Chapter 17

Computational Methods to Model Persistence

Alexandra Vandervelde, Remy Loris, Jan Danckaert, and Lendert Gelens

Abstract

Bacterial persister cells are dormant cells, tolerant to multiple antibiotics, that are involved in several chronic infections. Toxin–antitoxin modules play a significant role in the generation of such persister cells. Toxin–antitoxin modules are small genetic elements, omnipresent in the genomes of bacteria, which code for an intracellular toxin and its neutralizing antitoxin. In the past decade, mathematical modeling has become an important tool to study the regulation of toxin–antitoxin modules and their relation to the emergence of persister cells. Here, we provide an overview of several numerical methods to simulate toxin–antitoxin modules. We cover both deterministic modeling using ordinary differential equations and stochastic modeling using stochastic differential equations and the Gillespie method. Several characteristics of toxin–antitoxin modules such as protein production and degradation, negative autoregulation through DNA binding, toxin–antitoxin complex formation and conditional cooperativity are gradually integrated in these models. Finally, by including growth rate modulation, we link toxin–antitoxin module expression to the generation of persister cells.

Key words Modeling, Toxin–antitoxin, Persister, ODE, Stochastic, Gillespie

1 Introduction

Biological systems are typically very complex and their functioning is often not fully understood. Traditionally, biologists have used qualitative methods to understand these systems. However, as the behavior of biological systems is often non-intuitive, mathematical models can be a valuable tool to study their characteristics quantitatively. Such models have successfully been used to study the eukaryotic cell cycle [1, 2], the heart [3], and transmission of infectious diseases [4], just to name a few examples. Recently, several groups have applied mathematical modeling to investigate persistence [5–15]. Bacterial persister cells are subpopulations of rare, slow-growing cells exhibiting multidrug tolerance even though the rest of the population is susceptible to the applied antibiotics [16]. Persisters have been found to play a role in several human diseases, for example cystic fibrosis, tuberculosis, and candidiasis [17].

Jan Michiels and Maarten Fauvart (eds.), *Bacterial Persistence: Methods and Protocols*,
Methods in Molecular Biology, vol. 1333, DOI 10.1007/978-1-4939-2854-5_17,

Different molecular pathways leading to the formation of a persister cell have been proposed (reviewed in [18]). One such pathway involves a hierarchical cascade, including increased concentrations of ppGpp (the signaling nucleotide regulating the stringent response), PolyP (inorganic polyphosphate), protease Lon and activation of toxin–antitoxin modules [19]. In an alternative mechanism, persister cells are formed due to elevated free toxin levels, caused by the dynamics of toxin–antitoxin module expression [9]. The latter mechanism has extensively been translated to mathematical models [10–15], and such models will be the topic of this chapter.

As mentioned above, several important routes towards persister generation involve the activity of toxin–antitoxin modules. Toxin–antitoxin modules are small genetic elements, omnipresent in the genomes of bacteria and archaea [20]. Most toxin–antitoxin modules code for two components: a toxic protein that is able to inhibit cell growth and an antitoxin that can antagonize this toxic activity. Five types of toxin–antitoxin modules have currently been described, depending on the nature of the antitoxin and the mode of neutralization. In type I toxin–antitoxin modules, the antitoxin is an antisense RNA, which negatively regulates toxin translation [21]. Both the antitoxin and the toxin are proteins in type II toxin–antitoxin modules, and neutralization occurs through the formation of a non-toxic toxin–antitoxin complex [22–25]. Type III toxin–antitoxin modules consist of a toxic protein and an RNA antitoxin like type I modules, yet in this case, binding of the antitoxin to the toxin ensures the neutralization instead of gene expression regulation [26]. In type IV toxin–antitoxin modules, toxin and antitoxin are proteins as in type II, however, the antitoxin interacts directly with the toxin target to protect it from the toxin's activity, instead of interacting with the toxin. Finally, in type V toxin–antitoxin modules, the antitoxin is again a protein, which antagonizes the toxin by specifically cleaving its mRNA [27].

Currently, all mathematical modeling papers focus on archetypical two-component type II toxin–antitoxin modules, for which the link with persistence is best established [28–30]. Such toxin–antitoxin modules are polycistronic operons in which the gene for the toxin is preceded by the gene for the antitoxin. Exceptions to this genetic make-up exist, as operons with an inverted genetic organization [31, 32] and three-component type II toxin–antitoxin modules [33] have been discovered. In typical type II toxin–antitoxin modules, the toxin is either a monomer, like RelE and HipA [34, 35], or a homodimer, like CcdB and MazF [36, 37]. The antitoxin is typically a dimer with a DNA-binding domain and an intrinsically disordered toxin-binding domain. Therefore, the antitoxin has a shorter in vivo lifetime than the toxin as it is vulnerable to degradation by cellular proteases. Type II toxin–antitoxin modules are further regulated at the transcriptional level by the antitoxin and the non-toxic complexes.

For many type II toxin–antitoxin modules like *phd/doc*, *ccdAB*, and *relBE*, this regulation involves conditional cooperativity [38–40]. In this mechanism, the antitoxin alone has a low affinity for its binding site on the operator DNA. At low intracellular toxin:antitoxin ratios, the toxin acts as a corepressor for the antitoxin by forming a toxin–antitoxin complex with a higher affinity for the DNA. At high toxin:antitoxin ratios, however, transcription and translation of the toxin–antitoxin module has to resume to maintain a viable toxin:antitoxin ratio. In this case, the toxin functions as a derepressor for the antitoxin, often by forming a second, non-repressing type of toxin–antitoxin complex [38, 41]. Even within type II toxin–antitoxin modules, conditional cooperativity is not universal. For example, the MqsR toxin of the *mqsRA* toxin–antitoxin module destabilizes the binding of antitoxin MqsA to DNA [42], only functioning as a derepressor in the autoregulation. Finally, the number of binding sites for the antitoxin on the operator varies depending on the toxin–antitoxin module, from two in the *phd/doc* and *relBE* operon to eight in the *ccdAB* operon [40, 43, 44].

In this work, we will present an overview of numerical methods to describe toxin–antitoxin modules and persistence. First, we will distinguish between deterministic and stochastic modeling approaches. Toxin–antitoxin dynamics involve biochemical processes such as transcription and translation, which are intrinsically noisy due to the low copy number of DNA and mRNA [45–47]. Moreover, as the intracellular free toxin levels are generally very low, limited to a few proteins, stochastic effects are likely to play an important role specifically in toxin–antitoxin modules. Therefore, next to the deterministic modeling approach involving Ordinary Differential Equations (ODEs), we will introduce two stochastic numerical methods incorporating unavoidable noise and randomness, being Stochastic Differential Equations (SDEs) and the Gillespie algorithm. We will start by illustrating these numerical methods using a very simple system, including only protein production and degradation, then, we will step by step introduce the characteristics of toxin–antitoxin modules in these models. As a first characteristic, we will include the negative transcriptional regulation due to DNA binding in this system. Then, we will explicitly model the production of toxin and antitoxin separately and describe the formation of toxin–antitoxin complexes. As a next step, we include conditional cooperativity in the autoregulation of the operon. Finally, we will consider how toxin–antitoxin modules affect the whole cell, as the presence of free toxins can slow down cell growth by interfering with the basic metabolism. Including such growth rate modulation leads to interesting behavior at the level of the cell as well as on the population level. Depending on the model and parameters, two different populations can emerge, one growing normally, and one with a severely

Table 1
Model parameters for the bacteriophage P1 *phd/doc* toxin–antitoxin module [14]

Parameter	Meaning	Value	Units
ζ_U	Unbound mRNA transcription rate	0.116086	s^{-1}
ζ_B	Bound mRNA transcription rate	0	s^{-1}
ρ_A	Antitoxin translation rate	$0.137\ \zeta_U/d_m$	s^{-1}
ρ_T	Toxin translation rate	$0.053\ \zeta_U/d_m$	s^{-1}
V	Volume *E. coli* cell	3.612e+8	m^3
d_m	mRNA decay rate	0.00203	s^{-1}
$d_c = d_T = d_{AT} = d_{TAT}$	Decay rate due to cell cycle dilution	2.8881e−4	s^{-1}
d_A	Antitoxin decay rate	$4 \cdot d_c$	s^{-1}
α_C	Binding of antitoxin and toxin	8.79e+6	$M^{-1}s^{-1}$
θ_C	Unbinding of antitoxin and toxin	5.3e−5	s^{-1}
α_{AT}	Binding of complex (AT) to binding site on the operator	9625	$M^{-1}s^{-1}$
θ_{AT}	Unbinding of complex (AT) from a binding site on the operator	0.0028875	s^{-1}

decreased cellular growth rate, the persister cells. We describe how a two-state model can be used to model populations in which cells can switch from a normal, growing stage to a persister stage and back. Throughout this chapter, we will use a parameter set based on experimental data for the bacteriophage P1 *phd/doc* toxin–antitoxin module (Table 1), as introduced in ref. [14].

2 Numerical Methods

In this section, we will introduce ODEs as a means to model TA systems in a deterministic manner, and we will introduce two stochastic numerical methods, being SDEs and the Gillespie algorithm. We illustrate these different methods using a toy model, based on the simplest approximation for a toxin–antitoxin module. We assume that the toxin–antitoxin complex AT is produced as a single entity.

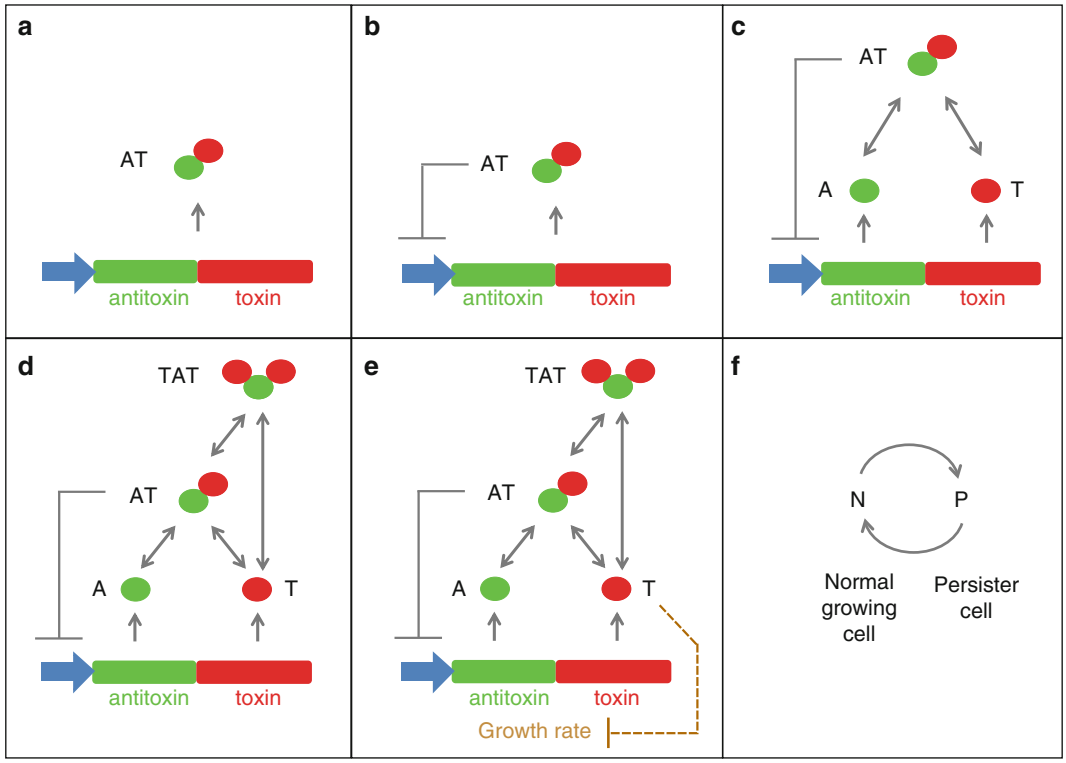

Fig. 1 Overview of the different toxin–antitoxin module topologies modeled in this chapter. (**a**) Direct production of toxin–antitoxin complex AT. (**b**) Direct production of complex AT and negative feedback through DNA binding. (**c**) Production of antitoxin A and toxin T, complex formation and negative feedback through DNA binding. (**d**) Regulation of toxin–antitoxin modules including conditional cooperativity. (**e**) Regulation of toxin–antitoxin modules including conditional cooperativity and growth rate modulation. (**f**) Modeling populations of normal cells and persister cells. Due to its susceptibility for degradation by cellular proteases, the antitoxin A has a shorter in vivo lifetime than the toxin T and complex AT and TAT, which decay with a rate corresponding to dilution due to cell division (Table 1)

