limits of each model. The ODE simulation gave us insightful, but rough directions. The stochastic simulations using the Langevin approximation are a good compromise to optimize the systems at the module level. The Langevin model helps finding the input-output function of each module that optimizes the overall system performance. The biophysical rates that give such ideal input-output function can be obtained with an optimization using the Gillespie model. These three levels of abstraction are complementary and consistent. For example, all models show the same dependence on the killing rate of the beta-cells: a very low sensitivity to parameter changes when the ratio of birth over killing rate is high and a linear dependence for low ratios.

To summarize, we successfully adapted design principles from engineering to synthetic biology. We properly apprehended the biological constraints by using different levels of modeling. But more than an academic exercise, this work gives bases for large circuit design and building: glocal analysis completed by the RS-HDMR algorithm revealed concrete manipulation targets to optimize the system. On a more scientific level, this gives insight about how homeostasis works: for example, heterogeneity in the stem cell population is necessary to have a gradual differentiation. We also show that the patterning depends on the mechanism that generates this asymmetry. An internal mechanism (oscillator) creates small clusters of identical phenotypes as recently discovered natural differentiating cells [192]. Whereas an external mechanism (lateral inhibition) produces an alternation on small distance similar to what is found in pattern formation [193].

Chapter 5

Discussion and Conclusion

This chapter starts with a discussion of the two sampling algorithms which I used for my glocal analysis. In the second section, I will link the different results of my thesis and discuss the advantages of the glocal approach. In particular, I will summarize what the glocal approach taught us for each of the studied applications and what could not have been discovered with a standard method. I will then consider the connectivity of the viable parameter vectors and discuss how to interpret this neutral viable space. Later, I will discuss extensions and other possible applications of the glocal analysis. Finally, I will conclude my thesis with possible improvements.

5.1 Sampling Algorithms

The glocal analysis is based on a broad sampling of the parameter space in order to gather a large amount of viable parameter vectors. Random sampling over parameter intervals can be done when the possible parameter values are restricted to a small interval and the number of parameters is small, as in the synthetic circuit analysis. However, brute force approaches become limited for higher number of dimensions, or when a larger number of vectors are necessary (for example when evaluating the viable volume) [55, 151]. Moreover, if the viable space has a complex shape, a random sampling in a hyperbox defined by parameter intervals loosely fits the viable region resulting in a large fraction of tested parameter vectors being rejected.

In this thesis, I proposed two methods that significantly improve sampling efficiency. They are based on the restriction of the integration domain in the parameter space to the most probable viable region. Both methods permit to acquire a large number of uniformly distributed viable parameter points, in comparison to previous methods based on uniform sampling in the entire parameter space [67, 150, 151]. This advantage is highly significant in high-dimensional parameter spaces.

The first method uses principal component analysis (PCA) to guide the sampling and obtain parameters more efficiently. This method requires very little adjustments, the only potential limitation being the initialization of the iterative procedure that requires a viable parameter vector. In the applications I presented (sections 4.1, 4.2 and 4.5), published data were used to define the initial conditions. However, even where such information is unavailable, random sampling and optimization techniques [194] are available to find such a vector. The drawback of this method is that efficiency decreases when the viable region differs strongly from an ellipsoidal shape as in the case of non-convex or poorly connected spaces for instance.

The second method involves two-stages: a coarse grained identification of viable regions followed by various applications of the PCA method. The initial exploration of the viable space allows identification of the regions where viable parameter vectors are found. It therefore overcomes the limitations of the PCA method as it can characterize viable regions that may be non-convex and poorly connected. Potential limitations of this approach include the choice of values for the algorithm parameters, e.g. maximum frequency of sampled viable points, bounds for the frequency of accepted iterations, and scaling factors for ellipsoid expansions. This approach also requires that the cost function maps continuously and injectively to the viability criteria. Such a cost function may not be trivial to find if multiple, unrelated interval criteria should be fulfilled.

I should add two comments that concern both methods. First, the borders of the sampling range should be restricted to realistic values. In some models, specific parameters may not be constrained, as only a combination or their ratio affects the systemic properties. This could result in parameters that cannot be individually identified which may bias the robustness measure. Having an *a priori* range for the sampling avoids such problems. The borders of the sampling range may be dictated by biophysical constraints, but also by the validity of the model. Note that a conservative choice of this range may only slow down marginally the efficiency of the algorithm as the iterative procedure quickly directs the sampling to viable regions.

The second comment refers to the number of dimensions. If both methods can cope much better with high dimensionality than brute force sampling, they are still limited by this factor. First, for a higher number of parameters, the complexity of the viable space can increase: a high-dimensional space allows more poorly connected viable regions to appear. Such regions can exponentially increase the minimum number of iterations needed for a complete sampling. Moreover, for a representative sampling, the number of viable vectors should also significantly increase with the dimension of the system.

Finally, I would like to emphasize that the interval approach used for the sampling algorithms, may better suit biological systems than a single valued approach. Whether one reports biological data or kinetic constants for biochemical reactions, the error intervals remain significant. This structural or practical unidentifiability of some parameters has been observed in many biochemical models [154, 155, 57]. For the potentially large class of models with this property, model parameters that yield an observed behavior cannot be uniquely identified even in the presence of arbitrarily abundant and precise data. In addition, even in the presence of error-free data, biological rates may fluctuate *in vivo* depending on the environmental conditions. Therefore, any unique parameter vector, even if exact, is representative of a unique condition. Due to these reasons, an approach using intervals for both the systemic properties and the viable parameter values, as in my glocal analysis, seems more correct to characterize biological systems.

5.2 Results Obtained With the Glocal Approach

In this section, I will review the different results obtained in this thesis emphasizing the aspects discovered with the glocal analysis. Most of the published work on the robustness of cellular circuits addresses either global or local robustness [51, 48, 49, 62]. My glocal approach overcomes the limitations of both global and local analyses. First, by generating large samples of parameter vectors, the method can estimate a viable volume of the parameter space that yields the correct values for the systemic properties. Second, by having a local measure for each parameter vector, the robustness evaluation is less easily misled by results derived from a particular chosen point in parameter space. This contrasts with parameter fitting that yields only single point estimate of the robustness.

The first advantage of the glocal analysis is that the analysis of parameter vectors, spanning over multiple orders of magnitude, shows how local robust-

ness varies in parameter space. The second advantage is that the combination of local and global analyses lends itself to a deeper mechanistic understanding of the circuit behavior. In particular, it can lead to the identification of key parameters important for robustness. Obvious applications include synthetic biology, where tunability of a synthetic circuit's robustness by changing key parameters is highly desirable. Third, the design of new experiments in order to discriminate between possible parameters or models can use the results of the robustness analysis and test specific perturbations with a high discrimination power. Fourth, the glocal method can also help deduce the differences between alternative architectures, independently of the parameters values. Finally, by studying different quantifiers of local robustness, one can obtain trade-offs between robustness and other system properties or network structures.

5.2.1 Glocal Robustness Analysis for Model Comparison

In the analysis and comparison of the two models of the cyanobacterial circadian clock (section 4.1), I first characterized the viable region of the parameter space for both models. The autocatalytic model shows a lack of robustness due to a strong correlation between two reaction rates, whereas the parameters of the two-sites model can vary independently in a broader region (figure 4.2). The most interesting results come from the distribution of the local robustness quantifiers. For both models I identified one critical rate that correlates with the robustness to molecular noise (figure 4.4C, D). Interestingly, in both models, the particular reaction related to this rate influences the feedback component showing that the feedback amplifies stochastic fluctuations. I also found that, in the autocatalytic model, the dephosphorylation rate correlates with the robustness to concentration perturbations (figure 4.4B). Glocal predications of this kind can guide new experiment to discriminate parameter values. Finally, the glocal approach showed that the two-sites model has greater local robustness over a larger range of parameters (figure 4.6G). Such a conclusion may not be obvious when focusing on the published parameter vector: in fact, the local robustness of the parameter vector published for the autocatalytic model is higher than the median robustness of all viable parameter vectors (figure 4.3F). This example shows that the glocal analysis is necessary to uncover such biased robustness results.

5.2.2 Glocal Analysis for Network Architecture Comparison

Twice in this thesis, I used the glocal approach to compare different network architectures. In the work with the *Drosophila* circadian clock (section 4.2), I compared two models with either one or two negative feedback loops. The global sampling revealed that the additional loop increases the robustness of the system, especially when the external entrainment is released. The use of the phase response curve as a local quantifier emphasizes the robustness advantage of multiple feedback loops. I also found that the rates of the reactions affecting the concentration of the nuclear complex discriminate between the different types of PRCs (Figure 4.10). This could explain why recent models of the *Drosophila* clock [107] integrate more complex control mechanisms of the nuclear components. In the second architecture comparison using the mitotic cycle models (section 4.5), we found that the robustness to molecular noise is higher for the model with an additional positive feedback loop. Here, the glocal analysis completes the analytical approach by showing that the robustness improvement is parameter-independent. To summarize, glocal results for architecture comparison imply that robustness differences are a consequence of the topology and not an artifact of a specific parameter vector.

5.2.3 Glocal Approach for Evolutionary Analyses

To study evolution of system architectures by addition of feedback loops, I used a generic model of the mitotic cycle. As the oscillations can occur with either a positive feedback or a negative feedback loop, a natural question to ask is why biological systems exhibit multiple feedback loops. Here, I tried to first answer that evolution is possible without disturbing the oscillations. First, in section 4.3, I proposed a new algorithm that mimics biological evolution with multiple small steps in the parameter space. With this method, I showed that the addition of a feedback loop is possible while maintaining some systemic properties. In the application of the two-stage sampling method (section 4.4), we also found that the system is more robust when the negative feedback is the main mechanism inducing oscillations, but robustness to parameter changes is even higher when a positive feedback loop is added. In these analyses the local aspect is not as present as in the applications discussed previously, but it may be used in the future to study the difference of local robustness through this evolutionary mechanism.

5.2.4 Robustness Analysis for Synthetic Circuits

To show the broad scope of the glocal method, I addressed the design of a synthetic biological circuit. The goal of the circuit is to maintain two populations of stem and β-cells knowing that stem cells divide and can also differentiate to β-cells which die at a constant rate. The first stage of the design is based on modularity. We started with a basic system to which new modules were added to increase performance and robustness. We obtained two alternative circuits that can maintain homeostasis under a large range of parameters and found the best parameter values for each system. For the optimization, the glocal analysis provides correlations between performance and parameter values that give directions for improvement. But, in order to be viable, such an artificial circuit should be on a par with the robustness of natural biological circuits, therefore the system should be optimized toward both performance and robustness. Therefore, the glocal analysis is important to choose not only the best parameter vector, but the one at the center of the region of performance in order to be less sensitive to small fluctuations. In conclusion, this work, which includes a glocal analysis of a synthetic circuit, is a proof of concept for synthetic circuit design.

5.3 Connectivity of the Viable Parameter Space

In my different works, I pointed out that the viable parameter space forms a connected set. If no rigorous proof of this affirmation can be made, it is reasonable to conjecture that the dense sampling and the continuity of the systemic properties mean that any region of the viable parameter region can be connected to another without leaving the viable region. The viable space formed by these parameter vectors can be considered as a 'neutral volume' [45] in which the systemic properties of the system are preserved. This observation is significant to understand how systems could evolve [57, 175, 102], in particular through gradual, small changes of individual parameters.

As shown in section 4.3, it is possible to evolve in the parameter space continuously while maintaining some macroscopic properties. Thus, robust circuits are accessible to natural selection through the connectedness of the neutral volume, without the need to change the system behavior itself. In this sense, evolution could favor robust circuits over the other: in the most robust regions, the features will be fulfilled for a broader range of perturbations.

It leads to the question of the relation between functionality and robustness: for a system to fulfill its features, such as period and amplitude for oscillatory systems [48, 151], many parameter vectors or even architectures are equivalent. But even with equivalent features, adaptation or robustness can differ. Recently, others have shown that frequency tunability is better for multi-loop oscillators[117]. Here, I have shown that an additional feedback loop in a system based on a negative feedback increases the robustness to parameter changes (section 4.4) or molecular noise (section 4.5). In relation to that, the natural question that arises is *"can the robustness to molecular noise (or any other local robustness property) be a driving force for the observed tendency of evolution to favor additional feedback loops?"*. It is also important to stress, considering the results of section 4.4, that an oscillatory system with a unique feedback loop can only evolve toward a system with multiple feedback loops without losing its cyclic behavior, favoring large networks.

5.4 Potential Applications for the Glocal Analysis

The different results in this thesis showed that my method is more than a theoretical concept: glocal robustness analysis provides new insights on the studied systems. I will now discuss extensions and other potential applications of the glocal analysis.

A first application is discrimination of models for a system with a well-defined robustness, such as the circadian clocks. This could be also used for systems with low robustness [50] where sensitivity to a certain type of perturbations is expected. In many systems, the experimental data cannot fully constrain model parameters and the inclusion of robustness could help restrict the parameter space. The glocal analysis results in correlations between parameter values and local robustness of the models. This information can be used in two ways: either to discriminate some regions of the parameter space if the system robustness is known, or to design new experiments to test the specific robustness and choose between concurrent parameter vectors.

The second application is related to architecture comparison with respect to robustness. As already applied in this thesis, the robustness of a specific architecture independently of its parameters can be assessed with glocal analysis. With this approach, even in the absence of precise parameter values, conclusions can be drawn. Indeed my glocal method has already been applied by others [159] to compare alternative topologies. Large scale parameter

sampling and the glocal robustness analysis could be integrated in ensemble modeling approaches; a method that has already been successfully used in different contexts [195, 71, 196].

Linking parameter space and network topologies can be done as discussed above with a unique system where certain reactions are inactive (very low parameter values) to reflect the different topologies. It results in a high-dimensional parameter space where a neutral space can be found. A connected path in this viable space can be interpreted as an evolutionary path where the systemic properties are conserved. Walks in the parameter space obtained with the evolutionary algorithm described in section 4.3.1 could be coupled to a glocal analysis to form a Metropolis-like algorithm [153]. This approach results in random walks in the parameter space biased toward regions with higher local robustness and may reveal why certain architectures are preferred to others [151, 2].

All these considerations can be reverted and applied to Synthetic Biology with a forward engineering approach. The field of Synthetic Biology, while producing new modules, is currently under pressure to build large functional circuits [120]. Along with feature and performance, modularity and robustness could be the key elements to design complex systems. On the one hand, modularity allows multiple combinations and extends system possibilities while keeping a small pool of elementary elements. On the other hand, these modules should be independent, yet compatible. It means that the modules should be able to function in a wide range of conditions (robust to parameter variations) and their input-output response should remain unaffected by other modules (robust to external perturbations). When applied to a module, the glocal approach can provide the functional parameter region and also suggest how robust the module is depending on the choice of parameters.

Another practical application of the glocal analysis is to understand and take advantage of drug cross-talk. With a proper knowledge of the system and a good model, the effects of the drugs can be hypothesized: drugs or any external interaction is reflected by a variation of some parameter in the model. On the one hand, this information could be used to understand why drugs have different side effects depending on the stress context [197]. For instance, the external stress, by changing some model parameters, could cause an unfortunate alteration of system robustness and the side effects of the drugs may be visible only in this case. On the other hand, some drugs, which may have little therapeutic effects, could decrease the robustness of the system and therefore open the possibility for another drug to act more efficiently. Such

dual-drug therapies are promising for some resistant cancer types [198]. In these two cases, a robustness analysis on a glocal scale could provide a better understanding of the drug effects and trigger new therapies.

5.5 Follow-up Work and Improvements

The first possible improvements concern the sampling algorithms. Both methods need some adjustments in order to be more robust and applicable to any sampling problem without tuning. As discussed, the PCA method is limited to ellipsoidal viable parameter space. To circumvent this problem the two-stage method, which can adapt to more complex shapes, was proposed, but its efficiency depends on the number of cluster and sampling iterations. To ensure a conservative coverage of the viable regions, the first stage of the method should span over larger ranges and therefore use more computational time. A way to circumvent this issue is to implement the algorithm on highly-parallelized graphic processor units with languages as CUDA for example [199]. Other variations of the Monte Carlo sampling used in statistical physics could be used to expand the initial, rough, parameter search. A promising option is Exchange Monte Carlo [200] where multiple parallel random walks explore the parameter space. Finally, another avenue is to restrict the exploration to the border of the viable region as done for bifurcation analysis [60], but this does not solve the dimensionality problem or the problems caused by unconnected regions.