This would be a good description for a hypothetical toxin–antitoxin module in which the antitoxin A and the toxin T are translated at equal rates, complex formation quickly reaches an equilibrium and the complex does not bind to the DNA operator site. As such, we obtain a genetic circuit consisting of AT being produced and degraded at a rate corresponding to dilution due to cell division. In the sketch shown in Fig. 1a, we illustrate this simple system. The genes on the DNA are transcribed into mRNA which in turn is translated into the corresponding complex AT. Throughout this paper we will neglect the intermediate step of mRNA production and assume that it happens fast enough such that it can be modeled as a modified translation rate of the protein itself. This model is not intended to provide an accurate description of TA systems, but is merely constructed to introduce the numerical methods and it will serve as a basis to construct more adequate toxin–antitoxin models in subsequent sections.

2.1 Deterministic Modeling Using ODEs

The toy system in Fig. 1a can be modeled in a deterministic way by using a single ODE. A differential equation is a mathematical equation which involves an unknown function, here AT, and its derivatives. In an ODE, the unknown function (the dependent variable AT) depends only on a single independent variable, in our case, time t [48]. The following ODE describes the time evolution of the number of AT complexes:

$$\frac{dAT(t)}{dt} = \rho_{AT} - d_{AT}AT(t), \tag{1}$$

The first term on the right-hand side, ρ_{AT}, is the average rate at which the complex AT is created through the process of transcription and translation from the DNA template. The second term on the right-hand side, $d_{AT}AT(t)$, models the average degradation rate of AT complexes and depends linearly on the amount of complex $AT(t)$ in the system at that time.

In order to numerically solve an ODE with a given initial value for the variables (here: AT), various methods have been developed. Such a numerical method typically uses the value of the variables at time t_0, called the initial condition, to make a prediction of those variables at a later time $t_0 + \delta t$, where δt is a chosen discrete time step. The simplest of such explicit methods is the Euler method [49], which is a first-order method, meaning that the error at a given time is proportional to the step size δt. Assume that we try to model the time evolution of a single protein X described by the following general ODE:

$$\frac{dX(t)}{dt} = F(t, X(t)), \tag{2}$$

where $F(t, X(t))$ is a function of time t and $X(t)$, with as initial condition $X(t = t_0) = X_0$. In order to use the Euler method to numerically solve this ODE, we first choose a discrete time step δt, which needs to be small enough to be numerically stable and to be able to capture the relevant dynamics of the ODE. The Euler method now makes a first prediction for time $t_1 = t_0 + \delta t$ after which $X(t_1)$ is used for the next prediction. In general, the value of X at time $t_{n+1} = t_n + \delta t$ is given by

$$X(t_{n+1}) = X(t_n) + \delta t F(t_n, X(t_n)) \tag{3}$$

For our system under study, Eq. 1, this numerical Euler method gives $AT(t_{n+1}) = AT(t_n) + \delta t(\rho_{AT} - d_{AT}AT(t_n))$. This method can easily be generalized to a system of multiple ODEs. As mentioned the Euler method is only a first-order method, which makes it more prone to numerical instabilities and errors. To overcome these drawbacks, this Euler method is often used as a basis to construct

more complicated and accurate methods. Here we will only introduce the Euler–Heun method [49], a quite robust second-order method which we will use throughout this paper. When using the Euler–Heun method, one first calculates an intermediate value $X^*(t_{n+1})$ after which this value and $X(t_n)$ are used to make an improved prediction of the actual value $X(t_{n+1})$. For the general ODE system, Eq. 2, this is done as follows:

$$X^*(t_{n+1}) = X(t_n) + \delta t F(t_n, X(t_n)) \tag{4}$$

$$X(t_{n+1}) = X(t_n) + \frac{\delta t}{2}(F(t_n, X(t_n)) + F(t_{n+1}, X^*(t_{n+1}))) \tag{5}$$

In Fig. 2a, we show the time evolution of the complex AT using the Euler–Heun method with $\delta t = 10^{-2}$ s and $AT(t_0) = 0$. The creation rate ρ_{AT} is chosen to be equal to the toxin creation rate ρ_T, given in Table 1. This is a good approximation if the complex formation between the antitoxin A and the toxin T is constantly near equilibrium, and if there is more A present in the system than T. The degradation rate $d_{AT} = d_c = \ln(2)/(40 \cdot 60\ \text{s})$ is assumed to be solely due to dilution caused by cell division every 40 min, *see* also Table 1. An example of a simple MATLAB code to numerically solve Eq. 1 can be found in Appendix 1. One can see in Fig. 2a that the number of complexes AT increases monotonically to a fixed level AT_{SS}. After the initial transient behavior, the properties of the system no longer change in time ($AT = AT_{SS}$), which is called a steady state solution. AT_{SS} can easily be calculated analytically from Eq. 1 by setting $\frac{dAT(t)}{dt} = 0$, yielding

$$AT_{SS} = \frac{\rho_{AT}}{d_{AT}}. \tag{6}$$

2.2 Stochastic Modeling Using SDEs

In the previous section, we introduced how to numerically solve an ODE equation using either a first-order Euler method or a second-order Euler–Heun method. This approach allows to model how the protein number evolves in time in a deterministic way. In other words, as long as one uses the same initial condition, every numerical simulation will provide you with exactly the same outcome for the time evolution of the protein level. In reality, however, most biological processes are to a certain extent stochastic. In toxin–antitoxin systems, noise may originate from the transcription and translation processes [45–47] and from the interactions of the free toxins, because they are generally present in very small amounts but have an important impact on the growth rate of the cell.

One way to introduce noise into the system is to use a SDE [50, 51], which is a differential equation in which one or more of the terms describes a stochastic process. In this work, we will limit ourselves to consider random white Gaussian noise. The SDE with white Gaussian noise that we use is the following:

$$\frac{dX(t)}{dt} = F(t, X(t)) + \eta(t), \tag{7}$$

similar as Eq. 2, where the extra last term is often called a Langevin noise term and $\eta(t)$ is the actual noisy process. Each sample of $\eta(t)$ has a normal distribution with zero mean, such that the signal is Gaussian white noise. The uncorrelated zero-mean stochastic term $\eta(t)$ is thus described by the correlation term $\langle \eta(t+\tau)\eta(t)\rangle = D\delta(\tau)$ where D is a constant diffusion noise strength. As we consider D to be a constant, the system is said to be subject to additive noise. In the case of multiplicative noise, extra care must be taken to solve the SDE [50, 51].

In order to solve the SDE, we use a similar Euler–Heun method, now adjusted to include the Langevin noise term:

$$X^*(t_{n+1}) = X(t_n) + \delta t F(t_n, X(t_n)) + \sqrt{D\delta t}\, r, \tag{8}$$

$$\begin{aligned} X(t_{n+1}) = X(t_n) + \frac{\delta t}{2}\left(F(t_n, X(t_n)) + F(t_{n+1}, X^*(t_{n+1}))\right) \\ + \frac{1}{2}\sqrt{D\delta t}\, r, \end{aligned} \tag{9}$$

where r is a random number taken from the standard normal distribution with standard deviation equal to 1. The presence of the time step δt under the square root in front of the noise term calls for an explanation. The analytical derivation is a bit technical, and for that we refer the reader to the specialized literature [51, 52]. Intuitively, it can be understood that, as one, e.g., decreases the time step, one should also rescale (decrease) the strength of the noise, if not, one would be injecting the same amount of noise in a shorter time interval which corresponds to an effectively higher noise level.

In Fig. 2b, we show the time evolution of the level of complex AT using the Euler–Heun method to solve the following SDE, corresponding to ODE Eq. 1

$$\frac{dAT(t)}{dt} = \rho_{AT} - d_{AT} AT(t) + \eta(t), \tag{10}$$

with $\delta t = 10^{-2}$ s, $\mathrm{AT}(t_0) = 0$ and $D = 25$. One can see that the result is essentially the same as the deterministic evolution (Fig. 2a) with a small noisy ripple superimposed on it.

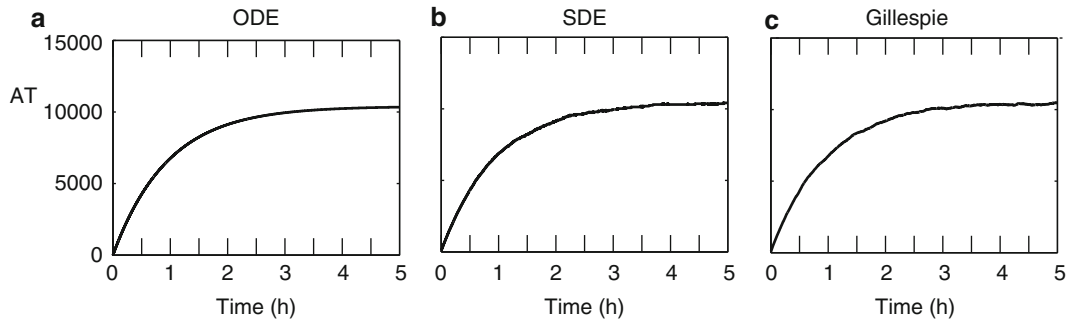

Fig. 2 Time evolution of the level of complex AT for a hypothetical toxin–antitoxin model describing direct production of complex AT (Fig. 1a). The system given by Eq. 1 was simulated for 5 h. The graphs show the results for a single cell. (**a**) Numerical simulation of the ODE using the Euler–Heun method. (**b**) Numerical simulation of the SDE given by Eq. 10 using the Euler–Heun method, $D = 25$. (**c**) Numerical simulation of stochastic equations using the Gillespie algorithm. Parameters are given in Table 1, with $\rho_{AT} = \rho_T$ and $d_{AT} = d_c$

2.3 Stochastic Modeling Using the Discrete Gillespie Method

The previous two methods, ODE and SDE, are both continuous methods that average out many discrete biochemical reactions. Therefore, these differential equations, both deterministic and stochastic, heavily rely on bulk reactions requiring many interactions. In contrast, the Gillespie algorithm allows for a more accurate discrete and stochastic simulation of a system involving biochemical reactions [53]. This approach is especially necessary when few molecules are present in the system. In the Gillespie method every single reaction is explicitly simulated. The algorithm is based on the random occurrence of collisions of molecules with a certain probability. Only collisions between two molecules are considered as the probability of three molecules colliding is very low.