Other improvements concern the formalism of the glocal approach. First, the specification of the viability criteria could use better semantics as described in linear temporal logic [146, 147]. Such formalism may also help to define a cost function necessary for the two-stage sampling algorithm. Second, the correlation analysis in the parameter space could be improved. In this sense, the use of the RS-HDMR algorithm [149, 201] provides higher order correlations and also uses nonlinear functions. Other analyses based on the geometry and the spatial distribution of the local quantifier values [202] could be integrated.

Concerning the algorithm for the evolutionary study, as discussed above, the random walk could be merged with a local quantifier to obtain a Metropolis-like evolution algorithm [156, 153]. Here also, the Exchange Monte-Carlo could help to explore the high-dimensional parameter space including different network topologies. These directions will provide a new tool for the study of evolution of biochemical network where the performance or the robustness could

be the driving force. Evolutionary algorithm may also be an alternate method for the design of circuits in synthetic biology where a specific behavior would be the driving force.

5.6 Conclusion

To characterize the behavior and robustness of cellular circuits is a major challenge for Systems Biology. The interest in this question is emphasized as robustness quantification can be used in different context. It can help testing models validity (natural systems being usually robust), give explanations for system architectures (different topologies show different robustness), or investigate system malfunctions (where robustness is abnormally weak or high). The applications of robustness also extend to synthetic biology or drug design. Yet, this field is still in its infancy: different robustness approaches can be found in the literature and a unified measure of robustness is still missing.

This thesis contributes to the advancement of the research in this field with a novel approach for robustness quantification based on a glocal analysis. It combines different methods to obtain a more objective measure. Throughout the different examples ranging from model discrimination to network comparison and circuit design, I showed that the glocal analysis is a powerful tool to quantify robustness. The application of the method is facilitated by the efficient sampling algorithms developed in this thesis. In conclusion, the glocal robustness analysis opens new possibilities in Systems Biology and the potential applications for this method are broad and numerous.

Bibliography

[1] Klipp E, Herwig R, Kowald A, Wierling C, Lehrach H (2005) Systems Biology in Practice: Concepts, Implementation and Application. Wiley-VCH, 1st edition. URL http://www.worldcat.org/isbn/3527310789.

[2] Alon U (2006) An Introduction to Systems Biology: Design Principles of Biological Circuits (Chapman & Hall/CRC Mathematical & Computational Biology). Chapman and Hall/CRC, 1st edition. URL http://www.worldcat.org/isbn/1584886420.

[3] Szallasi Z, Stelling J, Periwal V, editors (2006) System Modeling in Cellular Biology: From Concepts to Nuts and Bolts. The MIT Press, 1 edition. URL http://www.worldcat.org/isbn/0262195488.

[4] Murray JD (2007) Mathematical Biology: I. An Introduction (Interdisciplinary Applied Mathematics) (Pt. 1). Interdisciplinary applied mathematics. Springer, 3rd edition. URL http://www.worldcat.org/isbn/0387952233.

[5] Goldenfeld N, Kadanoff LP (1999) Simple lessons from complexity. Science 284: 87–89.

[6] Hairer E, Nørsett SP, Wanner G (2009) Solving Ordinary Differential Equations I: Nonstiff Problems (Springer Series in Computational Mathematics) (v. 1). Springer, 2nd edition. URL http://www.worldcat.org/isbn/3540566708.

[7] Laidler KJ (1987) Chemical Kinetics. Prentice Hall, 3rd edition. URL http://www.worldcat.org/isbn/0060438622.

[8] Wilkinson DJ (2006) Stochastic Modelling for Systems Biology (Chapman & Hall/CRC Mathematical & Computational Biology). Chapman and Hall/CRC, 1 edition. URL http://www.worldcat.org/isbn/1584885408.

[9] McQuarrie DA (1967) Stochastic approach to chemical kinetics. Journal of Applied Probability 4: 413-478.

[10] Gillespie DT (2007) Stochastic simulation of chemical kinetics. Annual review of physical chemistry 58: 35–55.

[11] Heinrich R, Schuster S (1996) The Regulation of Cellular Systems. New York, NY: Chapman & Hall.

[12] Arkin A, Ross J, McAdams HH (1998) Stochastic kinetic analysis of developmental pathway bifurcation in phage l-infected escherichia coli cells. Genetics 149: 1633-1648.

[13] Elowitz MB, Levine AJ, Siggia ED, Swain PS (2002) Stochastic gene expression in a single cell. Science 297: 1183–1186.

[14] Rao CV, Wolf DM, Arkin AP (2002) Control, exploitation and tolerance of intracellular noise. Nature 420: 231-237.

[15] Gillespie DT (1977) Exact stochastic simulation of coupled chemical reactions. The Journal of Physical Chemistry 81: 2340–2361.

[16] Van Kampen NG (2007) Stochastic Processes in Physics and Chemistry, Third Edition (North-Holland Personal Library). North Holland, 3rd edition. URL `http://www.worldcat.org/isbn/0444529659`.

[17] Schuss Z (2009) Theory and Applications of Stochastic Processes: An Analytical Approach (Applied Mathematical Sciences). Springer, 1st edition. URL `http://www.worldcat.org/isbn/1441916040`.

[18] Munsky B, Khammash M (2006) The finite state projection algorithm for the solution of the chemical master equation. Journal of Chemical Physics 124: 044104.

[19] Henzinger TA, Mateescu M, Wolf V (2009) Computer aided verification. In: Bouajjani A, Maler O, editors, Lecture Notes in Computer Science 2009, Springer Berlin / Heidelberg, chapter Sliding Window Abstraction for Infinite Markov Chains. pp. 337-352.

[20] Gillespie D (1976) A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. Journal of Computational Physics 22: 403–434.

[21] Gibson MA, Jehoshua B (2000) Efficient exact stochastic simulation of chemical systems with many species and many channels. J Phys Chem 104: 1876-1889.

[22] Cao Y, Hong L, Petzold LR (2004) Efficient formulation of the stochastic simulation algorithm for chemically reacting systems. J Chem Phys 121: 4059-4067.

[23] McCollum JM, Peterson GD, Cox CD, Simpson ML, Samatova NF (2006) The sorting direct method for stochastic simulation of biochemical systems with varying reaction execution behavior. Comp Biol Chem 30: 39-49.

[24] Matsumoto M, Nishimura T (1998) Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. ACM TOMACS 8: 3-30.

[25] Gillespie DT (2001) Approximate accelerated stochastic simulation of chemically reacting systems. J Chem Phys 115: 1716.

[26] Cao Y, Gillespie DT, Petzold LR (2006) Efficient step size selection for tau-leaping simulation method. J Chem Phys 124: 044109.

[27] Rathinam M, Petzold LR, Cao Y, Gillespie DT (2003) Stiffness in stochastic chemically reacting systems: the implicit tau-leaping method. J Chem Phys 119: 12784-12794.

[28] Tian T, Burrage K (2004) Binomial leap methods for simulating stochastic chemical kinetics. J Chem Phys 121: 10356-10364.

[29] Cao Y, Gillespie DT, Petzold LR (2005) Avoiding negative populations in explicit poisson tau-leaping. J Chem Phys 123: 054104.

[30] Haseltine EL, Rawlings JB (2002) Approximate simulation of coupled fast and slow reactions for stochastic chemical kinetics. J Chem Phys 117: 6959-6969.

[31] Rao CV, Arkin AP (2003) Stochastic chemical kinetics and the quasi-steady-state assumption: Application to the gillespie algorithm. J Chem Phys 118: 4999-5010.

[32] Cao Y, Gillespie D, Petzold L (2005) Multiscale stochastic simulation algorithm with stochastic partial equilibirum assumption for chemically reacting systems. J Comp Phys 206: 395-411.

[33] Cao Y, Gillespie D, Petzold L (2005) The slow-scale stochastic simulation algorithm. J Chem Phys 122: 014116.

[34] Cao Y, Gillespie D, Petzold L (2005) Accelerated stochastic simulation of the stiff enzyme-substrate reaction. J Chem phys 123: 144917.

[35] Kloeden PE, Platen E (1992) Numerical Solution of Stochastic Differential Equations (Stochastic Modelling and Applied Probability). Springer, corrected edition. URL `http://www.worldcat.org/isbn/3540540628`.

[36] Kitano H (2007) Towards a theory of biological robustness. Molecular Systems Biology 3.

[37] Nowak MA, Boerlijst MC, Cooke J, Smith JM (1997) Evolution of genetic redundancy. Nature 388: 167–171.

[38] Wagner A (2000) Robustness against mutations in genetic networks of yeast. Nature genetics 24: 355–361.

[39] McAdams HH, Arkin A (1997) Stochastic mechanisms in gene-expression. Proceedings of the National Academy of Sciences of the United States of America 94: 814–819.

[40] Swain PS, Elowitz MB, Siggia ED (2002) Intrinsic and extrinsic contributions to stochasticity in gene expression. Proceedings of the National Academy of Sciences of the United States of America 99: 12795–12800.

[41] Gonze D, Halloy J, Goldbeter A (2002) Robustness of circadian rhythms with respect to molecular noise. Proceedings of the National Academy of Sciences of the United States of America 99: 673–678.

[42] Gonze D, Goldbeter A (2006) Circadian rhythms and molecular noise. Chaos: An Interdisciplinary Journal of Nonlinear Science 16: 026110+.

[43] Ruoff P (1992) Introducing temperature-compensation in any reaction kinetic oscillator model. Journal of Interdisiplinary Cycle Research 23: 92–99.

[44] Cornish-Bowden A (2004) Fundamentals of Enzyme Kinetics. PORTLAND PRESS, 3rd edition. URL `http://www.worldcat.org/isbn/1855781581`.

[45] Wagner A (2007) Robustness and Evolvability in Living Systems: (Princeton Studies in Complexity). Princeton University Press, 1st edition. URL `http://www.worldcat.org/isbn/0691134049`.

[46] Morohashi M, Winn AE, Borisuk MT, Bolouri H, Doyle J, et al. (2002) Robustness as a measure of plausibility in models of biochemical networks. Journal of theoretical biology 216: 19–30.

[47] Stelling J, Sauer U, Szallasi Z, Doyle FJ, Doyle J (2004) Robustness of cellular functions. Cell 118: 675–685.

[48] Stelling J, Gilles ED, Doyle FJ (2004) Robustness properties of circadian clock architectures. Proceedings of the National Academy of Sciences of the United States of America 101: 13210–13215.

[49] Rand D, Shulgin B, Salazar J, Millar A (2006) Uncovering the design principles of circadian clocks: Mathematical analysis of flexibility and evolutionary goals. Journal of Theoretical Biology 238: 616–635.

[50] Wolf J, Becker-Weimann S, Heinrich R (2005) Analysing the robustness of cellular rhythms. Systems biology 2: 35–41.

[51] El-Samad H, Kurata H, Doyle JC, Gross CA, Khammash M (2005) Surviving heat shock: Control strategies for robustness and performance. Proceedings of the National Academy of Sciences of the United States of America 102: 2736–2741.

[52] Doyle J, Csete M (2005) Motifs, control, and stability. PLoS Biol 3: e392+.

[53] Carlson JM, Doyle J (2002) Complexity and robustness. Proceedings of the National Academy of Sciences of the United States of America 99: 2538–2545.

[54] Newman MEJ, Girvan M, Farmer JD (2002) Optimal design, robustness, and risk aversion. Physical Review Letters 89: 028301+.

[55] Eissing T, Allgöwer F, Bullinger E (2005) Robustness properties of apoptosis models with respect to parameter variations and intrinsic noise. IEE Proceedings - Systems Biology 152: 221–228.

[56] Dayarian A, Chaves M, Sontag ED, Sengupta AM (2009) Shape, size, and robustness: feasible regions in the parameter space of biochemical networks. PLoS computational biology 5: e1000256+.

[57] Daniels BC, Chen YJJ, Sethna JP, Gutenkunst RN, Myers CR (2008) Sloppiness, robustness, and evolvability in systems biology. Current opinion in biotechnology 19: 389–395.

[58] Ma L, Iglesias P (2002) Quantifying robustness of biochemical network models. BMC Bioinformatics 3: 38+.

[59] Battogtokh D, Tyson JJ (2004) Bifurcation analysis of a model of the budding yeast cell cycle. Chaos: An Interdisciplinary Journal of Nonlinear Science 14: 653–661.

[60] Leloup JC, Goldbeter A (2004) Modeling the mammalian circadian clock: Sensitivity analysis and multiplicity of oscillatory mechanisms. Journal of Theoretical Biology 230: 541–562.

[61] Doyle F, Gunawan R, Bagheri N, Mirsky H, To T (2006) Circadian rhythm: A natural, robust, multi-scale control system. Computers & Chemical Engineering 30: 1700–1711.

[62] Bagheri N, Stelling J, Doyle FJ (2007) Quantitative performance metrics for robustness in circadian rhythms. Bioinformatics 23: 358–364.

[63] Kurata H, El-Samad H, Iwasaki R, Ohtake H, Doyle JC, et al. (2006) Module-based analysis of robustness tradeoffs in the heat shock response system. PLoS computational biology 2.

[64] Csete ME, Doyle JC (2002) Reverse engineering of biological complexity. Science 295: 1664–1669.

[65] Ruoff P, Zakhartsev M, Westerhoff HV (2007) Temperature compensation through systems biology. The FEBS journal 274: 940–950.

[66] Hong CI, Conrad ED, Tyson JJ (2007) A proposal for robust temperature compensation of circadian rhythms. Proceedings of the National Academy of Sciences of the United States of America 104: 1195–1200.

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

[68] El-Samad H, Khammash M (2006) Regulated degradation is a mechanism for suppressing stochastic fluctuations in gene regulatory networks. Biophysical journal 90: 3749–3761.

[69] Thattai M, van Oudenaarden A (2001) Intrinsic noise in gene regulatory networks. Proceedings of the National Academy of Sciences of the United States of America 98: 8614–8619.

[70] Alon U, Surette MG, Barkai N, Leibler S (1999) Robustness in bacterial chemotaxis. Nature 397: 168–171.

[71] Ma W, Trusina A, El-Samad H, Lim WA, Tang C (2009) Defining network topologies that can achieve biochemical adaptation. Cell 138: 760–773.

[72] Goldbeter A (1997) Biochemical Oscillations and Cellular Rhythms: The Molecular Bases of Periodic and Chaotic Behaviour. Cambridge University Press. URL `http://www.worldcat.org/isbn/0521599466`.

[73] Tyson JJ, Chen KC, Novak B (2003) Sniffers, buzzers, toggles and blinkers: dynamics of regulatory and signaling pathways in the cell. Current Opinion in Cell Biology 15: 221–231.

[74] Tyson JJ, Novak B (2001) Regulation of the eukaryotic cell cycle: Molecular antagonism, hysteresis, and irreversible transitions. Journal of Theoretical Biology 210: 249–263.

[75] Wijnen H, Young MW (2006) Interplay of circadian clocks and metabolic rhythms. Annual review of genetics 40: 409–448.

[76] Smolensky M, Peppas N (2007) Chronobiology, drug delivery, and chronotherapeutics. Advanced Drug Delivery Reviews 59: 828–851.

[77] Partensky F, Hess WR, Vaulot D (1999) Prochlorococcus, a marine photosynthetic prokaryote of global significance. Microbiology and molecular biology reviews 63: 106–127.

[78] Ditty JL, Williams SB, Golden SS (2003) A cynaobacterial circadian timing mechanism. Annual Review of Genetics 37: 513–543.

[79] Ouyang Y, Andersson CR, Kondo T, Golden SS, Johnson CH (1998) Resonating circadian clocks enhance fitness in cyanobacteria. Proceedings of the National Academy of Sciences of the United States of America 95: 8660–8664.

[80] Ishiura M, Kutsuna S, Aoki S, Iwasaki H, Andersson CR, et al. (1998) Expression of a gene cluster kaiabc as a circadian feedback process in cyanobacteria. Science 281: 1519–1523.

[81] Nakajima M, Imai K, Ito H, Nishiwaki T, Murayama Y, et al. (2005) Reconstitution of circadian oscillation of cyanobacterial KaiC phosphorylation in vitro. Science 308: 414–415.

[82] Pattanayek R, Williams DR, Pattanayek S, Xu Y, Mori T, et al. (2006) Analysis of KaiA-KaiC protein interactions in the cyano-bacterial circadian clock using hybrid structural methods. The EMBO journal 25: 2017–2028.

[83] Pattanayek R, Williams DR, Pattanayek S, Mori T, Johnson CH, et al. (2008) Structural model of the circadian clock KaiB-KaiC complex and mechanism for modulation of KaiC phosphorylation. The EMBO journal 27: 1767–1778.

[84] Johnson CH, Egli M, Stewart PL (2008) Structural insights into a circadian oscillator. Science 322: 697–701.

[85] Murakami R, Miyake A, Iwase R, Hayashi F, Uzumaki T, et al. (2008) ATPase activity and its temperature compensation of the cyanobacterial clock protein KaiC. Genes to Cells 13: 387–395.

[86] Markson JS, O'Shea EK (2009) The molecular clockwork of a protein-based circadian oscillator. FEBS letters 583: 3938–3947.

[87] Mehra A, Hong CI, Shi M, Loros JJ, Dunlap JC, et al. (2006) Circadian rhythmicity by autocatalysis. PLoS computational biology 2.

[88] van Zon JS, Lubensky DK, Altena PRH, ten Wolde PR (2007) An allosteric model of circadian KaiC phosphorylation. Proceedings of the National Academy of Sciences 104: 7420–7425.

[89] Clodong S, Duhring U, Kronk L, Wilde A, Axmann I, et al. (2007) Functioning and robustness of a bacterial circadian clock. Molecular Systems Biology 3.

[90] Rust MJ, Markson JS, Lane WS, Fisher DS, O'Shea EK (2007) Ordered phosphorylation governs oscillation of a three-protein circadian clock. Science (New York, NY) 318: 809–812.

[91] Nishiwaki T, Satomi Y, Kitayama Y, Terauchi K, Kiyohara R, et al. (2007) A sequential program of dual phosphorylation of KaiC as a basis for circadian rhythm in cyanobacteria. The EMBO journal 26: 4029–4037.

[92] Mori T, Williams DR, Byrne MO, Qin X, Egli M, et al. (2007) Elucidating the ticking of an in vitro circadian clockwork. PLoS biology 5.

[93] Pattanayek R, Wang J, Mori T, Xu Y, Johnson CHH, et al. (2004) Visualizing a circadian clock protein: crystal structure of KaiC and functional insights. Molecular cell 15: 375–388.

[94] Mori T, Saveliev SV, Xu Y, Stafford WF, Cox MM, et al. (2002) Circadian clock protein KaiC forms ATP-dependent hexameric rings and binds DNA. Proceedings of the National Academy of Sciences of the United States of America 99: 17203–17208.

[95] Kageyama H, Nishiwaki T, Nakajima M, Iwasaki H, Oyama T, et al. (2006) Cyanobacterial circadian pacemaker: Kai protein complex dynamics in the KaiC phosphorylation cycle in vitro. Molecular cell 23: 161–171.

[96] Xu Y, Mori T, Pattanayek R, Pattanayek S, Egli M, et al. (2004) Identification of key phosphorylation sites in the circadian clock protein KaiC by crystallographic and mutagenetic analyses. Proceedings of the National Academy of Sciences of the United States of America 101: 13933–13938.

[97] Nishiwaki T, Satomi Y, Nakajima M, Lee C, Kiyohara R, et al. (2004) Role of KaiC phosphorylation in the circadian clock system of synechococcus elongatus PCC 7942. Proceedings of the National Academy of Sciences of the United States of America 101: 13927–13932.

[98] Pattanayek R, Mori T, Xu Y, Pattanayek S, Johnson CH, et al. (2009) Structures of KaiC circadian clock mutant proteins: a new phosphorylation site at T426 and mechanisms of kinase, ATPase and phosphatase. PloS one 4: e7529+.

[99] Emberly E, Wingreen NS (2006) Hourglass model for a protein-based circadian oscillator. Physical review letters 96.

[100] Yoda M, Eguchi K, Terada TP, Sasai M (2007) Monomer-shuffling and allosteric transition in KaiC circadian oscillation. PloS one 2: e408+.

[101] Eguchi K, Yoda M, Terada TP, Sasai M (2008) Mechanism of robust circadian oscillation of KaiC phosphorylation in vitro. Biophysical journal 95: 1773–1784.

[102] Hastings MH (2000) Circadian clockwork: two loops are better than one. Nature Reviews Neuroscience 1: 143–146.

[103] Bell-Pedersen D, Cassone VM, Earnest DJ, Golden SS, Hardin PE, et al. (2005) Circadian rhythms from multiple oscillators: lessons from diverse organisms. Nature reviews Genetics 6: 544–556.

[104] Goldbeter A (1995) A model for circadian oscillations in the drosophila period protein (PER). Proceedings: Biological Sciences 261.

[105] Leloup JC, Goldbeter A (1998) A model for circadian rhythms in drosophila incorporating the formation of a complex between the PER and TIM proteins. Journal of Biological Rhythms 13: 70–87.

[106] Hardin P (2006) Essential and expendable features of the circadian timekeeping mechanism. Current Opinion in Neurobiology 16: 686–692.

[107] Bagheri N, Lawson MJ, Stelling JA, Doyle FJ (2008) Modeling the drosophila melanogaster circadian oscillator via phase optimization. Journal of Biological Rhythms 23: 525–537.

[108] Dunlap JC (2006) Proteins in the neurospora circadian clockworks. The Journal of biological chemistry 281: 28489–28493.

[109] Más P (2005) Circadian clock signaling in arabidopsis thaliana: from gene expression to physiology and development. The International journal of developmental biology 49: 491–500.

[110] Beckerweimann S, Wolf J, Herzel H, Kramer A (2004) Modeling feedback loops of the mammalian circadian oscillator. Biophysical Journal 87: 3023–3034.

[111] Novák B, Tyson JJ (2008) Design principles of biochemical oscillators. Nature reviews Molecular cell biology 9: 981–991.

[112] Goodwin BC (1965) Oscillatory behavior in enzymatic control processes. Advances in enzyme regulation 3: 425–438.

[113] Tyson JJ, Csikasz-Nagy A, Novak B (2002) The dynamics of cell cycle regulation. Bioessays 24: 1095–1109.

[114] Pomerening JR, Sontag ED, Ferrell JE (2003) Building a cell cycle oscillator: hysteresis and bistability in the activation of Cdc2. Nature Cell Biology 5: 346–351.

[115] Lahav G, Rosenfeld N, Sigal A, Geva-Zatorsky N, Levine AJ, et al. (2004) Dynamics of the p53-Mdm2 feedback loop in individual cells. Nature Genetics 36: 147–150.

[116] Lewis J, Hanisch A, Holder M (2009) Notch signaling, the segmentation clock, and the patterning of vertebrate somites. Journal of Biology 8: 44+.

[117] Tsai TY, Choi YS, Ma W, Pomerening JR, Tang C, et al. (2008) Robust, tunable biological oscillations from interlinked positive and negative feedback loops. Science 321: 126–129.

[118] Elowitz MB, Leibler S (2000) A synthetic oscillatory network of transcriptional regulators. Nature 403: 335–338.

[119] Gardner TS, Cantor CR, Collins JJ (2000) Construction of a genetic toggle switch in Escherichia coli. Nature 403: 339–342.

[120] Purnick PEM, Weiss R (2009) The second wave of synthetic biology: from modules to systems. Nature Reviews Molecular Cell Biology 10: 410–422.

[121] Swinburne IA, Miguez DG, Landgraf D, Silver PA (2008) Intron length increases oscillatory periods of gene expression in animal cells. Genes & Development 22: 2342–2346.

[122] Tigges M, Marquez-Lago TT, Stelling J, Fussenegger M (2009) A tunable synthetic mammalian oscillator. Nature 457: 309–312.

[123] Stricker J, Cookson S, Bennett MR, Mather WH, Tsimring LS, et al. (2008) A fast, robust and tunable synthetic gene oscillator. Nature 456: 516–519.

[124] Kramer BP, Viretta AU, Baba MD, Aubel D, Weber W, et al. (2004) An engineered epigenetic transgene switch in mammalian cells. Nature Biotechnology 22: 867–870.

[125] Weiss R, Knight T (2001) Engineered communications for microbial robotics. In: Condon A, Rozenberg G, editors, DNA Computing, Berlin, Heidelberg: Springer Berlin / Heidelberg, volume 2054 of *Lecture Notes in Computer Science*, chapter 1. pp. 1-16–16. doi:10.1007/3-540-44992-2_1. URL `http://dx.doi.org/10.1007/3-540-44992-2_1`.

[126] Weber W, Schuetz M, Dénervaud N, Fussenegger M (2009) A synthetic metabolite-based mammalian inter-cell signaling system. Molecular bioSystems 5: 757–763.

[127] Chen MT, Weiss R (2005) Artificial cell-cell communication in yeast saccharomyces cerevisiae using signaling elements from arabidopsis thaliana. Nature Biotechnology : 1551–1555.

[128] You L, Cox RS, Weiss R, Arnold FH (2004) Programmed population control by cell-cell communication and regulated killing. Nature 428: 868–871.

[129] Weber W, Daoud-El Baba M, Fussenegger M (2007) Synthetic ecosystems based on airborne inter- and intrakingdom communication. Proceedings of the National Academy of Sciences of the United States of America 104: 10435–10440.

[130] Tabor JJ, Salis HM, Simpson ZBB, Chevalier AA, Levskaya A, et al. (2009) A synthetic genetic edge detection program. Cell 137: 1272–1281.

[131] Basu S, Mehreja R, Thiberge S, Chen MT, Weiss R (2004) Spatiotemporal control of gene expression with pulse-generating networks. Proceedings of the National Academy of Sciences of the United States of America 101: 6355–6360.

[132] Basu S, Gerchman Y, Collins CH, Arnold FH, Weiss R (2005) A synthetic multicellular system for programmed pattern formation. Nature 434: 1130–1134.

[133] Kobayashi H, Kærn M, Araki M, Chung K, Gardner TS, et al. (2004) Programmable cells: Interfacing natural and engineered gene networks. Proceedings of the National Academy of Sciences of the United States of America 101: 8414–8419.

[134] Hafner M, Koeppl H, Hasler M, Wagner A (2009) 'Glocal' robustness analysis and model discrimination for circadian oscillators. PLoS Computational Biology 5: e1000534+.

[135] Zamora Sillero E, Hafner M, Ibig A, Stelling J, Wagner A (2010) Efficient characterization of high-dimensional parameter spaces for systems biology. PLoS Computational Biology : submitted+.

[136] Hafner M, Sacré P, Symul L, Sepulchre R, Koeppl H (2010) Multiple feedback loops in circadian cycles: Robustness and entrainment as selection criteria. In: Nykter M, Ruusuvuori P, Carlberg C, Yli-Harja O, editors, Seventh International Workshop on Computational Systems Biology. pp. 43–46.

[137] Hafner M, Koeppl H, Wagner A (2009) Evolution of feedback loops in oscillatory systems. In: Third International Conference on Fundations of Systems Biology in Enginnering. pp. 157–160. URL http://arxiv.org/abs/1003.1231. 1003.1231.

[138] Gonze D, Hafner M (2010) Positive feedbacks contribute to the robustness of the cell cycle with respect to molecular noise. In: Lévine J, Müllhaupt P, editors, Advances in the Theory of Control, Signals and Systems, with Physical Modelling. Berlin: Springer, Lecture Notes in Control and Information Sciences Series, pp. in press+.

[139] Miller M, Hafner M, Sontag E, Subramanian S, Purnick P, et al. Design of a large scale synthetic biological circuit to maintain artificial tissue homeostasis. In preparation.

[140] Steuer R, Gross T, Selbig J, Blasius B (2006) Structural kinetic modeling of metabolic networks. Proceedings of the National Academy of Sciences 103: 11868–11873.

[141] Zhang Y, Rundell A (2006) Comparative study of parameter sensitivity analyses of the TCR-activated Erk-MAPK signalling pathway. Systems biology 153: 201–211.

[142] Barkai N, Leibler S (2000) Biological rhythms: Circadian clocks limited by noise. Nature 403: 267–268.

[143] Jaulin L, Kieffer M, Didrit O, Walter E (2001) Applied Interval Analysis. Springer, 1st edition. URL http://www.worldcat.org/isbn/1852332190.

[144] Kuipers B (1994) Qualitative Reasoning: Modeling and Simulation with Incomplete Knowledge (Artificial Intelligence). The MIT Press. URL http://www.worldcat.org/isbn/026211190X.

[145] Ackermann J (2002) Robust Control: The Parameter Space Approach (Communications and Control Engineering). Springer, 2nd edition. URL http://www.worldcat.org/isbn/1852335149.

[146] Fages F, Rizk A (2008) On temporal logic constraint solving for analyzing numerical data time series. Theoretical Computer Science 408: 55–65.

[147] Rizk A, Batt G, Fages F, Soliman S (2009) A general computational method for robustness analysis with applications to synthetic gene networks. Bioinformatics 25: 169–178.

[148] Fukunaga K (1990) Introduction to Statistical Pattern Recognition, Second Edition (Computer Science and Scientific Computing Series). Academic Press, 2nd edition. URL http://www.worldcat.org/isbn/0122698517.

[149] Feng X, Hooshangi S, Chen D, Li G, Weiss R, et al. (2004) Optimizing genetic circuits by global sensitivity analysis. Biophysical Journal 87: 2195–2202.

[150] Blüthgen N, Herzel H (2003) How robust are switches in intracellular signaling cascades? Journal of theoretical biology 225: 293–300.

[151] Wagner A (2005) Circuit topology and the evolution of robustness in two-gene circadian oscillators. Proceedings of the National Academy of Sciences of the United States of America 102: 11775–11780.

[152] Powell WB (2007) Approximate Dynamic Programming: Solving the Curses of Dimensionality (Wiley Series in Probability and Statistics). Wiley-Interscience, 1st edition. URL `http://www.worldcat.org/isbn/0470171553`.

[153] Press WH, Flannery BP, Teukolsky SA, Vetterling WT (1992) Numerical Recipes in C: The Art of Scientific Computing. Cambridge University Press, 2nd edition. URL `http://www.worldcat.org/isbn/0521431085`.

[154] Hengl S, Kreutz C, Timmer J, Maiwald T (2007) Data-based identifiability analysis of non-linear dynamical models. Bioinformatics 23: 2612–2618.

[155] Ljung L, Glad T (1994) On global identifiability for arbitrary model parametrizations. Automatica 30: 265–276.

[156] Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH, Teller E (1953) Equation of state calculation by fast computing machines. Journal of Chemical Physics 21: 1087–1092.

[157] Newman MEJ, Barkema GT (1999) Monte Carlo Methods in Statistical Physics. Oxford University Press, USA. URL `http://www.worldcat.org/isbn/0198517971`.

[158] Hoeffding W (1963) Probability inequalities for sums of bounded random variables. Journal of the American Statistical Association 58: 13–30.

[159] Zhang J, Yuan Z, Li HXX, Zhou T (2010) Architecture-dependent robustness and bistability in a class of genetic circuits. Biophysical journal 99: 1034–1042.

[160] Ruoff P, Mohsenzadeh S, Rensing L (1996) Circadian rhythms and protein turnover: the effect of temperature on the period lengths of clock mutants simulated by the goodwin oscillator. Die Naturwissenschaften 83: 514–517.

[161] Vilar JM, Kueh HYY, Barkai N, Leibler S (2002) Mechanisms of noise-resistance in genetic oscillators. Proc Natl Acad Sci U S A 99: 5988–5992.

[162] Pikovsky A, Maistrenko Y (2008) Synchronization: Theory and Application (NATO Science Series II: Mathematics, Physics and Chemistry). Springer, 1 edition. URL `http://www.worldcat.org/isbn/1402014171`.

[163] Rosenblum MG, Pikovsky AS, Kurths J (1996) Phase synchronization of chaotic oscillators. Physical Review Letters 76: 1804–1807.

[164] Khalil HK (2001) Nonlinear Systems (3rd Edition). Prentice Hall, 3rd edition. URL `http://www.worldcat.org/isbn/0130673897`.

[165] Mihalcescu I, Hsing W, Leibler S (2004) Resilient circadian oscillator revealed in individual cyanobacteria. Nature 430: 81–85.

[166] Johnson CH (2004) Global orchestration of gene expression by the biological clock of cyanobacteria. Genome biology 5.

[167] Citri A, Yarden Y (2006) EGF-ERBB signalling: towards the systems level. Nature Reviews Molecular Cell Biology 7: 505–516.

[168] Kitayama Y, Iwasaki H, Nishiwaki T, Kondo T (2003) KaiB functions as an attenuator of KaiC phosphorylation in the cyanobacterial circadian clock system. The EMBO journal 22: 2127–2134.

[169] Kitayama Y, Nishiwaki T, Terauchi K, Kondo T (2008) Dual kaic-based oscillations constitute the circadian system of cyanobacteria. Genes & development 22: 1513–1521.

[170] Winfree AT (2001) The Geometry of Biological Time. Springer, 2nd edition. URL `http://www.worldcat.org/isbn/0387989927`.

[171] Hardin PE (2005) The circadian timekeeping system of drosophila. Current biology 15.