We will illustrate the Gillespie algorithm with the simple system described deterministically by the ODE Eq. 1. Instead of a differential equation we now consider explicitly every possible reaction and the probability per unit time that a specific reaction occurs. This probability of each reaction i is also called a propensity p_i. In our simple example, there are two reactions that can occur, AT can be degraded (1) and created (2):

$$(1) \quad AT \quad \rightarrow \emptyset, \tag{11}$$

$$(2) \quad \emptyset \quad \rightarrow AT, \tag{12}$$

with corresponding propensities:

$$(1) \quad p_1 = d_{AT} AT, \tag{13}$$

$$(2) \quad p_2 = \rho_{AT}. \tag{14}$$

The Gillespie algorithm consists of four steps that are iterated:

1. *Computation of the random time step:* the probability that any reaction occurs is the sum of the propensities

$$p_0 = \sum_i p_i,$$

with i the number of each reaction. Randomly choose the time of the next event, δt, out of the exponential distribution $p_0 \exp[-p_0 t]$ as follows:

$$\delta t = \frac{1}{p_0} \ln\left(\frac{1}{r_1}\right),$$

with r_1 a uniform random number between 0 and 1.

2. *Selection of a random reaction:* consider the simple system of two reactions. The probability that the next reaction to occur is reaction (1) is p_1/p_0, and likewise for reaction (2) the probability is p_2/p_0. In general, reaction i will occur with probability p_i/p_0. The reaction is randomly selected from this distribution of probabilities. In practice this is done by picking a second uniform random number r_2 between 0 and 1. The selected reaction K is then found by looking for the value of K for which the following inequality is satisfied:

$$\sum_{i=1}^{K-1} p_i < r_2 \, p_0 \leq \sum_{i=1}^{K} p_i. \tag{15}$$

3. *Update the populations based on the reaction chosen:* In our example, if reaction (1) was chosen, we decrease the number of AT complexes by one, while if reaction (2) was chosen, we increase the number of AT complexes by one.
4. *Update the current time:* the time t in the simulation is updated to time $t + \delta t$.

An example of a simple MATLAB code to numerically solve Eq. 1 using the Gillespie algorithm above can be found in Appendix 2. Figure 2c shows the time evolution of the complex AT as obtained by using the Gillespie method. The same initial condition is used. Notice that there is no need to define a fixed time step δt as it is randomly chosen based on the propensities. As the steady-state solution AT_{SS} of the system equation is quite high, it is no surprise that the deterministic ODE, the SDE, and the Gillespie method all give approximately the same result in this case.

3 Negative Feedback Through DNA Binding

In this section, we allow the complex AT to bind to its own operator DNA, as shown in Fig. 1b. When AT is bound to the DNA, no transcription can take place and the production of AT is halted. This provides a negative feedback control system: when little AT is present in the system, the probability to bind to the DNA is low, and more AT is produced; in the presence of lots of AT, DNA binding is more likely and AT production stops. Assuming there is only one site on the operator DNA for AT to bind to, one can see that this is a very discrete on/off process. Either AT is produced at the fastest rate possible (no AT bound) or the production rate is zero (AT bound). This already illustrates that when choosing a numerical method to solve this system, the discrete Gillespie method is most appropriate. However, using an ODE or SDE description has the advantage of simplicity and allows for certain analytical derivations. ODEs have been used to describe DNA binding effects in toxin–antitoxin systems in various works [8, 10–12]. When using an ODE or SDE, the discrete process of DNA binding is approximated by including a negative feedback term in the growth rate as follows:

$$\frac{dAT(t)}{dt} = \frac{\rho_{AT}}{1 + \frac{AT(t)^n}{K^n}} - d_{AT}AT(t) + \eta(t), \tag{16}$$

with $K = \theta_{AT}/\alpha_{AT}$ where $\theta_{AT}(\alpha_{AT})$ are the unbinding (binding) rates of AT from a binding site on the DNA operator site. The relevant parameters used can be found in Table 1. The results of including such negative feedback regulation in the ODE and SDE model are shown in Fig. 3a, b. The top panels depict the fraction D_F of the time that the operator site on the DNA is free (unbound). In comparison, without DNA binding (*see* Fig. 2), this fraction D_F would always be 1. As the operator site on the DNA is on average bound by AT for 90 % of the time, it is no surprise that the steady state level of AT is decreased to approximately 1000, which is about 10 % of the level without DNA binding events. The difference between the ODE and SDE description is still minor, although the noisy fluctuations around the steady state level are already more pronounced in Fig. 3b.

When using the Gillespie algorithm, every individual DNA binding/unbinding event is implemented explicitly. The set of reactions in this case becomes:

$$(1) \quad AT \quad \rightarrow \emptyset, \tag{17}$$

$$(2) \quad D_F \quad \rightarrow AT, \tag{18}$$

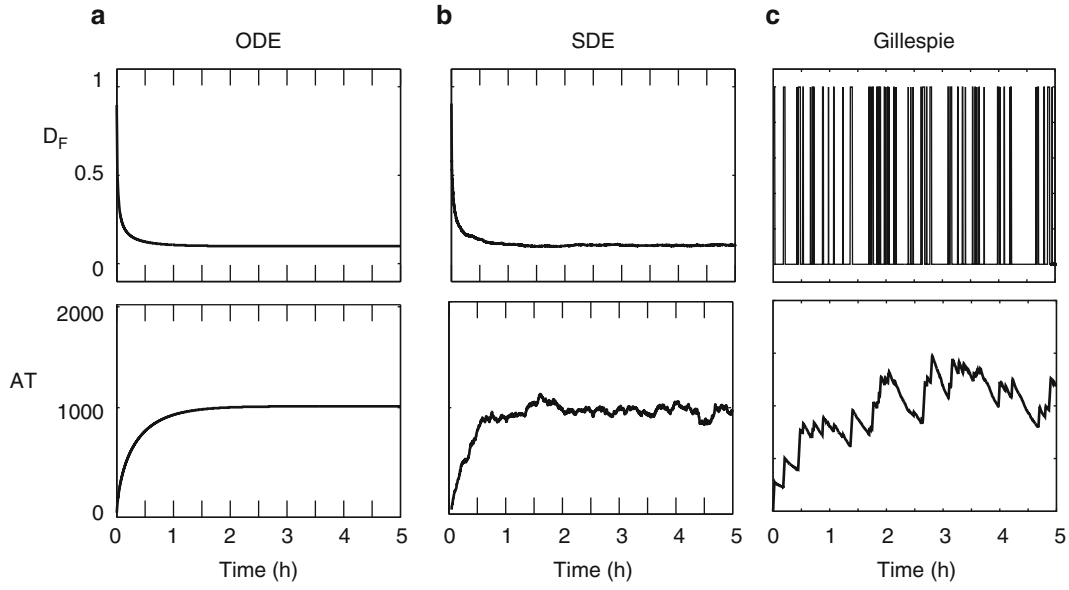

Fig. 3 Time evolution of the fraction D_F of time that the operator site on the DNA is unbound and the level of complex AT for a hypothetical toxin–antitoxin model describing direct production of complex AT and negative feedback through DNA binding (Fig. 1b). The systems given by Eq. 16 were simulated for 5 h. The graphs show the results for a single cell. (**a**) Numerical simulation of the ODE using the Euler–Heun method, $D = 0$. (**b**) Numerical simulation of the SDE using the Euler–Heun method, $D = 25$. (**c**) Numerical simulation of stochastic equations using the Gillespie algorithm. Parameters are given in Table 1

$$(3) \quad \text{Free operator DNA}(D_F = 1) + AT \rightarrow \text{Bound operator DNA}(D_F = 0), \tag{19}$$

$$(4) \quad \text{Bound operator DNA}(D_F = 0) \rightarrow \text{Free operator DNA}(D_F = 1) + AT, \tag{20}$$

with corresponding propensities:

$$(1) \quad p_1 = d_{AT} AT, \tag{21}$$

$$(2) \quad p_2 = \rho_{AT} D_F. \tag{22}$$

$$(3) \quad p_3 = \alpha_{AT} D_F AT. \tag{23}$$

$$(4) \quad p_4 = \theta_{AT}(1 - D_F). \tag{24}$$

In Fig. 3c the results obtained from the Gillespie algorithm are shown. The top panel clearly shows the discrete binding (unbinding)

events, which are accompanied by spikes in the production (degradation) of AT, which can be seen in the bottom panel. Such "bursty" dynamical behavior is typical for the evolution of mRNA levels in the presence of transcription factors that can bind to the DNA operator site [46, 47, 54, 55].

4 Sequestration of Toxin by Antitoxin

In the previous section, the complex AT was formed directly, while repressing its own production by binding to its own operator. In this section, AT is not created in a direct way. We rather introduce the antitoxin A and the toxin T that can bind to form the complex AT, which can still bind to the operator DNA (Fig. 1c). When it does so, it represses the transcription and translation of antitoxin and toxin, and therefore also indirectly its own production. We assume that the antitoxin alone is unable to bind to the operator. The relevant parameters for the antitoxin and toxin production and degradation are given in Table 1. These parameters are based on experimental measurements for the *phd/doc* TA system [14]. While antitoxin A is produced two to three times faster than the toxin protein T, it is also degraded four times faster. Faster creation and degradation rates for antitoxin than toxin are typical in toxin–antitoxin systems and have been shown to be essential for its operation [22]. The ODE and SDE description of this system is given as follows:

$$\frac{dA(t)}{dt} = \frac{\rho_A}{1 + \frac{AT(t)^n}{K^n}} - \alpha_C A(t)T(t) + \theta_C AT(t) - d_A A(t) + \eta(t), \tag{25}$$

$$\frac{dT(t)}{dt} = \frac{\rho_T}{1 + \frac{AT(t)^n}{K^n}} - \alpha_C A(t)T(t) + \theta_C AT(t) - d_T T(t) + \eta(t), \tag{26}$$

$$\frac{dAT(t)}{dt} = \alpha_C A(t)T(t) - \theta_C AT(t) - d_{AT} AT(t), \tag{27}$$

where $K = \theta_{AT}/\alpha_{AT}$ is defined the same as before, and $\alpha_C(\theta_C)$ describe the binding (unbinding) rates of A and T into the complex AT. Notice that AT is no longer produced in a direct way, but only indirectly through the binding of A and T. Furthermore, the Langevin noise term is only added to the evolution equations for A and T, as those are the only proteins that are actively created and are most likely to fluctuate in a stochastic manner. As before, Fig. 4a, b show the simulation results for the system simulated using the ODE and SDE.