[172] Ueda HR, Hagiwara M, Kitano H (2001) Robust oscillations within the interlocked feedback model of drosophila circadian rhythm. Journal of Theoretical Biology 210: 401–406.

[173] Sehgal A, Rothenfluh-Hilfiker A, Hunter-Ensor M, Chen Y, Myers MP, et al. (1995) Rhythmic expression of timeless: a basis for promoting circadian cycles in period gene autoregulation. Science (New York, NY) 270: 808–810.

[174] Hall JC, Rosbach M (1987) Genes and biological rhythms. Trends in Genetics 3: 185–191.

[175] François P, Hakim V (2004) Design of genetic networks with specified functions by evolution in silico. Proceedings of the National Academy of Sciences of the United States of America 101: 580–585.

[176] François P, Hakim V (2005) Core genetic module: The mixed feedback loop. Physical Review E 72.

[177] Fung E, Wong WW, Suen JK, Bulter T, Lee Sg, et al. (2005) A synthetic gene-metabolic oscillator. Nature 435: 118–122.

[178] Berg HC (1993) Random Walks in Biology. Princeton University Press, revised edition. URL `http://www.worldcat.org/isbn/0691000646`.

[179] Tyson JJ (1991) Modeling the cell division cycle: cdc2 and cyclin interactions. Proceedings of the National Academy of Sciences of the United States of America 88: 7328–7332.

[180] Goldbeter A (1991) A minimal cascade model for the mitotic oscillator involving cyclin and cdc2 kinase. Proceedings of the National Academy of Sciences of the United States of America 88: 9107–9111.

[181] Bornholdt S, Schuster HG, editors (2003) Handbook of Graphs and Networks: From the Genome to the Internet. Wiley-VCH. URL `http://www.worldcat.org/isbn/3527403361`.

[182] Hasty J, Dolnik M, Rottschäfer V, Collins JJ (2002) Synthetic gene network for entraining and amplifying cellular oscillations. Physical Review Letters 88.

[183] Goldbeter A (1993) Modeling the mitotic oscillator driving the cell division cycle. Comments on Theoretical Biology 3: 75–107.

[184] Kar S, Baumann WT, Paul MR, Tyson JJ (2009) Exploring the roles of noise in the eukaryotic cell cycle. Proceedings of the National Academy of Sciences of the United States of America 106: 6471–6476.

[185] Cattaneo E, McKay R (1990) Proliferation and differentiation of neuronal stem cells regulated by nerve growth factor. Nature 347: 762–765.

[186] Coleman ME, DeMayo F, Yin KC, Lee HM, Geske R, et al. (1995) Myogenic vector expression of insulin-like growth factor I stimulates muscle cell differentiation and myofiber hypertrophy in transgenic mice. Journal of Biological Chemistry 270: 12109–12116.

[187] Fujikawa T, Oh SH, Pi L, Hatch HM, Shupe T, et al. (2005) Teratoma formation leads to failure of treatment for type I diabetes using embryonic stem cell-derived insulin-producing cells. Am J Pathol 166: 1781–1791.

[188] Ryan EA, Paty BW, Senior PA, Bigam D, Alfadhli E, et al. (2005) Five-year follow-up after clinical islet transplantation. Diabetes 54: 2060–2069.

[189] León-Quinto T, Jones J, Skoudy A, Burcin M, Soria B (2004) In vitro directed differentiation of mouse embryonic stem cells into insulin-producing cells. Diabetologia 47: 1442–1451.

[190] Zhang D, Jiang W, Liu M, Sui X, Yin X, et al. (2009) Highly efficient differentiation of human ES cells and iPS cells into mature pancreatic insulin-producing cells. Cell research 19: 429–438.

[191] Angeli D (2004) Multi-stability in monotone input/output systems. Systems & Control Letters 51: 185–202.

[192] Kalmar T, Lim C, Hayward P, Muñoz Descalzo S, Nichols J, et al. (2009) Regulated fluctuations in Nanog expression mediate cell fate decisions in embryonic stem cells. PLoS Biol 7: e1000149+.

[193] Meinhardt H, Gierer A (2000) Pattern formation by local self-activation and lateral inhibition. Bioessays 22: 753–760.

[194] Moles CG, Mendes P, Banga JR (2003) Parameter estimation in biochemical pathways: A comparison of global optimization methods. Genome Research 13: 2467–2474.

[195] Kuepfer L, Peter M, Sauer U, Stelling J (2007) Ensemble modeling for analysis of cell signaling dynamics. Nature Biotechnology 25: 1001–1006.

[196] Müller D, Stelling J (2009) Precise regulation of gene expression dynamics favors complex promoter architectures. PLoS Comput Biol 5: e1000279+.

[197] Cosgrove BD, Alexopoulos LG, Hang TC, Hendriks BS, Sorger PK, et al. (2010) Cytokine-associated drug toxicity in human hepatocytes is associated with signaling network dysregulation. Molecular bioSystems 6: 1195–1206.

[198] Carracedo A, Pandolfi PP (2008) The PTEN-PI3K pathway: of feedbacks and cross-talks. Oncogene 27: 5527–5541.

[199] Sanders J, Kandrot E (2010) CUDA by Example: An Introduction to General-Purpose GPU Programming. Addison-Wesley Professional, 1 edition. URL `http://www.worldcat.org/isbn/0131387685`.

[200] Hukushima K, Nemoto K (1996) Exchange Monte Carlo method and application to spin glass simulations. Journal of the Physical Society of Japan 65: 1604–1608.

[201] Li G, Hu J, Wang SW, Georgopoulos PG, Schoendorf J, et al. (2006) Random sampling-high dimensional model representation (RS-HDMR) and orthogonality of its different order component functions. The Journal of Physical Chemistry A 110: 2474–2485.

[202] Fotheringham S, Rogerson PA, editors (2009) The SAGE Handbook of Spatial Analysis. Sage Publications Ltd. URL `http://www.worldcat.org/isbn/141291082X`.

[203] Khachiyan LG (1996) Rounding of polytopes in the real number model of computation. Mathematics of Operations Research 21.

[204] Spath H (1985) Cluster dissection and analysis: Theory, FORTRAN programs, examples (Ellis Horwood series in Computers and their applications). Halsted Press. URL `http://www.worldcat.org/isbn/0470201290`.

[205] Chatterjee S, Hadi AS, Price B (1999) Regression Analysis by Example, 3rd Edition. Wiley-Interscience, 3 edition. URL `http://www.worldcat.org/isbn/0471319465`.

Appendices

Appendix A

Technical Details for the Two-stage Sampling Method

Please note that this part has been written by Elias Zamora from the University of Zurich. I only adapted the terminology to match the other parts of the thesis. It could be found in the supplementary information of article *"Efficient Characterization of High-Dimensional Parameter Spaces for Systems Biology"*.

A.1 Minimum Volume Enclosing Ellipsoid Calculation

The ellipsoid with minimum volume that encloses a set of N data points in a p-dimensional space can be constructed by solving the following optimization problem

$$\begin{cases} \text{Minimize,} & \log\left[\det\left(\mathbf{A}^{-1}\right)\right], \quad \text{Vol} = v_0 \det\left(\mathbf{A}^{-1}\right)^{\frac{1}{2}}, \\ \text{Subject to,} & \left(\mathbf{k}_i - \mathbf{c}\right)\mathbf{A}\left(\mathbf{k}_i - \mathbf{c}\right)' \leq 1, \;\; i = 1, 2, \ldots, N, \end{cases} \tag{A.1}$$

by means of a solver based on the Khachiyan algorithm [203], where $\mathbf{A}$ is the $p \times p$ matrix of the ellipsoid equation in the center form $(\mathbf{k}-\mathbf{c})\cdot\mathbf{A}\cdot(\mathbf{k}-\mathbf{c})' = 1$, c is the center of the ellipsoid, and v_0 is the volume of the unit hypersphere in dimension p.

We estimated the runtime of this procedure as a function of the number of points in a data set. The results (see [159]) show a polynomial $O(N^{2.25})$ dependence on the number of points. The main effect of this polynomial complexity is to make the Monte Carlo integration computationally expensive, because it requires calculation of ellipsoids with minimum volume for large sets of viable parameter points involved in the definition of the integration domain.

To remedy this problem we developed a heuristic algorithm that approximates the ellipsoid obtained by the solver based on the Khachiyan algorithm, but requires much less time. Given a data set B_0 with N points from a p-dimensional space ($N \gg p$), the first step of this heuristic defines a set V_1 composed of n points ($n \ll N$) that are randomly chosen from the initial data set. Then, it calculates the ellipsoid with minimum volume that encloses V_1 by using the solver based on the Khachiyan Algorithm, and obtains the ellipsoid matrix $\mathbf{A}_1$, its center $\mathbf{c}_1$, and its volume Vol_1. After that, the algorithm defines a new data set

$$B_1 = \left\{ \mathbf{k} \in B_0 \,\middle|\, (\mathbf{k} - \mathbf{c}_1)\, \mathbf{A}_1\, (\mathbf{k} - \mathbf{c}_1)' > 1 \right\}, \tag{A.2}$$

that includes all the points in B_0 that were not enclosed by the ellipsoid defined by $\mathbf{A}_1$ and $\mathbf{c}_1$. In the second iteration the algorithm chooses n points randomly from B_1 and adds them to V_1 to form V_2. The algorithm then calculates the ellipsoid with minimum volume that encloses all the points in V_2, as well as its ellipsoid matrix $\mathbf{A}_2$, its center $\mathbf{c}_2$, and its volume Vol_2. The procedure then defines the data set B_2

$$B_2 = \left\{ \mathbf{k} \in B_1 \,\middle|\, (\mathbf{k} - \mathbf{c}_2)\, \mathbf{A}_2\, (\mathbf{k} - \mathbf{c}_2)' > 1 \right\}, \tag{A.3}$$

which forms the basis for the next iteration. The algorithm stops when the ellipsoid volumes converge or when the set B_i is empty. The results of the algorithm are the matrix $\mathbf{A}_i$, and the center $\mathbf{c}_i$ of the ellipsoid with minimum volume calculated in the last iteration before stopping.

By using the same data sets employed in the runtime test of the Khachiyan algorithm, we studied the complexity of our heuristic approach. The results show that, using $n = N/10$, the ellipsoid volumes estimated by the Khachiyan algorithm and our method are quite similar. Nevertheless, the complexity of our approach, albeit polynomial, has a much smaller exponent $O(N^{1.56})$ than the Khachiyan algorithm. In the analysis of the biochemical oscillator with two feedback loops, this property allowed us to calculate the ellipsoid with minimum volume that encloses the set of 19543 viable points found by EBS and AMC approximately 1000 times faster than with the Khachiyan algorithm.

A.2 Construction of the Integration Domain: Determination of the Number of clusters

To calculate the viable volume and to obtain a large set of uniformly distributed viable parameters efficiently, one cannot simply sample over the entire parameter space, because doing so would be too inefficient. It would be much better to perform a uniform sampling over a subspace $\mathcal{B}' \in \mathcal{K}$ that encloses the viable space as tightly as possible.

To construct such a subspace, we use the points obtained in the exploration steps (Adaptive Metropolis Monte Carlo and Ellipsoid-Based Sampling) to obtain a large set of viable parameter points $\mathcal{V}_{MC} \cup \mathcal{V}_{ES}$. To define the subspace I, we group this set into n_C clusters, and calculate the ellipsoids with minimum volume that enclose the viable parameter points grouped in every cluster. The integration domain is then defined by the union of all these ellipsoids.

We next explain the idea behind our heuristic clustering algorithm for cases where the viable space is not convex but could be well approximated by a set of n ellipsoid-like viable regions. In this case, a clustering algorithm should subdivide the space into $n_C = n$ clusters. Also in this case, n ellipsoids defined by n clusters will typically fill much less volume than $n-1$ ellipsoids defined by $n-1$ clusters (Figure A.1), for the following reason. If the viable space has been subdivided only into $n-1$ clusters, at least one of the ellipsoids will typically enclose points from a non ellipsoid-like region, and much of its volume will be filled by non viable points. When we use, on the other hand, n ellipsoids, every ellipsoid will cover one ellipsoid-like region, and many nonviable parameter points will not be covered by any ellipsoid (Figure SA.1), rendering the volume covered by ellipsoids smaller.

When we use $n+1$ ellipsoids defined by $n+1$ clusters, one of the "old" n ellipsoids becomes subdivided into two ellipsoids placed in the same convex region (Figure A.1). The total volume enclosed by the $n+1$ ellipsoids can not be much smaller than the volume filled by n ellipsoids, because the most part of the "old" ellipsoid volume was already filled by viable parameter points (Figure A.1).

In sum, if the viable space can be approximated by n-ellipsoids, the volume of these n-ellipsoids will typically be much smaller than the volume of $n-1$ ellipsoids that cover the viable space, but not much larger than the volume of $n+1$ ellipsoids. In order to choose a number of clusters similar to the number

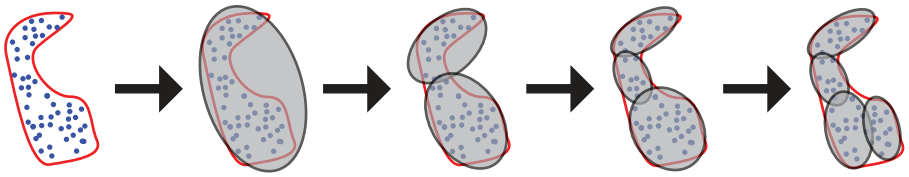

Figure A.1: Clustering of data points for integration domain definition (hypothetical example). Viable volumes are generally non-convex. Therefore, the single ellipsoid that encloses all the viable parameter points (second to left panel) will generally be much larger than the viable volume. It will typically also contain a high proportion of nonviable parameter points. Both features are undesirable. In the hypothetical example shown, when we group the viable points into 2 clusters, the sum of the ellipsoid volumes defined by them is smaller than in the case of a single cluster, but the two ellipsoids still enclose non-convex regions, and are filled by a high proportion of nonviable parameters. In this hypothetical example, the viable space is well approximated by a set of three ellipsoids; therefore, after grouping the viable points into 3 clusters, the ellipsoids defined by them enclose mainly viable parameter points and the sum of their volumes is much smaller than in the case of two clusters but not much higher than the volume defined by 4 clusters.

of ellipsoid-like regions of the parameter space we force this property to be hold. To do so, we found the following heuristic expression useful

$$n_C = \max_i \left(\frac{\mathrm{Vol}_{i+1}\mathrm{Vol}_{i-1}}{\mathrm{Vol}_i^2} \right), \quad \mathrm{Vol}_0 = \mathrm{Vol}_1. \tag{A.4}$$

Here, Vol_i is the sum of i ellipsoid volumes defined by grouping the viable points into i clusters. When the number of clusters increases, then each ellipsoid encloses fewer points, its mean axis length decreases, and the sum of the volume filled by the ellipsoids usually also decreases. That is, $\frac{\mathrm{Vol}_{i+1}}{\mathrm{Vol}_i} < 1$ will normally hold. According to the expression (A.4), n_C will be chosen such that, Vol_{n_C} is much smaller than Vol_{n_C-1} but not much larger than Vol_{n_C+1}, just as we would desire from a clustering algorithm.

Finally, it is worth pointing out that the nonuniform distribution of the set of viable parameter points $\mathcal{V}_{MC} \cup \mathcal{V}_{ES}$ does not allow the use of quality measures that emphasize homogeneity [204] to determine the number of clusters. The techniques used to explore the viable space and obtain $\mathcal{V}_{MC} \cup \mathcal{V}_{ES}$ can sample some part of the viable space in much more detail than others. This creates an artificial high concentration of viable points that is not proportional to the actual density of viable parameter points. Therefore, a method that emphasizes homogeneity would choose the number n_C of clusters based on differences in the density of viable parameter points in $\mathcal{V}_{MC} \cup \mathcal{V}_{ES}$ that are just an artefact of the exploration technique.

A.3 Acquisition of Viable Parameter Points near the Boundary of the Viable Space

In the beginning of every ellipsoid expansion, our EBS technique uses a bisection technique [153] to find $2p$ viable parameter points near the intersection between the boundary of the viable region and the straight lines parallel to the axes of the Cartesian coordinate system that pass trough the viable point $\mathbf{k}_i$. These lines are defined as

$$\mathbf{r}_j \equiv t\,\mathbf{e}_j + \mathbf{k}_i, \quad i = 1, 2, \ldots, p. \quad t \in \mathbb{R}, \tag{A.5}$$

where $\mathbf{e}_j$ is a vector of length one parallel to the j-th axis and $\mathbf{k}_i$ is the viable parameter point from which the j-th ellipsoid expansion starts.

The algorithm first determines whether the intersection point between the boundary of the parameter space and $\mathbf{r}_1$ is viable. If so, this point is stored. If the point is not viable, the algorithm defines the following parameter points

$$\mathbf{a} = \mathbf{k}_v, \quad \mathbf{b} = \Omega \cap \mathbf{r}_1, \quad t > 0, \tag{A.6}$$

where Ω stands for the boundary of the parameter space. Then, the algorithm determines whether the parameter point

$$\mathbf{c} = \frac{\mathbf{b} - \mathbf{a}}{2}. \tag{A.7}$$

is viable. After that, it updates $\mathbf{a}$ and $\mathbf{b}$

$$\begin{cases} \mathbf{a} = \mathbf{c},\ \mathbf{b} = \mathbf{b}, & \text{if } \mathbf{c} \text{ is viable,} \\ \mathbf{a} = \mathbf{a},\ \mathbf{b} = \mathbf{c}, & \text{if } \mathbf{c} \text{ is not viable,} \end{cases} \tag{A.8}$$

and calculates a new parameter point $\mathbf{c}$ from these updated values. It determines this points's viability, and continues iteratively in this manner until the Euclidean norm $\|\mathbf{b} - \mathbf{a}\|$ becomes smaller than a fixed threshold. At this point the parameter point $\mathbf{a}$ is saved as the final estimate of the intersection point between $\mathbf{r}_1$ and the boundary of the viable region. The procedure is repeated for negative values of t, as well as for all other lines (axes) $\mathbf{r}_i$ $(i > 1)$. The result is a set of $2p$ viable parameter points that are close to the boundary of the viable region.

A.4 Choice of starting points for ellipsoid expansions

To be able to explore non-convex viable spaces it is necessary to start ellipsoid expansions from viable parameter points placed in different regions of

parameter space. Thus, after the first ellipsoid expansion started from $\mathbf{k}_1$, we must choose a new starting viable parameter point $\mathbf{k}_2$ from the set composed by $\mathcal{V}_{MC}$ and $\mathcal{V}_{ES,1}$; that is, the set of viable parameter points obtained after the AMC exploration and the first ellipsoid expansion, respectively. We next explain how to choose $\mathbf{k}_2$.

We choose the new starting parameter point preferentially far from the old starting point $\mathbf{k}_1$, because we want to sample regions that have not yet been explored by EBS. To do so, we first calculate the maximum and minimum distances from $\mathbf{k}_1$ to all the viable parameter points included in $\mathcal{V}_{MC}$ and $\mathcal{V}_{ES,1}$

$$\begin{aligned} D_{max,1} &= \max_{\mathbf{k}} \|\mathbf{k}-\mathbf{k}_1\|, \quad \mathbf{k}\in\{\mathcal{V}_{MC},\,\mathcal{V}_{ES,1}\}, \\ D_{min,1} &= \min_{\mathbf{k}} \|\mathbf{k}-\mathbf{k}_1\|, \quad \mathbf{k}\in\{\mathcal{V}_{MC},\,\mathcal{V}_{ES,1}\}. \end{aligned} \tag{A.9}$$

Then, we introduce a stochastic variable D with probability density

$$\rho_1(D) = \begin{cases} 2\dfrac{D-D_{min,1}}{D_{max,1}-D_{min,1}}, & D\in[D_{min,1},\,D_{max,1}], \\ 0, & D\notin[D_{min,1},\,D_{max,1}]. \end{cases} \tag{A.10}$$

Thus, the stochastic variable D takes values close to $D_{max,1}$ with higher probability than values close to $D_{min,1}$.

Next, we sample a scalar D_1 from the distribution (A.10) and define the starting point $\mathbf{k}_2$ for the new ellipsoid expansion as

$$\mathbf{k}_2 = \min_{\mathbf{k}} \left(\|\mathbf{k}-\mathbf{k}_1\| - D_1\right), \quad \mathbf{k}\in\{\mathcal{V}_{MC},\,\mathcal{V}_{ES,1}\}. \tag{A.11}$$

The scalar D_1 has a high probability of being close to $D_{max,1}$; therefore, the starting point $\mathbf{k}_2$ has a high probability of being far from $\mathbf{k}_1$.

The next ellipsoid expansions follow an analogous principle. We calculate the maximum and minimum distances from all the viable points included in the set $\{\mathcal{V}_{MC},\,\mathcal{V}_{ES,1},\,\mathcal{V}_{ES,2},\ldots,\,\mathcal{V}_{ES,i}\}$ to the mean value of all the previous initial points

$$\begin{aligned} D_{max,i} &= \max_{\mathbf{k}} \|\mathbf{k}-\langle\mathbf{k}_v\rangle\|, \quad \mathbf{k}\in\{\mathcal{V}_{MC},\,\mathcal{V}_{ES,1},\ldots,\,\mathcal{V}_{ES,i}\}, \\ D_{min,i} &= \min_{\mathbf{k}} \|\mathbf{k}-\langle\mathbf{k}\rangle\|, \quad \mathbf{k}\in\{\mathcal{V}_{MC},\,\mathcal{V}_{ES,1},\ldots,\,\mathcal{V}_{ES,i}\}, \end{aligned} \tag{A.12}$$

where $\langle\mathbf{k}\rangle$ is the mean value of $\{\mathbf{k}_1,\,\mathbf{k}_{v,2},\ldots,\,\mathbf{k}_{v,i}\}$.

Once $D_{min,i}$ and $D_{max,i}$ are obtained, the stochastic variable D is redefined

$$\rho_1(D) = \begin{cases} 2\dfrac{D-D_{min,i}}{D_{max,i}-D_{min,i}}, & D\in[D_{min,i},\,D_{max,i}], \\ 0, & D\notin[D_{min,i},\,D_{max,i}]. \end{cases} \tag{A.13}$$

A scalar D_i sampled from the distribution (A.13) is used to define the starting point $\mathbf{k}_{i+1}$ for the new ellipsoid expansion

$$\mathbf{k}_{i+1} = \min_{\mathbf{k}} \left(\|\mathbf{k} - \langle \mathbf{k} \rangle\| - D_i \right), \quad \mathbf{k} \in \{\mathcal{V}_{MC}, \mathcal{V}_{ES,1}, \mathcal{V}_{ES,2}, \ldots, \mathcal{V}_{ES,i}\}. \tag{A.14}$$

Because the scalar D_i has a high probability of being close to $D_{max,i}$, the starting point $\mathbf{k}_{i+1}$ has a high probability of being far from $\langle \mathbf{k} \rangle$.

Appendix B

Equations for the Models of the Drosophila Circadian Clock

In this section, the equations of the two models of the *Drosophila* circadian clock are presented (see figure 4.8).

The one-loop model, whose equations follow, was published by Goldbeter in 1995 [104].

$$
\begin{aligned}
\frac{\mathrm{d}[\text{per mRNA}]}{\mathrm{d}t} &= k_1 \frac{k_2^4}{k_2^4 + [\text{nPER}]} - k_3 \frac{[\text{per mRNA}]}{k_4 + [\text{per mRNA}]} - k_{16}[\text{per mRNA}] \\
\frac{\mathrm{d}[\text{PER}]}{\mathrm{d}t} &= k_5[\text{per mRNA}] - k_6 \frac{[\text{PER}]}{k_{10} + [\text{PER}]} + k_7 \frac{[\text{PER}^*]}{k_{10} + [\text{PER}^*]} \\
&\quad -k_{16}[\text{PER}] \\
\frac{\mathrm{d}[\text{PER}^*]}{\mathrm{d}t} &= k_6 \frac{[\text{PER}]}{k_{10} + [\text{PER}]} - k_7 \frac{[\text{PER}^*]}{k_{10} + [\text{PER}^*]} - k_8 \frac{[\text{PER}^*]}{k_{10} + [\text{PER}^*]} \\
&\quad +k_9 \frac{[\text{PER}^{**}]}{k_{10} + [\text{PER}^{**}]} - k_{16}[\text{PER}^*] \\
\frac{\mathrm{d}[\text{PER}^{**}]}{\mathrm{d}t} &= +k_8 \frac{[\text{PER}*]}{k_{10} + [\text{PER}^*]} - k_9 \frac{[\text{PER}^{**}]}{k_{10} + [\text{PER}^{**}]} - k_d(t) \frac{[\text{PER}^{**}]}{k_{12} + [\text{PER}^{**}]} \\
&\quad -k_{13}[\text{PER}^{**}] + k_{14}[\text{nPER}] - k_{16}[\text{PER}^{**}] \\
\frac{\mathrm{d}[\text{nPER}]}{\mathrm{d}t} &= +k_{13}[\text{PER}^{**}] - k_{14}[\text{nPER}] - k_{15}[\text{nPER}] \qquad \text{(B.1)}
\end{aligned}
$$

with PER** degradation that depends on light as $k_d(t) = k_{11} * (1.5 + ((atan(10 * (\mathrm{mod}(t-12, 24) - 12)) - atan(10 * (\mathrm{mod}(t, 24) - 12))) \frac{2}{\pi})$. With this function, k_d

is equal to $2k_{11}$ from 0 to 12 hours and k_{11} from 12 to 24 hours with a smooth continuous transition.

The two-loop model, whose equations follow, was published by Leloup and Goldbeter in 1998 [105].

$$
\begin{aligned}
\frac{\mathrm{d}[\text{per mRNA}]}{\mathrm{d}t} &= k_1 \frac{k_2^4}{k_2^4 + [\text{nTIM-PER}]} - k_3 \frac{[\text{per mRNA}]}{k_4 + [\text{per mRNA}]} - k_{26}[\text{per mRNA}] \\
\frac{\mathrm{d}[\text{PER}]}{\mathrm{d}t} &= k_5[\text{per mRNA}] - k_6 \frac{[\text{PER}]}{k_{10} + [\text{PER}]} + k_7 \frac{[\text{PER}^*]}{k_{10} + [\text{PER}^*]} \\
&\quad -k_{26}[\text{PER}] \\
\frac{\mathrm{d}[\text{PER}^*]}{\mathrm{d}t} &= k_6 \frac{[\text{PER}]}{k_{10} + [\text{PER}]} - k_7 \frac{[\text{PER}^*]}{k_{10} + [\text{PER}^*]} - k_8 \frac{[\text{PER}^*]}{k_{10} + [\text{PER}^*]} \\
&\quad +k_9 \frac{[\text{PER}^{**}]}{k_{10} + [\text{PER}^{**}]} - k_{26}[\text{PER}^*] \\
\frac{\mathrm{d}[\text{PER}^{**}]}{\mathrm{d}t} &= +k_8 \frac{[\text{PER}*]}{k_{10} + [\text{PER}^*]} - k_9 \frac{[\text{PER}^{**}]}{k_{10} + [\text{PER}^{**}]} - k_{11} \frac{[\text{PER}^{**}]}{k_{12} + [\text{PER}^{**}]} \\
&\quad -k_{20}[\text{PER}^{**}][\text{TIM}^{**}] + k_{21}[\text{TIM-PER}] - k_{26}[\text{PER}^{**}] \\
\frac{\mathrm{d}[\text{tim mRNA}]}{\mathrm{d}t} &= k_1 \frac{k_2^4}{k_2^4 + [\text{nTIM-PER}]} - k_3 \frac{[\text{tim mRNA}]}{k_4 + [\text{tim mRNA}]} - k_{26}[\text{tim mRNA}] \\
\frac{\mathrm{d}[\text{TIM}]}{\mathrm{d}t} &= k_{13}[\text{tim mRNA}] - k_{14} \frac{[\text{TIM}]}{k_{10} + [\text{TIM}]} + k_{15} \frac{[\text{TIM}^*]}{k_{10} + [\text{TIM}^*]} \\
&\quad -k_{26}[\text{TIM}] \\
\frac{\mathrm{d}[\text{TIM}^*]}{\mathrm{d}t} &= k_{14} \frac{[\text{TIM}]}{k_{10} + [\text{TIM}]} - k_{15} \frac{[\text{TIM}^*]}{k_{10} + [\text{TIM}^*]} - k_{16} \frac{[\text{TIM}^*]}{k_{10} + [\text{TIM}^*]} \\
&\quad +k_{17} \frac{[\text{TIM}^{**}]}{k_{10} + [\text{TIM}^{**}]} - k_{26}[\text{TIM}*] \\
\frac{\mathrm{d}[\text{TIM}^{**}]}{\mathrm{d}t} &= +k_{16} \frac{[\text{TIM}*]}{k_{10} + [\text{TIM}*]} - k_{17} \frac{[\text{TIM}^{**}]}{k_{10} + [\text{TIM}^{**}]} - k_d(t) \frac{[\text{TIM}^{**}]}{k_{19} + [\text{TIM}^{**}]} \\
&\quad -k_{20}[\text{PER}^{**}][\text{TIM}^{**}] + k_{21}[\text{TIM-PER}] - k_{26}[\text{TIM}^{**}] \\
\frac{\mathrm{d}[\text{TIM-PER}]}{\mathrm{d}t} &= k_{20}[\text{PER}^{**}][\text{TIM}^{**}] - k_{21}[\text{TIM-PER}] - k_{22}[\text{TIM-PER}] \\
&\quad +k_{23}[\text{nTIM-PER}] - k_{24}[\text{TIM-PER}] \\
\frac{\mathrm{d}[\text{nTIM-PER}]}{\mathrm{d}t} &= k_{22}[\text{TIM-PER}] - k_{23}[\text{nTIM-PER}] - k_{25}[\text{nTIM-PER}]
\end{aligned}
\tag{B.2}
$$

with TIM** degradation that depends on light as $k_d(t) = k_{18} * (1.5 + ((atan(10 * (\mathrm{mod}(t-12, 24) - 12)) - atan(10 * (\mathrm{mod}(t, 24) - 12))) \frac{2}{\pi})$. With this function, k_d is equal to $2k_{18}$ from 0 to 12 hours and k_{18} from 12 to 24 hours with a smooth continuous transition.

In the original model by Leloup and Goldbeter [105], the kinetics parameters of the two loops were identical: $k_{13} = k_5$, $k_{14} = k_6$, $k_{15} = k_7$, $k_{16} = k_8$, $k_{17} = k_9$, $k_{18} = k_{11}$ and $k_{18} = k_{12}$. In this work, we distinguished the symmetrical two-loop model where this constraint is kept for the sampling procedure

and the asymmetrical two-loop model where each parameter is individually sampled.

Appendix C

Models, Algorithms and Supplementary Results for the Synthetic Biology Circuit Design

C.1 Langevin Model for the Systems 2, 3 and 4

The results for the time-scale analysis, the module optimization and the different robustness analysis are performed using Langevin simulations [10]. The Langevin simulations are a compromise between efficiency and a realistic implementation of molecular noise. A modulated white noise is added to each reaction at each integration step. The parameter that control the amplitude of the noise is Ω. We refer to it as 'volume of the cell', because it makes the correspondence between the concentration and the amount of molecules in the cell. For example, a volume of 200 means that a concentration of 1 (which is the average value for most of the components when they are active) corresponds to a total of 200 molecules in the cell. The maximum number of cells in the system set to N_{max}. The number of cells during the simulation should not reach this threshold otherwise it may mean that the number of cells may not be under control. We choose a maximum of 150 cells for all simulations (other values does not change qualitatively the comparisons between the systems). The output of the simulations is the number of committed cells over time.

Ideally, it should be constant with the smallest possible fluctuations. We measure this property with the signal to noise ratio S/N of the fraction of committed cells ρ_c in a simulation of duration T:

$$S/N = \frac{\overline{\rho_c}}{\sqrt{1/T \int_0^T (\rho_c(t) - \overline{\rho_c})^2 \, dt}} \qquad \text{where} \quad \overline{\rho_c} = 1/T \int_0^T \rho_c(t) dt \qquad \text{(C.1)}$$

The simulations are started a certain time (usually 500 hours) before the 'recording' of the populations to leave enough time for the system to relax and reach its long-run regime.

The evolution of each component in each model (main text, figures 4.21 to 4.24) follows a Hill kinetic with a coefficient of $n = 4$ that assumes cooperativity. This modelism simplifies the activation and inhibition mechanisms. As describe in the section 4.6.2, such specific input-output functions is not trivial and can be obtained by adjusting the binding and dissociation rates in the mechanistic model. If necessary, a higher sensitivity could be obtained by increasing the number of elements in the cascade.

We will now details the equations for the different elements present in systems 2 to 4. We choose the following nomenclature. The maximum rate is named k_p^α and half-rate constant H_α for component α. Degradation follows mass-action kinetics (rate k_d^α) for all components. Finally, diffusion is a linear function (rate k_{diff}) of the difference between internal and external ($AI\alpha_{out}$) concentrations.