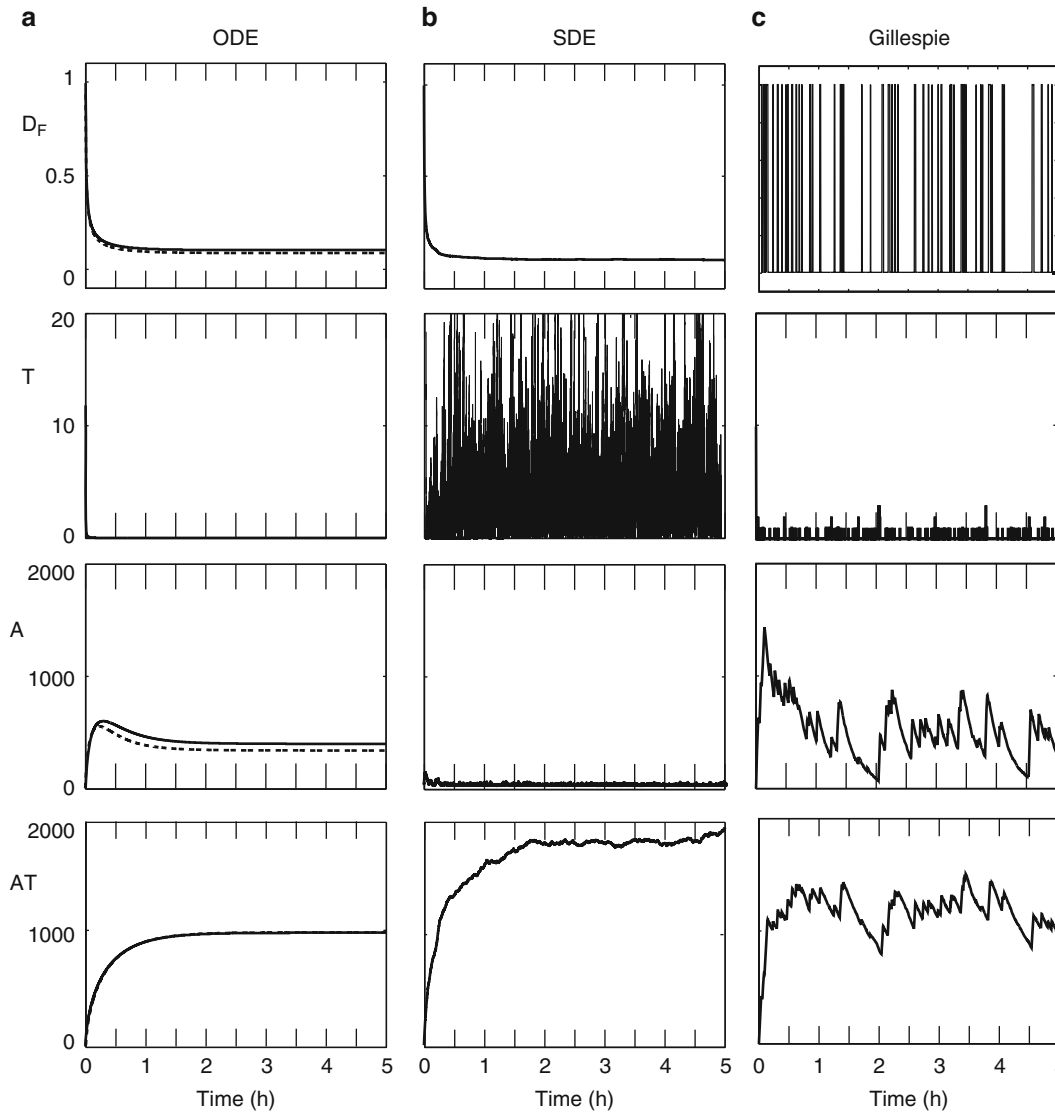

Fig. 4 Time evolution of the fraction D_F of time that the operator site on the DNA is unbound, the level of toxin T, antitoxin A and complex AT for a hypothetical toxin–antitoxin model describing production of antitoxin A and toxin T, complex formation and negative feedback through DNA binding (Fig. 1c). The systems given by Eqs. 25–27 were simulated for 5 h. The graphs show the results for a single cell. (**a**) Numerical simulation of the ODE using the Euler–Heun method, $D = 0$. The *dashed line* shows a simulation with the algebraic approximation (28). (**b**) Numerical simulation of the SDE using the Euler–Heun method, $D = 25$. (**c**) Numerical simulation of stochastic equations using the Gillespie algorithm. Parameters are given in Table 1

The results obtained by the ODE simulations are in fact very similar to the ones obtained through direct creation of AT (*see* Fig. 3a). This shows that the averaged response obtained by the deterministic ODE equations in (27) are adequately simplified by assuming that all toxins are quickly sequestered by the antitoxin A, forming the complex AT. This leads to a steady state value of

$T_{SS} \approx 0$ and the steady state value of AT is approximately the same as in Fig. 3a where we assumed it was created at the rate ρ_T.

A common approximation made in several papers is to assume that the process of complex formation through binding of A and T is constantly in equilibrium, *see*, e.g., [10, 12]. In this case, the amount of AT complexes in the system is determined by numerically simulating the evolution of antitoxin A and toxin T, after which one calculates AT at each discretization step using

$$AT(t) = \frac{A(t)T(t)}{K_C}, \tag{28}$$

which is thus solely determined by the ratio of the binding (unbinding) rates of A and T, $K_C = \theta_C/\alpha_C$. This approach is accurate for a wide range of binding (unbinding) rates of A and T. For instance, using the current parameter set (Table 1), the results of this approach are shown in Fig. 4a in the dashed lines. Apart from a minor difference in the level of antitoxin A, the approximation matches Eq. 27 which explicitly model the binding of A to T exceptionally well.

When introducing the Langevin noise terms, the results shown in Fig. 4b are quite different though. The toxin levels now fluctuate significantly and for these noise levels the antitoxin is no longer able to sequester all toxin, leading to larger toxin levels and low antitoxin levels. This shows that the effect of relatively small levels of noise can become significant due to the small numbers of toxin proteins. It also motivates the necessity to use the more correct discrete Gillespie algorithm to simulate stochastic effects.

The set of reactions used in the Gillespie algorithm are:

$$\begin{aligned}(1)\quad &\text{Free operator DNA}(D_F = 1) + AT \\ &\quad \rightarrow \text{Bound operator DNA}(D_F = 0),\end{aligned} \tag{29}$$

$$\begin{aligned}(2)\quad &\text{Bound operator DNA}(D_F = 0) \\ &\rightarrow \text{Free operator DNA}(D_F = 1) + AT,\end{aligned} \tag{30}$$

$$(3)\quad D_F \rightarrow A, \tag{31}$$

$$(4)\quad D_F \rightarrow T, \tag{32}$$

$$(5)\quad A \rightarrow \emptyset, \tag{33}$$

$$(6) \quad T \rightarrow \emptyset, \tag{34}$$

$$(7) \quad AT \rightarrow \emptyset, \tag{35}$$

$$(8) \quad A + T \rightarrow AT, \tag{36}$$

$$(9) \quad AT \rightarrow A + T, \tag{37}$$

with corresponding propensities:

$$(1) \quad p_1 = \alpha_{AT} D_F AT, \tag{38}$$

$$(2) \quad p_2 = \theta_{AT}(1 - D_F). \tag{39}$$

$$(3) \quad p_3 = \rho_A D_F, \tag{40}$$

$$(4) \quad p_4 = \rho_T D_F, \tag{41}$$

$$(5) \quad p_5 = d_A A, \tag{42}$$

$$(6) \quad p_6 = d_c T, \tag{43}$$

$$(7) \quad p_7 = d_c AT, \tag{44}$$

$$(8) \quad p_8 = \alpha_C A \cdot T, \tag{45}$$

$$(9) \quad p_9 = \theta_C AT, \tag{46}$$

The resulting time evolution obtained from the Gillespie method is shown in Fig. 4c. On average it gives a similar result as the results obtained from the ODE. However, the stochastic fluctuations in A and AT are considerably larger than with the SDE, while the toxin level remains at a stable low level. This shows that

the on/off toggling of the DNA transcription, only present in the Gillespie method, is essential to get the stochastic bursting behavior in the production of the various proteins.

Although we will further elaborate the model for the autoregulation of toxin–antitoxin modules in the next section, it should be noted that the model presented above can already be useful for toxin–antitoxin modules for which no indications of conditional cooperativity have been found, such as the *hipBA* operon [56]. Indeed, similar models have been published by Rotem et al., Koh and Dunlop and Feng et al. [9, 11, 15], often including the dimerization of the antitoxin HipB, repression by the antitoxin alone and growth rate modulation as explained below.

5 Conditional Cooperativity

Conditional cooperativity plays a pivotal role in the transcriptional regulation of many type II toxin–antitoxin modules [38, 40]. In this mechanism, the toxin acts as a corepressor for the DNA-binding antitoxin at low intracellular toxin:antitoxin ratios and as a derepressor at high toxin:antitoxin ratios. Here, we model conditional cooperativity solely via the binding of toxin T to the complex AT to form a secondary complex TAT which is unable to bind to the operator, as shown in Fig. 1d. In the presence of an excess of toxin T, this leads to a decrease in the level of AT, and consequently in a reduced repression through DNA binding. Direct derepression through toxins binding to AT complexes on the DNA itself is not included in this basic model.

A simplified view of conditional cooperativity can be obtained by only considering the binding (unbinding) events of A, T, AT, and TAT and neglecting active creation and degradation of these proteins and protein complexes. In Fig. 5a, we plot the amount of AT complexes in function of the total amount of toxin T_{tot} ($T_{\mathrm{tot}} = T + AT + 2 \cdot TAT$) present in the system, while keeping the total amount of antitoxin proteins A_{tot} ($A_{\mathrm{tot}} = A + AT + TAT$) fixed (e.g., $A_{\mathrm{tot}} = 1000$). We assume an immediate redistribution of toxin and antitoxin proteins among free antitoxin A, free toxin T, and complexes AT and TAT, based on the dissociation constants for the formation of complex AT from A and T and the formation of complex TAT from complex AT and T, which are chosen to be equal and are given by $K_C = \theta_C/\alpha_C$:

$$AT = \frac{A \cdot T}{K_C}, \tag{47}$$

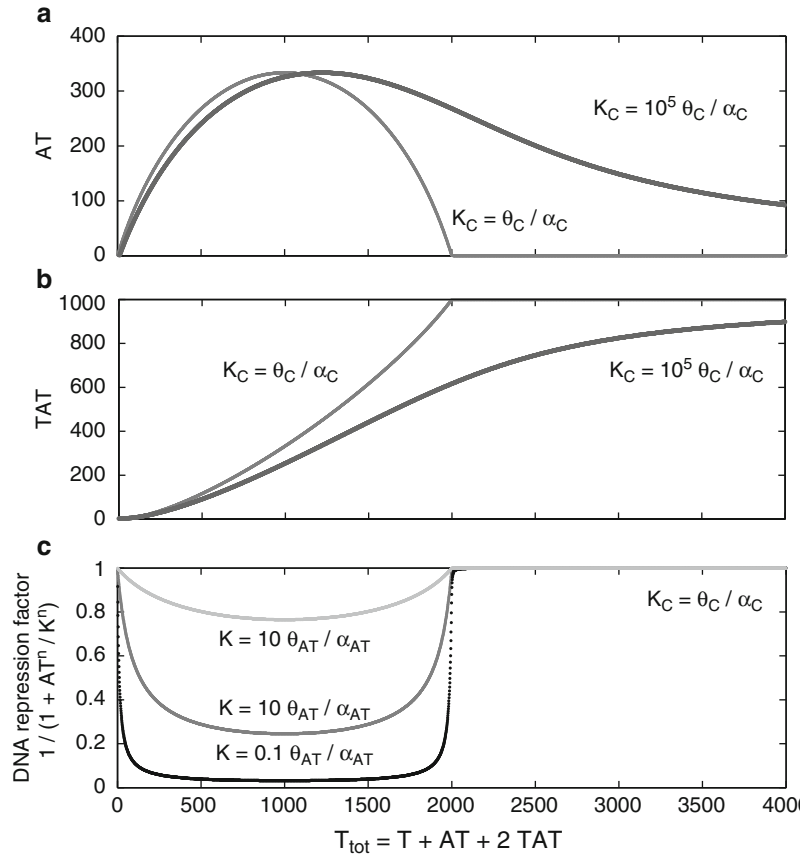

Fig. 5 Formation of complexes AT and TAT can lead to conditional cooperativity in a toxin–antitoxin system. Level of complex AT (**a**), complex TAT (**b**), and the DNA repression factor (**c**) as a function of the total toxin level T_{tot} for a fixed total antitoxin level $A_{tot} = 1000$, assuming immediate redistribution of free antitoxin, free toxin, complex AT, and complex TAT according to Eqs. 47 and 48. In panel (**a**) and (**b**), the *light gray line* represents physiological parameters, whereas the *dark gray line* represents a 10^5 times increased dissociation constant for the binding of toxin to antitoxin. In panel (**c**), the DNA repression factor is shown for three different values of the dissociation constant for the binding of complex AT to a binding site on the operator. Please note that the repression factor is defined so that it equals 1 when there is no repression and 0 when there is full repression. Parameters are given in Table 1

$$TAT = \frac{AT \cdot T}{K_C}, \tag{48}$$

Figure 5a, b shows that for values of $T_{tot} < 2A_{tot}$, the free toxins are efficiently sequestered in both complexes AT and TAT. As soon as the total amount of toxin T_{tot} exceeds twice the total amount of antitoxin A_{tot} present in the system (here $A_{tot} = 1000$),

AT complexes are removed from the system in favor of TAT complexes and free toxins. This transition is found to be a sharp one for physiological parameters, but can be smoothed out by greatly increasing the dissociation constant K_C, as shown in Fig. 5a, b. This transition has an immediate effect on the negative feedback through DNA repression, as the DNA repression factor was defined as $1/\left(1+\frac{AT(t)^n}{K^n}\right)$. As $AT \approx 0$ when $T_{\text{tot}} > 2A_{\text{tot}}$, no DNA repression takes place and all proteins can be transcribed and translated. However, when $T_{\text{tot}} < 2A_{\text{tot}}$ the AT complexes can bind to the DNA. The resulting repression is strong or weak, depending on the dissociation constant K for the binding of complex AT to a binding site on the operator. The stronger the DNA binding affinity, the stronger the resulting repression, as shown in Fig. 5c.

Although Fig. 5 provides insight into how conditional cooperativity can trigger a transition between two qualitatively different regions of operation (DNA repression vs. no DNA repression), the actual protein levels are dynamical variables that continuously influence each other in time. The ODE and SDE description of this system providing the time evolution of every protein level is given as follows:

$$\frac{dA(t)}{dt} = \frac{\rho_A}{1+\frac{AT(t)^n}{K^n}} - \alpha_C A(t)T(t) + \theta_C AT(t) - d_A A(t) + \eta(t), \tag{49}$$

$$\begin{aligned}\frac{dT(t)}{dt} = {} & \frac{\rho_T}{1+\frac{AT(t)^n}{K^n}} - \alpha_C A(t)T(t) + \theta_C AT(t) \\ & - \alpha_C AT(t)T(t) + \theta_C TAT(t) - d_T T(t) + \eta(t),\end{aligned} \tag{50}$$

$$\begin{aligned}\frac{dAT(t)}{dt} = {} & \alpha_C A(t)T(t) - \theta_C AT(t) - d_{AT} AT(t) \\ & + \theta_C TAT(t) - \alpha_C AT(t)T(t),\end{aligned} \tag{51}$$

$$\frac{dTAT(t)}{dt} = \alpha_C AT(t)T(t) - \theta_C TAT(t) - d_{TAT} TAT(t), \tag{52}$$

Notice that we again explicitly model all binding events and the approximation of Eqs. 47–48 is not used. Comparing Fig. 4a and Fig. 6a shows that the amount of free antitoxin A is approximately doubled and the amount of complex AT approximately halved

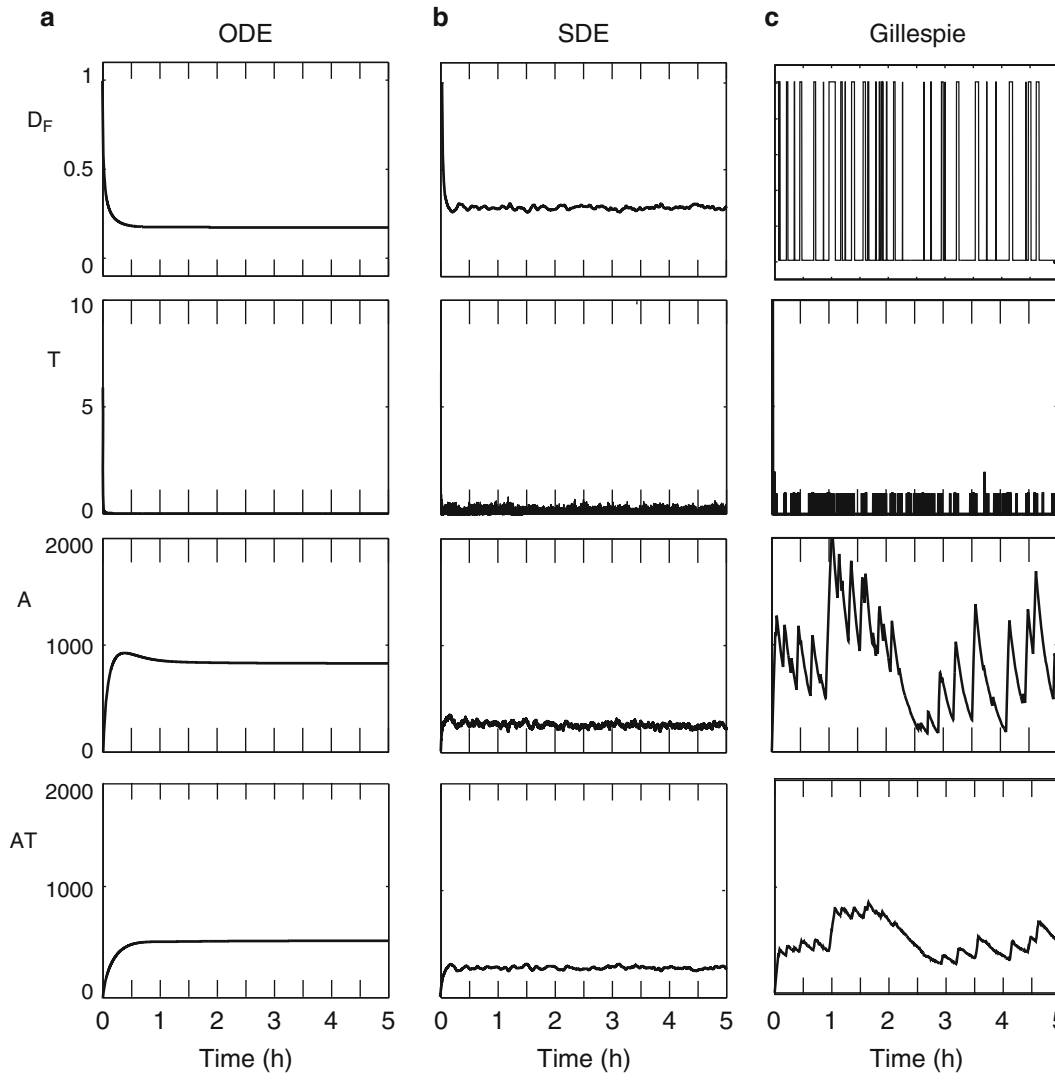

Fig. 6 Time evolution of the fraction D_F of time that the operator site on the DNA is unbound, the level of toxin T, antitoxin A and complex AT for a toxin–antitoxin model including conditional cooperativity (Fig. 1d). The systems given by Eqs. 49–52 were simulated for 5 h. The graphs show the results for a single cell. (**a**) Numerical simulation of the ODE using the Euler–Heun method, $D = 0$. (**b**) Numerical simulation of the SDE using the Euler–Heun method, $D = 25$. (**c**) Numerical simulation of stochastic equations using the Gillespie algorithm. Parameters are given in Table 1

when conditional cooperativity is included in the model. This is no surprise as AT complexes now help in sequestering toxins, and thus antitoxins have to do half the work in the presence of conditional cooperativity. Furthermore, including conditional cooperativity has a stabilizing effect in the presence of noise. Simulations using the

SDE become more robust, effectively decreasing the variation in the toxin level. Similarly as in Fig. 4b the average level of antitoxin A is decreased in the presence of moderate noise levels, due to the more frequent random production of toxins T that need to be sequestered into complexes.

The set of reactions used in the Gillespie algorithm are the same as before, with the addition of three extra reactions to account for the conditional cooperatively:

$$(10) \quad AT + T \rightarrow TAT, \tag{53}$$

$$(11) \quad TAT \rightarrow AT + T, \tag{54}$$

$$(12) \quad TAT \rightarrow \emptyset, \tag{55}$$

with corresponding propensities:

$$(10) \quad p_{10} = \alpha_C AT \cdot T, \tag{56}$$

$$(11) \quad p_{11} = \theta_C TAT, \tag{57}$$

$$(12) \quad p_{12} = d_c TAT, \tag{58}$$

Figure 6c shows similar results as before, but with larger variability of the free antitoxin level.