C.1.1 Quorum Sensing Module

In systems 2 to 4, $AI1$ is the signaling component for uncommitted cells and its expression, through $I1$, is regulated by the toggle switch ($R6$ and $R7$ cross-inhibition): its production is inhibited by $R7$ as $R7$ is produced only when the cell is committed. On the contrary, $AI2$, the signal for committed cells, is inhibited by the other component of the switch, $R6$, that is produced in uncommitted cells. We simplify the equations by having the concentration of the AI depending directly on $R6$ and $R7$. The dynamical equations for the

concentration of these components are therefore:

$$\begin{aligned}\frac{\mathrm{d}\ AI1}{\mathrm{d}t} &= k_p^{AI1}\frac{H_{AI1}^n}{H_{AI1}^n + R7^n} \\ &\quad -k_d^{AI1}AI1 \\ &\quad +k_{diff}(AI1_{out} - AI1) \end{aligned} \tag{C.2}$$

$$\begin{aligned}\frac{\mathrm{d}\ AI2}{\mathrm{d}t} &= k_p^{AI2}\frac{H_{AI2}^n}{H_{AI2}^n + R6^n} \\ &\quad -k_d^{AI2}AI2 \\ &\quad +k_{diff}(AI2_{out} - AI2) \end{aligned} \tag{C.3}$$

And the time evolution of the external concentrations of these components follows:

$$\begin{aligned}\frac{\mathrm{d}\ AI1_{out}}{\mathrm{d}t} &= -\sum_{Cells}\frac{k_{diff}}{N_{max}}(AI1_{out} - AI1_{Cells}) \\ &\quad -k_d^{AI1}AI1_{out} \end{aligned} \tag{C.4}$$

$$\begin{aligned}\frac{\mathrm{d}\ AI2_{out}}{\mathrm{d}t} &= -\sum_{Cells}\frac{k_{diff}}{N_{max}}(AI2_{out} - AI2_{Cells}) \\ &\quad -k_d^{AI2}AI2_{out} \end{aligned} \tag{C.5}$$

Out of these equations, the values for the dynamical equilibrium of the concentrations of $AI1$ and $AI2$ in uncommitted cells can be calculated as a function of the number of uncommitted and committed cells. We assume that the production rate in eq. (C.2) and (C.3) is either zero or maximal ($k_p^{AI\alpha}$) depending on the state of each cell. The size of both cell populations is expressed as a fraction of N_{max}: $\rho_u = \frac{N_u}{N_{max}}$ for uncommitted cells and $\rho_c = \frac{N_c}{N_{max}}$ for the committed ones.

$$\overline{AI1(\rho_u,\rho_c)} = \frac{k_p^{AI1}}{k_d^{AI1}}\frac{(k_d^{AI1}+k_{diff})(k_d^{AI1}+\rho_u k_{diff}) + \rho_c k_{diff}k_d^{AI1}}{(k_{diff}+k_d^{AI1})(k_{diff}+k_d^{AI1}+\rho_u k_{diff}+\rho_c k_{diff})} \tag{C.6}$$

$$\overline{AI2(\rho_u,\rho_c)} = \frac{k_p^{AI2}}{k_d^{AI2}}\frac{\rho_c k_{diff}^2}{(k_{diff}+k_d^{AI2})(k_{diff}+k_d^{AI2}+\rho_u k_{diff}+\rho_c k_{diff})} \tag{C.7}$$

We can notice that the sensitivity of the $AI1$ function to ρ_c ($\left|\frac{\partial AI1}{\partial \rho_c}\right|$) is lower than the one for $AI2$ due to the internal production of $AI1$ in uncommitted cells. The difference between both modules is reduced when the diffusion is increased. Note that we are not interested in the concentration of $AI1$ and

$AI2$ in committed cells since their fate are not influenced by the concentration of $AI1$ or $AI2$ (they remain committed until their death).

Knowing these functions allows calculating the half-rate constant for the production terms of the elements in the population control modules, $A1/R1$ and $R2$. We name $\overline{AI1} = AI1(\overline{\rho_u}, \overline{\rho_t})$ and $\overline{AI2} = AI2(\overline{\rho_u}, \overline{\rho_t})$ the concentration of $AI1$, resp. $AI2$, in uncommitted cells that corresponds to a fraction of uncommitted cells $\overline{\rho_u}$ (chosen around 0.45) and of committed cells $\overline{\rho_c}$ (around 0.4). $AI1$ binds to the receptor $Rec1$ that control the production of the component $A1$ (or for system 5, the repressor $R1$). $AI2$ binds to $Rec2$ and activates the production of the repressor $R2$. In the models, the expressions of $A1/R1$ and $R2$ is simplified: they depend directly on the concentration of $AI1$ or $AI2$.

$$\begin{aligned}
\frac{\mathrm{d}\ A1}{\mathrm{d}t} &= k_p^{A1}\frac{AI1^n}{H_{A1}^n + AI1^n} \quad \text{with} \quad H_{R1} = \overline{AI1} \\
&\quad -k_d^{A1} A1 \qquad \text{for systems 3 and 4} \qquad \text{(C.8)} \\
\frac{\mathrm{d}\ R1}{\mathrm{d}t} &= k_p^{R1}\frac{H_{R1}^n}{H_{R1}^n + AI1^n} \quad \text{with} \quad H_{R1} = \overline{AI1} \\
&\quad -k_d^{R1} R1 \qquad \text{for system 5} \qquad \text{(C.9)} \\
\frac{\mathrm{d}\ R2}{\mathrm{d}t} &= k_p^{R2}\frac{H_{R2}^n}{H_{R2}^n + AI2^n} \quad \text{with} \quad H_{R2} = \overline{AI2} \\
&\quad -k_d^{R2} R2 \qquad \text{(C.10)}
\end{aligned}$$

C.1.2 AND Gate and Toggle Switch

We will now focus on the AND gate and the toggle switch. The AND gate is integrating the information of the two quorum sensing modules (and the oscillator or the throttle in systems 3 and 4). To integrate all inputs, systems 3 and 4 comprise an activator $A3$ and an additional repressor $R4$ in system 3. For all systems, the final signal is $R5$. The connections are made such that cells differentiate only if the number of uncommitted cells is high enough and if the number of committed cells is too low. The dynamical equations follow the same Hill term for activation/inhibition and mass-action for degradation.

For system 2, that only have $R5$, the equation is:

$$\begin{aligned}
\frac{\mathrm{d}\ R5}{\mathrm{d}t} &= k_p^{R5}\frac{A1^n}{H_{R5-1}^n + A1^n}\frac{H_{R5-2}^n}{H_{R5-2}^n + R2^n} \\
&\quad -k_d^{R5} R5 \qquad \text{(C.11)}
\end{aligned}$$

In system 3, the AND gate includes two additional elements $A3$ and $R4$ to incorporate the information of the oscillator whose output is $Ro2$ (see below):

$$\begin{aligned}
\frac{\mathrm{d}\ A3}{\mathrm{d}t} &= k_p^{A3}\frac{A1^n}{H_{A3-1}^n + A1^n}\frac{H_{A3-4}^n}{H_{A3-4}^n + R4^n} \\
&\quad -k_d^{A3}A3 \qquad \text{(C.12)} \\
\frac{\mathrm{d}\ R4}{\mathrm{d}t} &= k_p^{R4}\frac{H_{R4}^n}{H_{R4}^n + Ro2^n} \\
&\quad -k_d^{R4}R4 \qquad \text{(C.13)} \\
\frac{\mathrm{d}\ R5}{\mathrm{d}t} &= k_p^{R5}\frac{A3^n}{H_{R5-3}^n + A3^n}\frac{H_{R5-2}^n}{H_{R5-2}^n + R2^n} \\
&\quad -k_d^{R5}R5 \qquad \text{(C.14)}
\end{aligned}$$

In system 4, the throttle through the signaling molecule $AI3$ is acting on $R5$ as the other signaling molecules of the quorum sensing module. We also need to include an element $A3$ prior to $R5$:

$$\begin{aligned}
\frac{\mathrm{d}\ A3}{\mathrm{d}t} &= k_p^{A3}\frac{H_{A3-1}^n}{H_{A3-1}^n + R1^n}\frac{H_{A3-2}^n}{H_{A3-2}^n + R2^n} \\
&\quad -k_d^{A3}A3 \qquad \text{(C.15)} \\
\frac{\mathrm{d}\ R5}{\mathrm{d}t} &= k_p^{R5}\frac{A3^n}{H_{R5-3}^n + A3^n}\frac{H_{R5-t}^n}{H_{R5-t}^n + AI3^n} \\
&\quad -k_d^{R5}R5 \qquad \text{(C.16)}
\end{aligned}$$

The toggle switch comprises the two repressors, $R6$ and $R7$, inhibiting each other. All cells start their life in the uncommitted state with a high level of $R6$, therefore a low level of $R7$. The component of the toggle switch, $R6$, is inhibited (and the toggle may change state) when $R5$ is produced. In addition to this regulation, $R6$ is inhibited by $R7$ to produce the switch behavior that is related to the cell fate; $R6$ is therefore a NOR gate with these two inputs. The equations for these components are:

$$\begin{aligned}
\frac{\mathrm{d}\ R6}{\mathrm{d}t} &= k_p^{R6}\frac{H_{R6-5}^n}{H_{R6-5}^n + R5^n}\frac{H_{R6-7}^n}{H_{R6-7}^n + R7^n} \\
&\quad -k_d^{R6}R6 \qquad \text{(C.17)} \\
\frac{\mathrm{d}\ R7}{\mathrm{d}t} &= k_p^{R7}\frac{H_{R7}^n}{H_{R7}^n + R6^n} \\
&\quad -k_d^{R7}R7 \qquad \text{(C.18)}
\end{aligned}$$

The equation for $R7$ is different for the system 5 as the toggle includes the throttle system. The details are discussed below.

C.1.3 Cell Fate

Cells, in their uncommitted state, can divide and, in the committed state, die. The division process has to be controlled in order to avoid proliferation of the cells. This is implemented with the production of a growth arrest factor (GAF) controlled by the quorum sensing. The commitment is controlled by the component $GATA$: when $GATA$ reaches a certain threshold the cell is considered as committed and slowly dies. These processes are implemented as follow.

Division Process

GAF is controlled by $A1/R1$ (quorum sensing of the uncommitted cells):

$$\begin{aligned} \frac{\mathrm{d}\ GAF}{\mathrm{d}t} &= k_p^{GAF}\frac{A1^n}{H_{GAF}^n + A1^n} \\ &\quad -k_d^{GAF}GAF \qquad \text{for systems 3 and 4} \qquad \text{(C.19)} \\ \frac{\mathrm{d}\ GAF}{\mathrm{d}t} &= k_p^{GAF}\frac{H_{GAF}^n}{H_{GAF}^n + R1^n} \\ &\quad -k_d^{GAF}GAF \qquad \text{for system 5} \qquad \text{(C.20)} \end{aligned}$$

If GAF is below a threshold, th_{GAF}, the cell grows. To model that, we use an integrator for the division depending on GAF level:

$$Div(t) = \int_{t_0}^{t} k_b\Theta(th_{GAF} - GAF(\tau)) - \frac{k_b}{3}\Theta(GAF(\tau) - th_{GAF})d\tau$$

Where t_0 is the birth of the cell (beginning of the simulation or after cell division) and Θ is the Heaviside function ($\Theta(x) = 0$ if $x < 0$ and $\Theta(x) = 1$ if $x \geq 0$). The division rate is k_b. We choose $k_b = \frac{1}{96}h^{-1}$, such that the average time for division (in absence of GAF), is set to $96h$. The value of Div cannot be negative.

Division occurs when $Div(t) \geq 1$. The two daughter cells inherit the concentrations of all the components of the mother cell except the value Div that is reset to zero.

Commitment Process

$GATA$ is inhibited by $R6$. When the toggle switches to high level of $R7$, $GATA$ starts to be produced:

$$\begin{aligned}\frac{\mathrm{d}\ GATA}{\mathrm{d}t} &= k_p^{GATA}\frac{H_{GATA}^n}{H_{GATA}^n + R6^n} \\ &\quad -k_d^{GATA}GATA \end{aligned} \tag{C.21}$$

$GATA$ concentration quickly rises and when it reaches a threshold, the cell changes fate and becomes committed. This is a strong simplification as the biological process is much more complex, but it is sufficient for the purpose of this analysis. As a committed cell, it can only die. The lifetime of the committed cells is implemented with an integrator for cell death:

$$Death(t) = \int_{t_0}^{t} k_k d\tau$$

Where t_0 is the time of commitment. The division rate is k_k. We choose $k_k = \frac{1}{200}h^{-1}$, such that the average time for division (in absence of GAF), is set to $200h$.

Cell dies when $Death(t)$ reaches 1.

C.1.4 System 3 – Implementation of an Oscillator

The aim of the oscillator is to differentiate the behavior the cells through the population, therefore one of its main qualities should be irregularity. The simplest possible oscillator is made of a component Ao that activates itself (autopositive feedback) and regulates the expression of a repressor Ro that inhibits Ao. The readout of the system is implemented with two successive components: a second repressor $Ro2$ is acting on $R4$ that represses cell commitment. With proper parameter values, this system generates short interval of low $R4$ concentration with an irregular latency where $R4$ concentration remains high. The

equations of the oscillator are:

$$\begin{aligned}
\frac{\mathrm{d}\ Ao}{\mathrm{d}t} &= k_p^{Ao}\left(k_0^{Ao} + \frac{Aosc^n}{H_{Ao-A}^n + Ao^n}\right)\frac{H_{Ao-R}^n}{H_{Ao-R}^n + Rosc^n} \\
&\quad -k_d^{Ao} Aosc \qquad \text{(C.22)} \\
\frac{\mathrm{d}\ Ro}{\mathrm{d}t} &= k_p^{Ro}\frac{Ao^n}{H_{Ro}^n + Ao^n} \\
&\quad -k_d^{Ro} Ro \qquad \text{(C.23)} \\
\frac{\mathrm{d}\ Ro2}{\mathrm{d}t} &= k_p^{Ro2}\frac{Ao^n}{H_{Rosc2}^n + Ao^n} \\
&\quad -k_d^{Ro2} Ro2 \qquad \text{(C.24)}
\end{aligned}$$

Linking the oscillator and the quorum sensing module to the toggle switch occurs in two steps as described above. The element $A3$ is controlled by both $A1$ and $R4$, therefore $A3$ is produced only when $A1$ is high (enough uncommitted cells) and $R4$ low (Ao peaking).

C.1.5 System 4 – Implementation of a Throttle

The production of the signaling molecule, $AI3$, is controlled by an activator At and represses by $R7$ such that $AI3$ is produced transiently when the toggle is switching. The use of an intermediate component ensures that the toggle will not be able to switch back and also allows an amplification of the $AI3$ production. $AI3$ diffuses through the membrane, gives rise to a external concentration $AI3_{out}$ that enters other cells. The dynamical equations are:

$$\begin{aligned}
\frac{\mathrm{d}\ At}{\mathrm{d}t} &= k_p^{At}\frac{H_{At-6}^n}{H_{At-6}^n + R6^n} \\
&\quad -k_d^{At} At \qquad \text{(C.25)} \\
\frac{\mathrm{d}\ AI3}{\mathrm{d}t} &= k_p^{AI3}\frac{At^n}{H_{AI3-t}^n + At^n}\frac{H_{AI3-7}^n}{H_{AI3-7}^n + R7^n} \\
&\quad -k_d^{AI3} AI3 \\
&\quad +k_{diff}(AI3_{out} - AI3) \qquad \text{(C.26)} \\
\frac{\mathrm{d}\ AI3_{out}}{\mathrm{d}t} &= -\sum_{Cells}\frac{k_{diff}}{N_{max}}(AI3_{out} - AI3_{Cells}) \\
&\quad -k_d^{AI3} AI3_{out} \qquad \text{(C.27)}
\end{aligned}$$

The remaining elements are similar to systems 2 and 3. There are only two differences from system 2: first, the addition of $A3$ in order that $AI3$ is

acting on $R5$ (equation above). Second, the control of $R7$ by At instead of $R6$ directly:

$$\begin{aligned} \frac{\mathrm{d}\ R7}{\mathrm{d}t} &= k_p^{R7} \frac{At^n}{H_{R7}^n + At^n} \\ &\quad -k_d^{R7} R7 \end{aligned} \tag{C.28}$$

C.1.6 Logic Tables for Systems 2 to 4

To clarify the mechanisms of the different systems, we express the component interactions in a logic table (tables C.1 to C.3) emphasizing the relations that lead to commitment.