6 Growth Rate Modulation: Bistability and Metastability

Until now, we did not take into account that the toxin has an effect on the cellular metabolism. In reality, toxins can affect, e.g., translation, DNA replication, and the cell wall [57]. Therefore, the free toxin concentration will have an impact on the production rates of toxin and antitoxin, and more generally, by interfering with the global metabolism, on the cellular growth rate (Fig. 1e). We introduce a modulation factor,

$$\gamma_T = \frac{1}{1 + \frac{T(t)^{nn}}{KK^{nn}}},$$

with $KK = 1$ the threshold above which the toxin inhibits cell growth and $nn = 4$ the Hill factor. We assume that free toxin has

a symmetric effect on the production and degradation rates, reducing them with the same factor, while the degradation rate of the antitoxin remains the same.

The ODE and SDE description of this system is given as follows:

$$\frac{dA(t)}{dt} = \frac{\rho_A}{1 + \frac{AT(t)^n}{K^n}}\gamma_T - \alpha_C A(t)T(t) + \theta_C AT(t) - d_A A(t) + \eta(t), \tag{59}$$

$$\frac{dT(t)}{dt} = \frac{\rho_T}{1 + \frac{AT(t)^n}{K^n}}\gamma_T - \alpha_C A(t)T(t) + \theta_C AT(t) - \alpha_C AT(t)T(t) + \theta_C TAT(t) - d_T\gamma_T T(t) + \eta(t), \tag{60}$$

$$\frac{dAT(t)}{dt} = \alpha_C A(t)T(t) - \theta_C AT(t) - d_{AT}\gamma_T AT(t) + \theta_C TAT(t) - \alpha_C AT(t)T(t), \tag{61}$$

$$\frac{dTAT(t)}{dt} = \alpha_C AT(t)T(t) - \theta_C TAT(t) - d_{TAT}\gamma_T TAT(t), \tag{62}$$

The results are shown in Fig. 7a, b for an initial condition with a small excess of toxin: $A(t_0) = 1$ and $T(t_0) = 15$. In the deterministic case (ODE), the system finds itself initially in a situation where the toxin level is higher than the critical threshold KK above which cell growth is inhibited. The strength of this inhibition strongly depends on the Hill factor nn, which determines the sharpness of the transition from normal cell growth ($T < KK$) and reduced cell growth ($T > KK$). The increased level of initial toxin leads to a metastable state where one only slowly returns to low amounts of toxin and high amounts of antitoxin and complexes. In the end, the system is forced to return to the normal growth state due to the slow dilution of toxin in combination with the slow creation of antitoxin which sequesters the free toxins. A higher initial amount of toxin leads to an increasingly more slowly return to the controlled state. In the case for the SDE, the stochastic variations in protein level help to return more quickly to the controlled state.

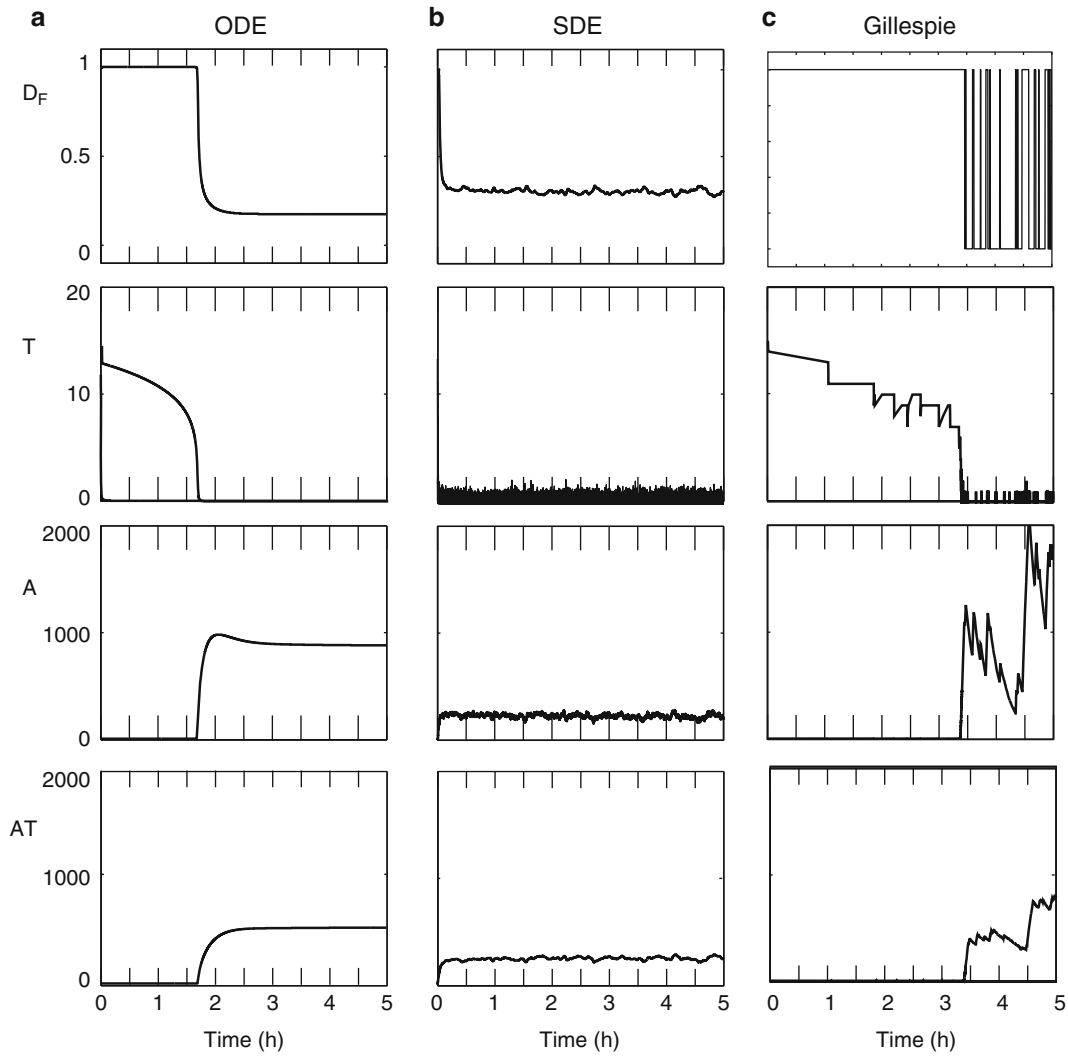

Fig. 7 Time evolution of the fraction D_F of time that the operator site on the DNA is unbound, the level of toxin T, antitoxin A and complex AT for a toxin–antitoxin model including conditional cooperativity and cell growth modulation (Fig. 1e). The systems given by Eqs. 59–62 were simulated for 5 h. The graphs show the results for a single cell. (**a**) Numerical simulation of the ODE using the Euler–Heun method, $D = 0$. (**b**) Numerical simulation of the SDE using the Euler–Heun method, $D = 25$. (**c**) Numerical simulation of stochastic equations using the Gillespie algorithm. Parameters are given in Table 1

The set of reactions used in the Gillespie algorithm remain the same, but the following propensities are adjusted to account for the growth rate modulation:

$$(3) \quad p_3 = \rho_A \gamma_T D_F, \tag{63}$$

$$(4) \quad p_4 = \rho_T \gamma_T D_F, \tag{64}$$

$$(6) \quad p_6 = d_c \gamma_T T, \tag{65}$$

$$(7) \quad p_7 = d_c \gamma_T AT, \tag{66}$$

$$(12) \quad p_{12} = d_c \gamma_T TAT, \tag{67}$$

Similar results were obtained for the Gillespie simulations (Fig. 7c). However, in the controlled state with low amounts of toxin, the stochastic variations of the antitoxin and complex levels are much larger than in the SDE approach.

These results indicate that noise can play an essential role in the formation of persister cells. Large stochastic excursions can drive the system above the threshold KK for growth inhibition. If these excursions are large enough, the growth rate can be reduced for potentially very long times. During this time the cell finds itself in a dormant state. In the end, however, the system relaxes back to the normally growing state, which is the only stable attractor. As shown in Fig. 7, noise can help to drive the system back to the stable growing state faster. Noise can thus work in both directions, stimulating the entry into a dormant state, and back to the normal state. Koh and Dunlop analyzed how various gene circuit architectures can give rise to more or less noise in the system and as such the frequency of persister cell creation might be influenced [11]. Furthermore, using the Gillespie algorithm, we showed that while the toxin level is generally controlled to be very low, noise could trigger very large pulses of free toxin [14]. This is illustrated in Fig. 8a, b. We analyzed the frequency of such rare extreme events in the absence of growth rate modulation, which is illustrated in Fig. 8a and shows that the probability of spikes in the toxin level becomes exponentially lower as their amplitude increases. When introducing growth rate modulation, these events were demonstrated to drive the cell into a metastable state of dormancy and returned back to normal growth conditions after a very long time (Fig. 8c).

For the parameter set used in this work, the system only admits one stable solution, the one in which cells grow normally and toxin levels are low and under control. Changing the initial conditions of the systems and/or adding noise can drive the system to a metastable state where it can reside for a potentially very long time, but eventually it returns to the normal growth conditions. No bistability has been found in the deterministic ODE system. Several authors have shown, however, that the presence of growth rate modulation can lead to bistability, where the normal state and the persister state are both stable [10, 12, 15]. In that case, noise can drive stochastic switches between both of these stable states.

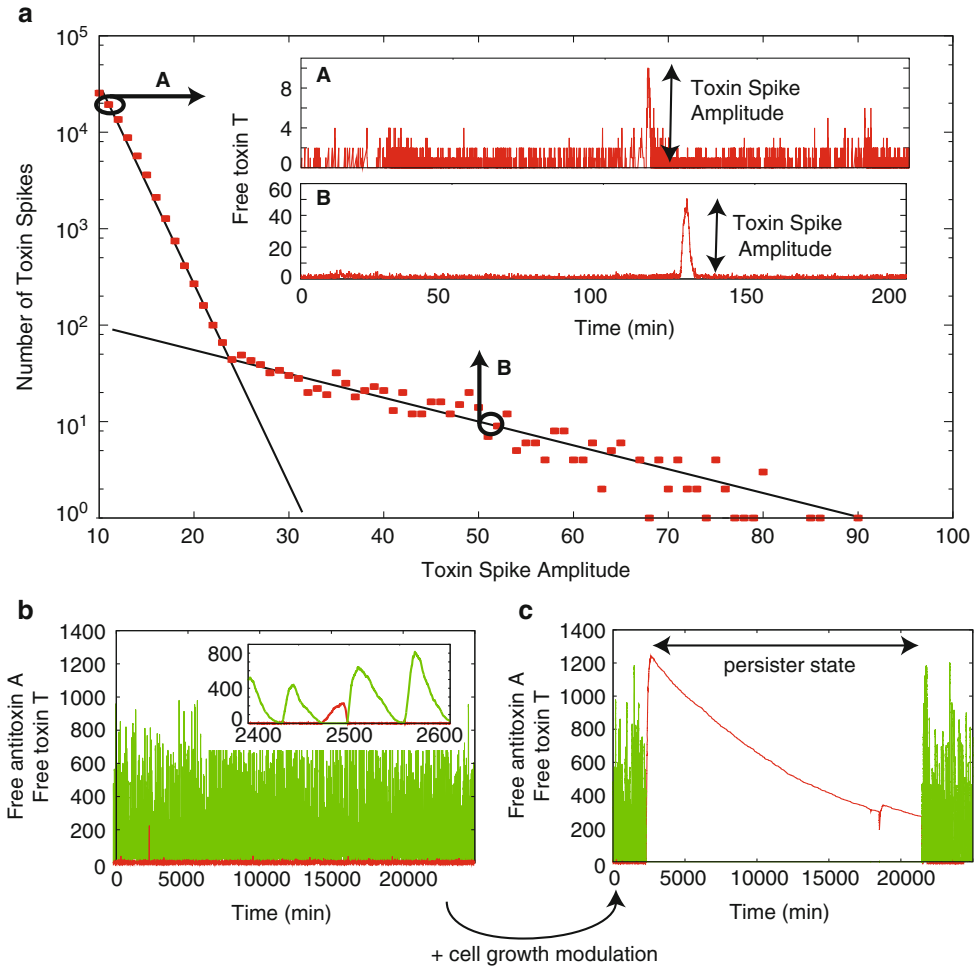

Fig. 8 Toxin–antitoxin module dynamics can cause large toxin spikes, providing a route to persister generation through growth rate suppression. (**a**) Number of toxin spikes in function of their amplitude. Two characteristic scaling laws (A) and (B) related to stochastic variation were found. (**b**) and (**c**) Level of free toxin (*red*) and antitoxin (*green*), respectively excluding (**b**) and including (**c**) toxic feedback effects. Figure adapted from [14]. For more details on model, parameter set and the presence of the two characteristic scaling laws in (**a**), *see* [14]

Cataudella et al. have shown how conditional cooperativity can help to mediate bistability between the normal state and the persister state [12]. In other works, such as [10, 15], bistability in the absence of conditional cooperativity was analyzed. However, no bistability has been observed in the absence of growth rate modulation, showing that this is an essential ingredient to achieve bistability.

7 Modeling Populations of Persister Cells

Previous sections focussed on modeling the dynamics of TA systems within a single cell. Such studies reveal potential mechanisms that can lead to elevated toxin levels driving a single cell into a dormant state, characterized by a much slower growth rate than normal cells. In reality, cells exist and grow within a larger population of cells. In order to understand how the single cell dynamics translates to the dynamics of whole cell populations, a simple two-state model can be used. One state is the normal (N) cell, while the other state is the persister (P) cell. Cells can actively switch from N to P and P to N (the switching rates are defined as a and b, respectively), while the growth rate of both states is given by μ_N and μ_P:

$$\frac{dN}{dt} = -aN + bP + \mu_N N \tag{68}$$

$$\frac{dP}{dt} = aN - bP + \mu_P P \tag{69}$$

This model was first introduced by Balaban et al. as a model for persisters created through normal growth (type II) [7]. The phenotypic switching rates a and b can be estimated from the underlying single cell dynamics and will depend on the system parameters, noise strength, and type of dynamics (e.g., bistability [58] vs. metastability [14]). In the presence of nutritional stress, cells tend to switch predominantly to the high toxin state and switch back much more rarely [14, 58]. This translates to $a > b$. While the cell spends most of its time in the persister state, the persister fraction of the overall cell population is only a minority. This is due to the fact that the normal cell population has a much larger individual cell growth rate with respect to the persisters. The persister fraction in the whole cell population is thus greatly determined by both phenotypic switching rates and the growth rates. Figure 9 shows an analysis of the dependence of the persister fraction on both switching rates a and b. One can clearly see that the switching rate to get into persistence strongly controls the persister fractions. In nutritional stress conditions, an increase in the switching rate a (N to P) thus immediately leads to an increased persister fraction. The return rate to normal cell growth (b) has little influence on the persister fraction, provided that it is slower than the growth rate μ_N.

The model for persisters generated through normal growth by Balaban et al. was further investigated by Kussell et al. Next to the deterministic approach, suitable for large bacterial populations, they also include stochastic simulations using the Gillespie algorithm,

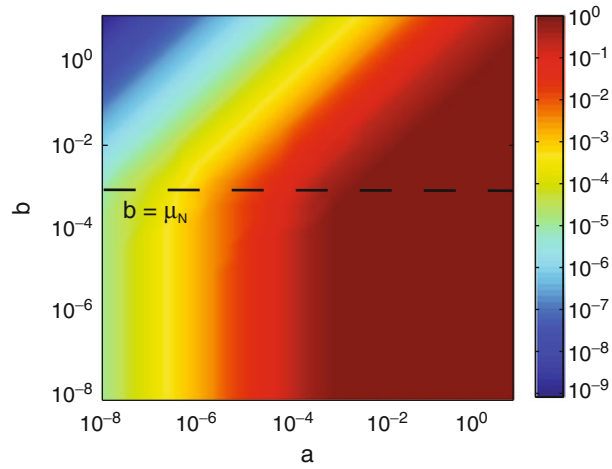

Fig. 9 Parameter scan showing the persister fraction in function of the switching rates *a* (from normal to persister cell) and *b* (from persister cell to normal cell). Parameters: $\mu_N = d_c$, $\mu_P = 0$

which better represents the behavior for small populations. This work shows that the optimal switching rate between normal cells and persisters depends more on the frequency of environmental changes than on the specific characteristics of each environment [6]. In 2007, Cogan published another model focusing on the dynamics of cell populations, yet incorporating toxin–antitoxin modules [8].

8 Discussion

When comparing different mathematical modeling papers on toxin–antitoxin modules and persisters, it becomes clear that different models can give different results, both depending on the system one is trying to simulate and on the assumptions that were made for a particular system. For example, for the *hipBA* operon, Koh and Dunlop concluded that the emergence of persistence is not caused by bistability [11], whereas Feng et al.—after inclusion of growth rate modulation—motivate that it is [15].

This indicates that model building is a crucial step when examining toxin–antitoxin modules and persister cells using numerical methods. Depending on the particular system one is modeling, the parameters and equations will differ from the ones presented here. Particular things that should be taken into account are the monomeric or multimeric state of the toxin and antitoxin, the toxin–antitoxin complexes that can be formed and the details of the transcriptional regulation. For the latter, one could consider, e.g., the amount of binding sites for the antitoxin on the DNA operator, the fact if the antitoxin alone can cause repression, and if conditional cooperativity has been found in the system.

As more and more details about the mechanisms regulating toxin–antitoxin modules are being elucidated experimentally, new interactions can of course be added to the models. For example, the explicit mode of action is known for several toxins, such as inhibition of translation by phosphorylation of elongation factor Tu (EF-Tu) for Doc [59] or ribosome-dependent degradation of mRNA for RelE [60]. The effect of the mode of action of the toxin on the toxin–antitoxin dynamics could be investigated by integrating the different toxic activities into a general model. Furthermore, for the *hipBA* operon, experimental results indicate that the autophosphorylation of toxin HipA plays a role in reverting from a persister to a growth phenotype. This behavior has not been included in *hipBA* models up to now.

Various models have been constructed to describe the single cell dynamics of type II TA systems [5, 9–15]. Although these models have often used different equations and different numerical methods (deterministic vs. stochastic), one general conclusion that can be drawn from these modeling efforts is that growth rate inhibition at higher toxin levels is an essential property to allow for persister cell creation. Without such growth rate dependence on the toxin level the system is found to be monostable with only minor short stochastic excursions to states with higher toxin levels. When including growth rate inhibition, two possible outcomes have been found that can lead to persister formation. Either the system is found to be metastable, where stochastic excursions can drive the system into a dormant state [14]. The system can reside in this state for a long transient time, but eventually returns to the stable solution of the system, corresponding to a low toxin level. The system has also been shown to allow for bistability [10, 12, 15]. This situation is fundamentally different from metastability as in the absence of noise there are two stable states (low and high toxin level). In the presence of noise, stochastic mode hopping between both these states can be found.

In this work, we did not consider the fact that one bacterium mostly contains many toxin–antitoxin modules. Fasani et al. did include this in their model and conclude that multiple toxin–antitoxin modules can be coupled to provide a strong hysteretic switch between the normal growing and the persister phenotype [13].

Apart from the actual model and its parameters, the outcome of simulations for toxin–antitoxin module dynamics depends heavily on whether randomness and noise are included. This was clearly illustrated in Fig. 4, where the level of the free toxin was close to zero when the ODE approach was used, but fluctuated heavily when relatively low levels of noise were added in the SDE approach. These fluctuations are greatly reduced when the Gillespie algorithm is used. This approach is more realistic because every reaction is simulated explicitly. As the gene regulation in toxin–antitoxin modules is a discrete process, where transcription and translation can

only take place when the operator is unbound, and as there are usually very few free toxin molecules in the cell, the Gillespie algorithm is the most appropriate method to simulate stochastic effects in this case. Although the ODE and SDE approach are less realistic than the Gillespie algorithm, they can be very useful because of their simplicity and the possibility to derive analytical solutions. Steady state solutions can be calculated in ODE systems and their stability can readily be analyzed using dynamical systems theory. In the case of the SDE, stationary probability distributions can be derived by solving the stationary Fokker–Planck equation corresponding to the SDE [51].

Acknowledgements

This research was supported by the Vlaams Interuniversitair Instituut voor Biotechnologie (VIB), by the Research Foundation - Flanders (FWO-Vlaanderen) for project support and individual support (A.V. and L.G.), by the Belgian American Educational Foundation (L.G.), and by the Onderzoeksraad of the Vrije Universiteit Brussel. The authors thank Lydia Hill, Abel Garcia-Pino, and Egon Geerardyn for fruitful discussions.