Component	Logical relation	$\rho_u < \overline{\rho_u}$		$\rho_u > \overline{\rho_u}$	
		$\rho_c > \overline{\rho_c}$	$\rho_c < \overline{\rho_c}$	$\rho_c > \overline{\rho_c}$	$\rho_c < \overline{\rho_c}$
$A1$	$= AI1 > \overline{AI1}$	low	low	high	**high**
GAF	$= A1$	low	low	high	high
$R2$	$= AI2 > \overline{AI2}$	high	low	high	**low**
$R5$	$= A1 \bigwedge \neg R2$	low	low	low	**high**

Component	Logical relation	$R5$ low	$R5$ high ($\rho_u > \overline{\rho_u}$ AND $\rho_c < \overline{\rho_c}$)
$R6$	$= \neg R5$	high	**low**
$R7$	$= \neg R6$	low	**high**
$GATA$	$= \neg R6$	low	**high**
Cell fate	$= \neg R6$	Uncommitted	**Committing**

Table C.1: Logic table for system 2.

Component	Logical relation	Ao low	Ao high (transient while Ro is low)
Ro	$= Ao$	low	high
$Ro2$	$= Ao$	low	**high**
$R4$	$= \neg Ro2$	high	**low**

Component	Logical relation	$\rho_u < \overline{\rho_u}$		$\rho_u > \overline{\rho_u}$			
		$\rho_c > \overline{\rho_c}$	$\rho_c < \overline{\rho_c}$	$\rho_c > \overline{\rho_c}$		$\rho_c < \overline{\rho_c}$	
		$R4$ not important		$R4$ high	$R4$ low	$R4$ high	$R4$ low
$A1$	$= AI1 > \overline{AI1}$	low	low	high	high	high	**high**
GAF	$= A1$	low	low	high	high	high	high
$R2$	$= AI2 > \overline{AI2}$	high	low	high	high	low	**low**
$A3$	$= A1 \bigwedge \neg R4$	low	low	low	high	low	**high**
$R5$	$= A3 \bigwedge \neg R2$	low	low	low	low	low	**high**

Component	Logical relation	$R5$ low	$R5$ high ($\rho_u > \overline{\rho_u}$ AND $R4$ low AND $\rho_c < \overline{\rho_c}$)
$R6$	$= \neg R5$	high	**low**
$R7$	$= \neg R6$	low	**high**
$GATA$	$= \neg R6$	low	**high**
Cell fate	$= \neg R6$	Uncommitted	**Committing**

Table C.2: Logic table for system 3 (oscillator).

Component	Logical relation	$\rho_u < \overline{\rho_u}$		$\rho_u > \overline{\rho_u}$	
		$\rho_c > \overline{\rho_c}$	$\rho_c < \overline{\rho_c}$	$\rho_c > \overline{\rho_c}$	$\rho_c < \overline{\rho_c}$
$R1$	$= AI1 < \overline{AI1}$	high	high	low	**low**
GAF	$= \neg R1$	low	low	high	high
$R2$	$= AI2 > \overline{AI2}$	high	low	high	**low**
$A3$	$= \neg(R1 \bigvee R2)$	low	low	low	**high**

Component	Logical relation	$A3$ low		$A3$ high ($\rho_u > \overline{\rho_u}$ AND $\rho_c < \overline{\rho_c}$)	
		$AI3$ low	$AI3$ high	$AI3$ low	$AI3$ high
$R5$	$= A3 \bigwedge \neg AI3$	low	low	low	**high**
$R6$	$= \neg R5$	high	high	high	**low**
$R7$	$= \neg R6$	low	low	low	**high**
$GATA$	$= \neg R6$	low	low	low	**high**
Cell fate	$= \neg R6$	Uncommitted			**Committing**

Component	Logical relation	$R6$ high $R7$ low	$R6$ low $R7$ high	$R6$ and $R7$ medium (transient)
At	$= \neg(R6 \bigvee R7)$	low	low	high
$AI3$	$= At$	low	low	high

Table C.3: Logic table for the system 4 (throttle).

C.1.7 Parameters and State Variables for Langevin Models of Systems 2 to 4

Module	System 2			System 3			System 4		
	Name	var. #	Init. val.	Name	var. #	Init. val.	Name	var. #	Init. val.
	In each cell								
Quorum singaling	$AI1$	1	high	identical			identical		
	$AI2$	2	zero	identical			identical		
Quorum sensing module	$A1$	3	medium	identical			identical		
	$R2$	4	low	identical			identical		
	$A3$	5	low	identical			identical		
Commitment module	$R5$	6	low	identical			identical		
	$R6$	7	high	identical			identical		
	$R7$	8	low	identical			identical		
Cell fate components	GAF	9	low	identical			identical		
	$GATA$	10	low	identical			identical		
Additional modules	–			Ao	11	random (low to med.)	At	11	low
	–			Ro	12	low	$AI3$	12	low
	–			$Ro2$	13	low	–		
	–			$R4$	14	low	–		
Cell fate	Div	(intern)	random	identical			identical		
	$Death$	(intern)	inactive	identical			identical		
	External								
Quorum singaling (extern)	$AI1_{out}$	1∗	at eq.	identical			identical		
	$AI2_{out}$	2∗	zero	identical			identical		
	–			–			$AI3_{out}$	3∗	zero

Table C.4: State variable numbering. Initial values are based on an initial population of uncommitted cells only below the target fraction. 'High', 'medium' and 'low' refer to values corresponding to $0.95 \cdot k_p/k_d$, $0.5 \cdot k_p/k_d$ and $0.05 \cdot k_p/k_d$, respectively (see table C.5).

Module	System 2			System 3			System 4		
	Name	para. #	Value	Name	para. #	Value	Name	para. #	Value
Quorum singaling	k_p^{AI1}	1	2.0	identical			identical		
	H_{AI1}	2	1.5	identical			identical		
	k_d^{AI1}	3	0.2	identical			identical		
	k_p^{AI2}	4	2.0	identical			identical		
	H_{AI2}	5	0.45	identical			identical		
	k_d^{AI2}	6	0.2	identical			identical		
	k_{diff}	7	5.0	identical			identical		
Quorum sensing module	k_p^{A1}	8	1.0	identical to Syst. 2			k_p^{R1}	8	1.0
	H_{A1}	9	2.7	identical to Syst. 2			H_{R1}	9	2.7
	k_d^{A1}	10	1.0	identical to Syst. 2			k_d^{R1}	10	1.0
	k_p^{R2}	11	1.0	identical			identical		
	H_{R2}	12	2.0	identical			identical		
	k_d^{R2}	13	1.0	identical			identical		
	–			k_p^{A3}	14	2.0	identical		
	–			H_{A3-1}	15	0.5	identical		
	–			H_{A3-4}	16	0.5	H_{A3-2}	16	0.5
	–			k_d^{A3}	17	2.0	identical		
Commitment module	k_p^{R5}	18	1.0	identical			identical		
	H_{R5-1}	19	0.6	H_{R5-3}	19	0.6	identical		
	H_{R5-2}	20	0.5	identical to Syst. 2			H_{R5-t}	20	0.9
	k_d^{R5}	21	1.0	identical			identical		
	k_p^{R6}	22	1.0	identical			identical		
	H_{R6-5}	23	0.5	identical			identical		
	H_{R6-7}	24	0.5	identical			identical		
	k_d^{R6}	25	1.0	identical			identical		
	k_p^{R7}	26	3.0	identical			identical		
	H_{R7}	27	0.4	identical			identical		
	k_d^{R7}	28	1.0	identical			identical		
Cell fate parameters	k_p^{GAF}	29	1.7	identical			identical		
	H_{GAF}	30	0.7	identical to Syst. 2			H_{GAF}	30	0.3
	k_d^{GAF}	31	1.0	identical			identical		
	k_p^{GATA}	32	1.0	identical			identical		
	H_{GATA}	33	0.2	identical			identical		
	k_d^{GATA}	34	1.0	identical			identical		
Additional modules	–			k_p^{Ao}	35	50	k_p^{At}	35	4.0
	–			k_0^{Ao}	36	0.0002	H_{At}	36	0.5
	–			H_{Ao-A}	37	0.5	k_d^{At}	38	4.0
	–			H_{Ao-R}	38	0.01	k_p^{AI3}	39	200
	–			k_d^{Ao}	39	0.1	H_{AI3-t}	40	0.6
	–			k_p^{Ro}	40	2.5	H_{AI3-7}	37	0.5
	–			H_{Ro}	41	0.5	k_d^{AI3}	41	0.5
	–			k_d^{Ro}	42	0.04	–		
	–			k_p^{Ro2}	43	5	–		
	–			H_{Ro2}	44	0.2	–		
	–			k_d^{Ro2}	45	2.0	–		
	–			k_p^{R4}	46	1.0	–		
	–			H_{R4}	47	0.9	–		
	–			k_d^{R4}	48	1.0	–		

Table C.5: Parameters for the Langevin models of systems 2 to 4.

C.1.8 Details for the Different Simulations

The results presented in the section 4.6.1 for system 2 to 4 and the analysis (expect the quorum sensing optimization) are performed with stochastic simulations using a Langevin algorithm [10] with the equations given above. For all simulations, if not mentioned otherwise, Ω equals 200, the maximal number of cells is 150, the average division time (in absence of negative feedback) is 96 hours ($k_b = \frac{1}{96}h^{-1}$) and the average survival time of a committed cell is 200 hours ($k_k = \frac{1}{200}h^{-1}$). All simulations start with a low uncommitted cell density and evolved for 1000 hours and the recording lasts 3000 hours.

Spatial Simulations

For the results about spatial bias, spatial simulations using the Langevin model are performed. To mimic the spatial distribution, the volume of the system is divided in a grid with $N_{max} = 150$ boxes (the grid has grid has dimensions 6, 5 and 5), each with the same volume as a cell. Each cell is occupying a box on the grid. The diffusion can occur between the cell and its box or between the boxes. As diffusion is physically faster than any other process in the cell, it is simulated without the noise term (standard ordinary differential equation). Diffusion is occurring from one edge to the other one (periodic boundary conditions) in order to mimic a larger system and avoid border effect. A cell dividing is pushing the other cells in chain until there is an empty box, therefore daughter cells are contiguous. For these simulations, the recording time was 10^4 hours to obtain more significant results.

Robustness to Variation of the Killing Rate k_k

To test the influence of the killing rate in the models, we change the value of k_k from $\frac{1}{45}h^{-1}$ to $\frac{1}{550}h^{-1}$ (corresponding to a ratio k_b/k_k of 0.5 to 6) as shown in figure 4.26A, D and 4.27B, D. For each value of k_k, 24 simulations are performed and the average S/N value is plotted in figure 4.26A, D and the population densities in 4.26B, E. The error bar is the standard deviation of the results of the 24 simulations.

Robustness to Variations of Molecular Noise Amplitude

To test the influence of stochastic fluctuations in the models, we change the volume from 40 to 1400 as shown in figure 4.26C, F. For each value of Ω, 24

simulations are performed and the average S/N value is plotted in figure 4.26C, F. The error bar is the standard deviation of the results of the 24 simulations.

Time-scale Optimization

We sampled 360 points randomly in the time-scale space with a uniform distribution in the logarithmic domain over 1 order of magnitude for each parameter ($TS_\alpha \in [\frac{1}{3}, 3]$ for $\alpha = \{QS, QM, R5, R6, R7, At\}$). For each time-scale set, we performed 8 long simulations (1000 units of time for relaxation, 2000 units of time of recording, volume of 200, $N_{max} = 150$) and average their signal to noise ratio.

Module Optimization

We use the same procedure than the time-scale optimization: we randomly sampled the parameter space in the log10 domain of the additional module (oscillator in system 4 and throttle in system 5). We allowed one order of magnitude around the nominal values (see table C.5). For each set (2000 for system 4 and 1000 for system 5), we run a stochastic simulation with the Langevin model and recorded the S/N value as in the time-scale optimization.

C.2 Algorithms Used for the Analysis of the Systems

C.2.1 Patterning and Neighbor Density Analysis

Analysis of distances between pairs of committed and uncommitted cells was performed to identify patterning between the two cell-types. For a given committed or uncommitted reference cell, the ratio of committed to uncommitted neighbors at a given distance was calculated for all distances. We define $p(c)_{i,d,t}$ as the observed fraction of committed cell neighbors at distance d for the ith cell at time t. The normalized score $Z(c)_{i,d,t}$ for that observation is then described by

$$Z(c)_{i,d,t} = \frac{p(c)_{i,d,t} - \mu_t}{\sigma_{i,t}} \tag{C.29}$$

where μ_t represents the overall fraction of committed cells at time t and $\sigma_{i,t}$ describes the standard deviation of the observed probability given by the stan-

dard form:

$$\sigma_{i,t} = \sqrt{\frac{\mu_t(1-\mu_t)}{n_{i,t}}} \tag{C.30}$$

where $n_{i,t}$ is the number of total neighbors observed for the i^{th} cell at time t. Normalized Z-scores are combined into an average Z-score, $\bar{Z}(c)_d$, for each distance value,

$$\bar{Z(c)}_d = \sum_i \sum_t Z(c)_{i,d,t}/\sqrt{N_d} \tag{C.31}$$

where N_d is the total number of sample Z-scores $Z(c)_{i,d,t}$ for each distance.

C.2.2 RS-HDMR Sensitivity Analysis

RS-HDMR is a tool to deduce non-linear interactions between a set of inputs and an output [201]. In this application, input-output relationships are defined as the effects of parametric variation on hysteresis of the UPC module response to fluctuations in population density. More specifically, we define hysteresis as the difference between the forward and reverse response values of population density (ρ_p), where the UPC module's output ($[pA2.Rec1.AI1]$) is 50% of maximum, or (max output + min output)/2. We focus on absolute levels of hysteresis rather than normalizing to the average population density threshold, because we focus on systems with similar average thresholds and consider absolute changes in population density to be a relevant optimization feature of our system in the context of its biomedical application.

We performed RS-HDMR sensitivity analysis on datasets describing neighborhoods of parameter space around optimal parameter vectors obtained from GA runs. 150 total optimal parameter vectors were obtained from 150 independent GA runs. Random sampling around each optimal parameter vector was from a normal distribution $N(\mu, \sigma)$ where μ is the optimized parameter's value and $\sigma = \mu/20$. Empirical evidence suggested that significantly broader sampling resulted in too many parameter sets that did not yield QS behavior. Sample size of the training set was 2000, and the resultant model was tested on unsampled points for validation purposes.

RS-HDMR describes the independent and cooperative effects of p parameters $\mathbf{k} = (k_1, k_2, ...k_p)$ on an output, $y = f(\mathbf{k})$, in terms of a hierarchy of RS-HDMR component functions:

$$f(\mathbf{k}) = f_0 + \sum_{i=1}^{p} f_i(k_i) + \sum_{1 \le i < j \le p} f_{ij}(k_i, k_j) + ... + f_{12...p}(k_1, k_2, ...k_p) \tag{C.32}$$

Here f_0 represents the mean value of $f(\mathbf{k})$ over the sample space, the first-order component function $f_i(k_i)$ describes the generally non-linear independent contribution of the i^{th} input variable to the output, the second-order component function $f_{ij}(k_i, k_j)$ describes the pairwise cooperative contribution of k_i and k_j, and further terms describe higher order cooperative contributions. In this work, we only consider the first-order RS-HDMR component functions in order to perform efficient high-throughput analyses of local parameter "neighborhoods." We approximate RS-HDMR component functions as weighted orthonormal basis functions, which take the following form:

$$f_i(k_i) \approx \sum_{r=1}^{n_o} \alpha_r^i \varphi_r^i(k_i) \tag{C.33}$$

where n_o is an integer (generally ≤ 3 for most applications), $\{\alpha\}$ are constant weighting coefficients to be determined, and the basis functions $\{\varphi\}$ are optimized from the distribution of sample data points to follow conditions of orthogonality [201]. Basis functions are approximated here as non-linear polynomials, where

$$\varphi_1^i(k_i) = a_1 k_i + a_0 \qquad \varphi_2^i(k_i) = b_2 k_i^2 + b_1 k_i + b_0 \qquad \varphi_3^i(k_i) = c_3 k_i^3 + c_2 k_i^2 + c_1 k_i + c_0 \tag{C.34}$$

The coefficients a_0,a_1,b_0,...c_3 are calculated using Monte Carlo integration under constraints of orthogonality, such that when integrated over all data points,

$$\int \varphi_r(k)dk \approx 0 \;\; \forall\, r \qquad \int \varphi_r^2(k)dk \approx 1 \;\; \forall\, r \qquad \int \varphi_p(x)\varphi_q(x)dx \approx 0 \;\; (p \neq q) \tag{C.35}$$

Optimal basis functions are weighted by coefficients (α_r^i), which are calculated from least-squares regression. Only inputs and their respective component functions measured as significant by the statistical F-test were included in RS-HDMR expansions [205]. The resultant expansion in Eq. C.32 serves both as a predictive model of network response due to its parametric interactions and as a statistical representation of the underlying system.

The relative strength of response to parametric changes can be quantitatively determined through sensitivity analysis based on the respective RS-HDMR component functions. A global sensitivity analysis may be calculated

from the RS-HDMR expansion through a decomposition of the total variance σ^2 of an output species, $f(\mathbf{k})$, into hierarchical contributions from the individual RS-HDMR component functions. The RS-HDMR expansion may be given in terms of the n_p significant component functions $g_l(l = 1, 2, 3, ...n_p)$, such that

$$f(\mathbf{k}) - f_0 = \sum_{i=1}^{n} f_i(k_i) + ... + \epsilon = \sum_{l=1}^{n_p} g_l + \epsilon \tag{C.36}$$

where ϵ represents any residual error of the model. The total variance, σ^2, of an output variable $f(\mathbf{k})$ is then defined as follows, with integration over all data points:

$$\begin{aligned} \sigma^2 &= \int \left[f(\mathbf{k}) - f_0\right]^2 dk = \int \left[\sum_{l=1}^{n_p} g_l + \epsilon\right]^2 dk \\ &= \int \left[\sum_{l=1}^{n_p} (g_l)^2 + \sum_{l=1}^{n_p}\sum_{i\neq l}^{n_p} (g_l)(g_i) + \epsilon^2\right] dk \end{aligned} \tag{C.37}$$

When the input variables are sampled independently of one another, the RS-HDMR component functions are calculated to be mutually orthogonal. However, under conditions of correlation among input variables, the orthogonality of distinct component functions may not be strictly upheld. Consequently, the output variance σ^2 can be decomposed in terms of independent and correlated contributions of the RS-HDMR component functions, where the correlated contributions are described as the summed pairwise-covariances of the individual component functions ($\int \sum_{l=1}^{n_p} \sum_{i\neq l}^{n_p} (g_l)(g_i)$).

The sensitivity indices, $S_i(i = 1, 2, ..., n_p)$, are then defined as the portion of the total variance σ^2 represented by the l^{th} component function out of n_p total number of functions. The relationship between sensitivity indices and the output variance σ^2 is given by the following:

$$S_i = \frac{1}{\sigma^2} \int \left[(g_l)^2 + \sum_{i\neq l}^{n_p} (g_l)(g_i)\right] dk \tag{C.38}$$

The magnitudes of S_i were analyzed to quantify the relative strength of connections between hysteresis and the model parameters (Fig. 4.29D and Fig. 4.30).

C.3 Supplementary Results

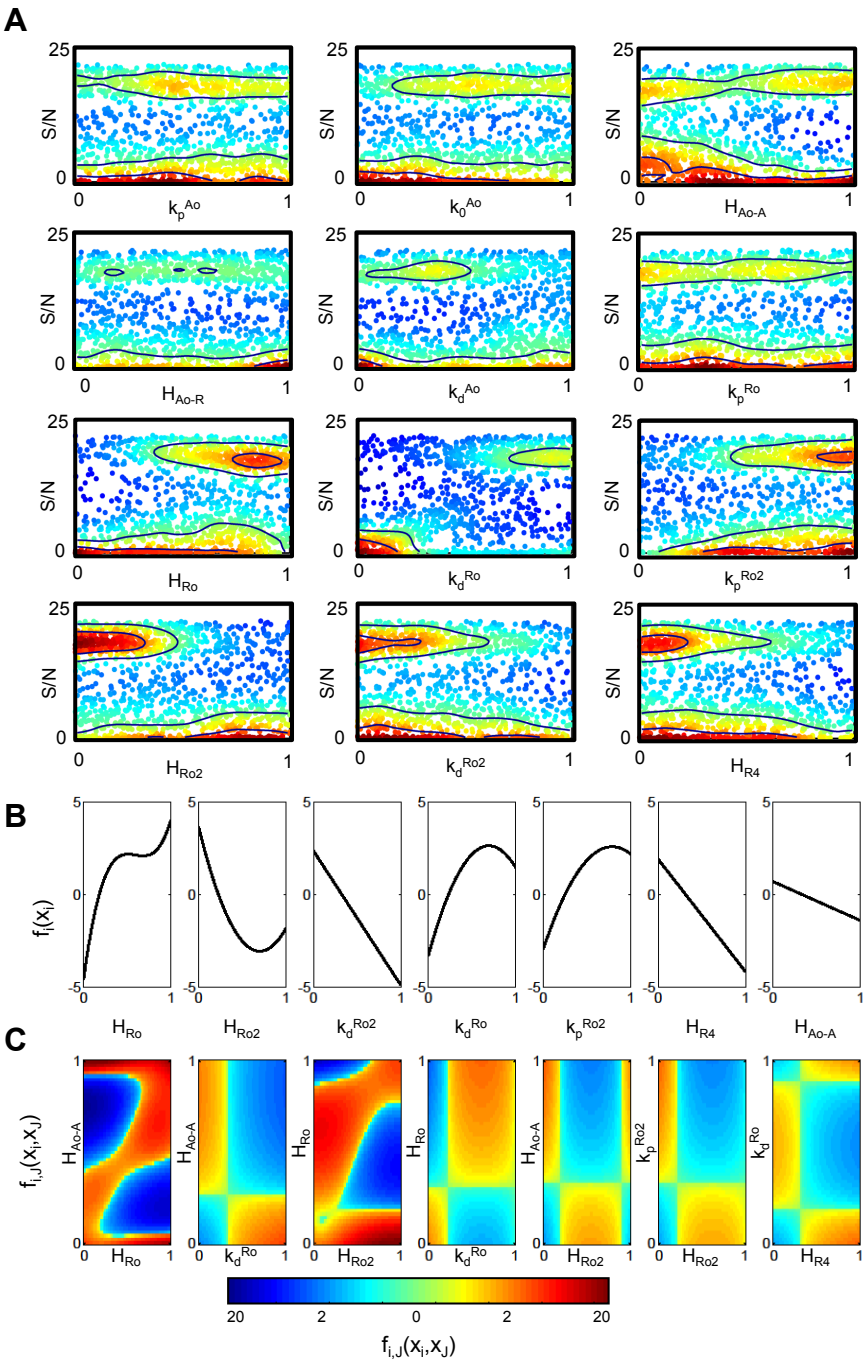

Figure C.1: RS-HDMR analysis of the oscillator module. (A) Oscillator rate contstants (see Table C.5) were randomly varied across one order of magnitude around initial values (linearly normalized to range [0,1], uniform distribution in the log space) to produce roughly 2000 parameter sets. Simulations of each parameter set yielded a corresponding S/N value, which is plotted here as a function of the individual parameters. Each point represents an individual parameter set. Warmer colors indicate higher point density; contour lines also indicate point density. (B-D) RS-HDMR parametric sensitivity analysis of data in *A*, describing the influence of parameter variation on observed S/N. (B) RS-HDMR first-order component functions, in order of decreasing sensitivity index S_i. (C) Second-order RS-HDMR component functions in order of decreasing sensitivity index S_l.

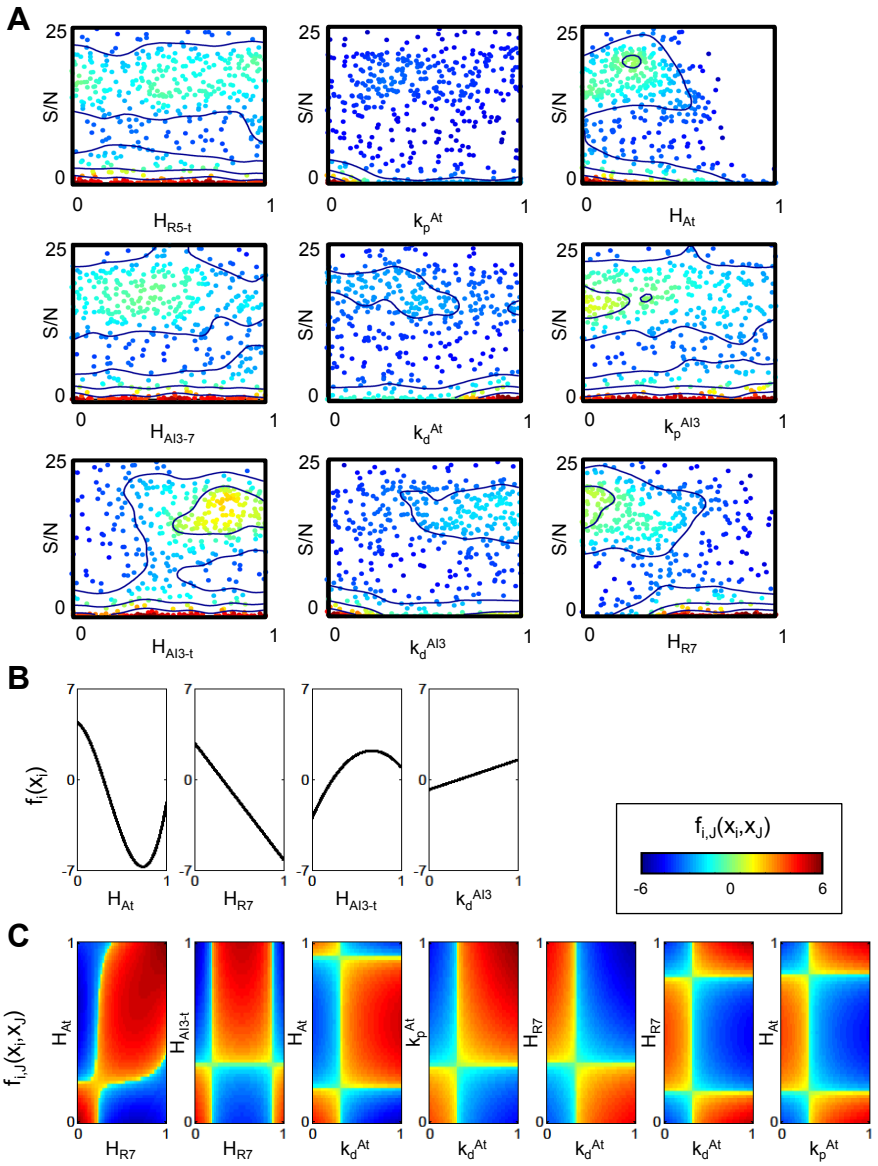

Figure C.2: RS-HDMR analysis of the throttle module. (A) Throttle rate contstants (see Table C.5) were randomly varied across one order of magnitude around initial values (linearly normalized to range [0,1], uniform distribution in the log space) to produce roughly 1000 parameter sets. Simulations of each parameter set yielded a corresponding S/N value, which is plotted here as a function of the individual parameters. Each point represents an individual parameter set. Warmer colors indicate higher point density; contour lines also indicate point density. (B-D) RS-HDMR parametric sensitivity analysis of data in A, describing the influence of parameter variation on observed S/N. (B) RS-HDMR first-order component functions, in order of decreasing sensitivity index S_i. (C) Total RS-HMDR sensitivity indices S_i^T.

Marc HAFNER

Personal Information

Place and Date of Birth:	Lausanne, Switzerland — 5 Nov 1982
Address:	Ramiers 16, 1022 Chavannes, Switzerland
Phone:	+41 78 759 7118
email:	marc.hafner@a3.epfl.ch

Education

May 2007 - Dec 2010	PhD thesis at Ecole Polytechnique Fédérale de Lausanne in collaboration University of Zürich, entitled *'Quantitative Analysis of Robustness in Systems Biology: Combining Global and Local Approaches'*, supervised by Prof. H. Koeppl, Prof. A. Wagner and Prof. M. Hasler, funded by SystemsX.ch IPhD grant.
Sept 2006 - March 2007	Master project at Ecole Polytechnique Fédérale de Lausanne, entitled *'The Polyglutamine aggregation: Structural or kinetic threshold?'*, supervised by Prof. P. De Los Rios.
Oct 2000 - March 2007	Bachelor and Master in Physics with specialization in biophysics, Ecole Polytechnique Fédérale de Lausanne.

Research Experience

Sept - Nov 2009	Scientific visit at Ron Weiss' laboratory, Massachusetts Institute of Technology. Further visits in 2010 and on-going collaboration.
April 2008	Visits and collaboration with Prof. Rodolphe Sepulchre, University of Liége, Belgium. Further visits in 2009.
Nov 2007	Visit of Dr. Ralf Steuer in the group of Prof. Markus Kollman at the Humboldt University, Berlin, Germany.
Aug - Oct 2005	Internship at the Institute of applied radiophysics, University Hospital, Lausanne, development of a protocol for the positioning of the patient in the prostate treatment.

Publications

- M. Milles*, M. Hafner*, E. Sontag, S. Subramanian, P. Purnick, N. Davidsohn and R. Weiss, *Design of a large scale synthetic biological circuit to maintain artificial tissue homeostasis*, in preparation. *equal contribution.
- E. Zamora-Sillero, M. Hafner, A. Ibig, J. Stelling and A. Wagner, *Effcient characterization of high-dimensional parameter spaces for systems biology*, in BMC Systems Biology (2010), submitted.
- M. Hafner, T. Petrov, J. Lu and H. Koeppl, *Rational design of robust biomolecular circuits: from specification to parameters*, in Design and Analysis of Bio-Molecular Circuits (2011), edited by H. Koeppl, D. Densmore, M. di Bernardo and G. Setti, Springer, New York. In press.
- M. Hafner and H. Koeppl, *Stochastic Simulations in Systems Biology*, in Handbook of Research on Computational Science and Engineering: Theory and Practice (2010), edited by J. Leng and W. Sharrock, IGI Global In press.
- D. Gonze, M. Hafner, *Positive feedbacks contribute to the robustness of the cell cycle with respect to molecular noise*, in Advances in the Theory of Control, Signals and Systems, with Physical Modelling, Lecture Notes in Control and Information Sciences, Vol. 407 (2011), pp. 283-295 edited by J. Lévine, P. Müllhaupt, Springer, Berlin.
- M. Hafner, P. Sacré, L. Symul, R. Sepulchre and H. Koeppl, *Multiple feedback loops in circadian cycles: Robustness and entrainment as selection criteria*, in Proceedings of the Seventh International Workshop on Computational Systems Biology (2010), edited by M. Nykter, P. Ruusuvuori, C. Carlberg and O. Yli-Harja, pp. 43-46.
- M. Hafner, H. Koeppl, M. Hasler and A. Wagner, *'Glocal' robustness analysis and model discrimination for circadian oscillators*, in PLoS Computational Biology (2009), vol. 5(10), e1000534.
- M. Hafner, H. Koeppl and A. Wagner, *Evolution of feedback loops in oscillatory systems* in Proceedings of the Third International Conference on Fundations of Systems Biology in Enginnering (2009), pp. 157-160, http://arxiv.org/abs/1003.1231.
- H. Koeppl, M. Hafner, and V. Danos, *Rule-based modeling for protein-protein interaction networks - the cyanobacterial circadian clock as a case study* in Proceedings of the Sixth Workshop on Computational Systems Biology (2009). Best paper award.
- H. Koeppl, M. Hafner, and R. Steuer. *Semi-quantitative stability analysis constrains saturation levels in metabolic networks* in Proceedings of the Sixth Workshop on Computational Systems Biology (2009).

Teaching Experience

Feb - June 2009	Supervision of the Master Thesis of Laura Symul: *'Circadian cycle: robustness and entrainment'*
Sept 2008 - Jan 2009	Teaching assistant for Master level lecture of Prof. Hasler, Dynamical systems theory for engineers
Sept 2007 - June 2008	Teaching assistant for Bachelor level lecture of Prof. Hasler, Circuits and Systems

Working Experience

2007 - Present	Work on the World Cups, World Championships and 2008 Olympics Games with the International Rowing Federation as part of the media team.
Apr 2006 - Oct 2007	Work with an independent trader: Simulation of investment on the exchange rate.
2001 - 2007	Substitute teacher at the elementary school (Lausanne)

Skills

Languages	French, mother tongue English, fluent, very good written knowledge (C2 of European Language Scale) German, fluent, good written knowledge (B2)
Computer	Windows, Linux - Office - programming in C++, Fortran, Igor, Matlab, Perl - Photoshop

Interests

Rowing	Competition at an international level from 1999 to 2007 (World Championships). Coach and technical director at the rowing club Lausanne-Sports Aviron (from 2008). Coach at the Junior World Championships 2010.
Photography	Small photo sessions for the International Rowing Federation during competitions. Illustration of theater play 'Rame' played at Théâtre de Vidy, Lausanne.
Leisure	sport, music, piano