Appendix 1: Numerical Code to Solve an ODE/SDE

A simple matlab code to solve Eq. 10, with ($D \neq 0$) or without noise ($D = 0$), can be found here below:

```
function ODE_SDE

%% parameters
prodAT  = 0.0530 * 0.116086/0.00203;
degrAT  = 2.8881e-4;
D       = 25;
dt      = 0.01;     % [s] simulation time step
dt_save = 10;       % [s] plotting time step
t_end   = 5*60*60; % [s] final time

%% initialize system
AT       = 0;
t_saved  = [];
AT_saved = [];
count    = 0;

%% simulate the stochastic differential equation
for n = 0:((t_end)/dt)
    t = n * dt;

```

```
    %% Euler-Heun
    noise = sqrt(D) * sqrt(dt) * randn(); % sample from the
    noise

    AT_star = AT + dt * F(AT) + noise;
    AT      = AT + (dt/2)*(F(AT) + F(AT_star)) + noise/2;
    AT      = max(AT, 0); % force protein concentration to
              be positive

    %% Save data
       if (count == dt_save/dt)
       t_saved(end+1) = t;
       AT_saved(end+1) = AT;
       count = 0;
    end
    count = count + 1;

end

%% plot the results
figure;
plot(t_saved./3600, AT_saved, 'k');
xlabel('Time (h)')
ylabel('AT')

%% definition of the differential equation
    function dATdt = F(TA)
    dATdt = prodAT - degrAT*TA;
  end
end
```

Appendix 2: Numerical Code Using the Gillespie Algorithm

A simple matlab code to solve Eq. 1 using the stochastic Gillespie Algorithm can be found here below:

```
% Gillespie code
% There are 2 reactions and there is one species AT

%% Parameters
prodAT = 0.0530*0.116086/0.00203; % reaction 0 -> AT
degrAT = 2.8881e-4;               % reaction AT -> 0

%% Initialization
AT         = 0;       % [AT] initial concentration AT
t          = 0;       % [s] starting time
t_end      = 5*60*60; % [s] final time
t_saved    = [];      % [s] stored times
AT_saved   = [];

```

```
%% Simulation
while t <= t_end
  %% Update propensities
  p1 = degrAT * AT;
  p2 = prodAT;

  %% Computation of the random time step
  p0 = p1 + p2;
  r1 = rand();
  r2 = rand();
  dt = 1/p0 * log(1/r1); % [s] next time step

  %% Selection of random reaction
  %% Update the population based on selected reaction
  yr2 = r2 * p0;
  if yr2 <= p1
    % reaction 1
    AT = AT - 1;
  else
    % reaction 2
    AT = AT + 1;
  end

  %% Update the current time
  t = t + dt;

  %% Save population information
  t_saved(end+1) = t;
  AT_saved(end+1) = AT;

end

%% plot the results
figure;
plot(t_saved./3600, AT_saved, 'k');
xlabel('Time (h)')
ylabel('AT')
```

References

1. Pomerening JR, Sontag ED, Ferrell JE Jr (2003) Building a cell cycle oscillator: hysteresis and bistability in the activation of Cdc2. Nat Cell Biol 5(4):346–351
2. Novak B, Tyson JJ (1993) Numerical analysis of a comprehensive model of M-phase control in Xenopus oocyte extracts and intact embryos. J Cell Sci 106(Pt 4):1153–1168
3. Noble D (2004) Modeling the heart. Physiology (Bethesda) 19:191–197
4. Grassly NC, Fraser C (2008) Mathematical models of infectious disease transmission. Nat Rev Microbiol 6(6):477–487
5. Cataudella I, Trusina A, Sneppen K, Gerdes K, Mitarai N (2012) Conditional cooperativity in toxin-antitoxin regulation prevents random

toxin activation and promotes fast translational recovery. Nucleic Acids Res 40(14):6424–6434

6. Kussell E, Kishony R, Balaban NQ, Leibler S (2005) Bacterial persistence: a model of survival in changing environments. Genetics 169 (4):1807–1814
7. Balaban NQ, Merrin J, Chait R, Kowalik L, Leibler S (2004) Bacterial persistence as a phenotypic switch. Science 305(5690):1622–1625
8. Cogan NG (2007) Incorporating toxin hypothesis into a mathematical model of persister formation and dynamics. J Theor Biol 248(2):340–349
9. Rotem E, Loinger A, Ronin I, Levin-Reisman I, Gabay C, Shoresh N, Biham O, Balaban NQ (2010) Regulation of phenotypic variability by a threshold-based mechanism underlies bacterial persistence. Proc Natl Acad Sci USA 107 (28):12541–12546
10. Lou C, Li Z, Ouyang Q (2008) A molecular model for persister in E. coli. J Theor Biol 255 (2):205–209
11. Koh RS, Dunlop MJ (2012) Modeling suggests that gene circuit architecture controls phenotypic variability in a bacterial persistence network. BMC Syst Biol 6:47
12. Cataudella I, Sneppen K, Gerdes K, Mitarai N (2013) Conditional cooperativity of toxin - antitoxin regulation can mediate bistability between growth and dormancy. PLoS Comput Biol 9(8):e1003174
13. Fasani RA, Savageau MA (2013) Molecular mechanisms of multiple toxin-antitoxin systems are coordinated to govern the persister phenotype. Proc Natl Acad Sci USA 110(27): E2528–E2537
14. Gelens L, Hill L, Vandervelde A, Danckaert J, Loris R (2013) A general model for toxin-antitoxin module dynamics can explain persister cell formation in E. coli. PLoS Comput Biol 9(8):e1003190
15. Feng J, Kessler DA, Ben-Jacob E, Levine H (2014) Growth feedback as a basis for persister bistability. Proc Natl Acad Sci USA 111 (1):544–549
16. Lewis K (2010) Persister cells. Annu Rev Microbiol 64:357–372
17. Fauvart M, De Groote VN, Michiels J (2011) Role of persister cells in chronic infections: clinical relevance and perspectives on anti-persister therapies. J Med Microbiol 60(Pt 6):699–709
18. Maisonneuve E, Gerdes K (2014) Molecular mechanisms underlying bacterial persisters. Cell 157(3):539–548
19. Maisonneuve E, Castro-Camargo M, Gerdes K (2013) (p)ppGpp controls bacterial persistence by stochastic induction of toxin-antitoxin activity. Cell 154(5):1140–1150
20. Pandey DP, Gerdes K (2005) Toxin-antitoxin loci are highly abundant in free-living but lost from host-associated prokaryotes. Nucleic Acids Res 33(3):966–976
21. Fozo EM, Hemm MR, Storz G (2008) Small toxic proteins and the antisense RNAs that repress them. Microbiol Mol Biol Rev 72 (4):579–589
22. Gerdes K, Maisonneuve E (2012) Bacterial persistence and toxin-antitoxin loci. Annu Rev Microbiol 66:103–123
23. Buts L, Lah J, Dao-Thi MH, Wyns L, Loris R (2005) Toxin-antitoxin modules as bacterial metabolic stress managers. Trends Biochem Sci 30(12):672–679
24. Yamaguchi Y, Park JH, Inouye M (2011) Toxin-antitoxin systems in bacteria and archaea. Annu Rev Genet 45:61–79
25. Blower TR, Salmond GP, Luisi BF (2011) Balancing at survival's edge: the structure and adaptive benefits of prokaryotic toxin-antitoxin partners. Curr Opin Struct Biol 21 (1):109–118
26. Blower TR, Short FL, Rao F, Mizuguchi K, Pei XY, Fineran PC, Luisi BF, Salmond GP (2012) Identification and classification of bacterial Type III toxin-antitoxin systems encoded in chromosomal and plasmid genomes. Nucleic Acids Res 40(13):6158–6173
27. Wang X, Lord DM, Cheng HY, Osbourne DO, Hong SH, Sanchez-Torres V, Quiroga C, Zheng K, Herrmann T, Peti W, Benedik MJ, Page R, Wood TK (2012) A new type V toxin-antitoxin system where mRNA for toxin GhoT is cleaved by antitoxin GhoS. Nat Chem Biol 8 (10):855–861
28. Maisonneuve E, Shakespeare LJ, Jorgensen MG, Gerdes K (2011) Bacterial persistence by RNA endonucleases. Proc Natl Acad Sci USA 108(32):13206–13211
29. Helaine S, Cheverton AM, Watson KG, Faure LM, Matthews SA, Holden DW (2014) Internalization of Salmonella by macrophages induces formation of nonreplicating persisters. Science 343(6167):204–208
30. Tripathi A, Dewan PC, Barua B, Varadarajan R (2012) Additional role for the ccd operon of F-plasmid as a transmissible persistence factor. Proc Natl Acad Sci USA 109(31):12497–12502
31. Tian QB, Ohnishi M, Tabuchi A, Terawaki Y (1996) A new plasmid-encoded proteic killer

gene system: cloning, sequencing, and analyzing hig locus of plasmid Rts1. Biochem Biophys Res Commun 220(2):280–284

32. Yamaguchi Y, Park JH, Inouye M (2009) MqsR, a crucial regulator for quorum sensing and biofilm formation, is a GCU-specific mRNA interferase in Escherichia coli. J Biol Chem 284(42):28746–28753
33. Hallez R, Geeraerts D, Sterckx Y, Mine N, Loris R, Van Melderen L (2010) New toxins homologous to ParE belonging to three-component toxin-antitoxin systems in Escherichia coli O157:H7. Mol Microbiol 76 (3):719–732
34. Overgaard M, Borch J, Gerdes K (2009) RelB and RelE of Escherichia coli form a tight complex that represses transcription via the ribbon-helix-helix motif in RelB. J Mol Biol 394 (2):183–196
35. Schumacher MA, Piro KM, Xu W, Hansen S, Lewis K, Brennan RG (2009) Molecular mechanisms of HipA-mediated multidrug tolerance and its neutralization by HipB. Science 323(5912):396–401
36. Loris R, Dao-Thi MH, Bahassi EM, Van Melderen L, Poortmans F, Liddington R, Couturier M, Wyns L (1999) Crystal structure of CcdB, a topoisomerase poison from E. coli. J Mol Biol 285(4):1667–1677
37. Li GY, Zhang Y, Chan MC, Mal TK, Hoeflich KP, Inouye M, Ikura M (2006) Characterization of dual substrate binding sites in the homodimeric structure of Escherichia coli mRNA interferase MazF. J Mol Biol 357(1):139–150
38. Garcia-Pino A, Balasubramanian S, Wyns L, Gazit E, De Greve H, Magnuson RD, Charlier D, van Nuland NA, Loris R (2010) Allostery and intrinsic disorder mediate transcription regulation by conditional cooperativity. Cell 142 (1):101–111
39. Afif H, Allali N, Couturier M, Van Melderen L (2001) The ratio between CcdA and CcdB modulates the transcriptional repression of the ccd poison-antidote system. Mol Microbiol 41 (1):73–82
40. Overgaard M, Borch J, Jorgensen MG, Gerdes K (2008) Messenger RNA interferase RelE controls relBE transcription by conditional cooperativity. Mol Microbiol 69(4):841–857
41. De Jonge N, Garcia-Pino A, Buts L, Haesaerts S, Charlier D, Zangger K, Wyns L, De Greve H, Loris R (2009) Rejuvenation of CcdB-poisoned gyrase by an intrinsically disordered protein domain. Mol Cell 35(2):154–163
42. Brown BL, Lord DM, Grigoriu S, Peti W, Page R (2013) The Escherichia coli toxin MqsR destabilizes the transcriptional repression complex formed between the antitoxin MqsA and the mqsRA operon promoter. J Biol Chem 288(2):1286–1294
43. Magnuson R, Lehnherr H, Mukhopadhyay G, Yarmolinsky MB (1996) Autoregulation of the plasmid addiction operon of bacteriophage P1. J Biol Chem 271(31):18705–18710
44. Dao-Thi MH, Charlier D, Loris R, Maes D, Messens J, Wyns L, Backmann J (2002) Intricate interactions within the ccd plasmid addiction system. J Biol Chem 277 (5):3733–3742
45. McAdams HH, Arkin A (1999) It's a noisy business! Genetic regulation at the nanomolar scale. Trends Genet 15(2):65–69
46. Elowitz MB, Levine AJ, Siggia ED, Swain PS (2002) Stochastic gene expression in a single cell. Science 297(5584):1183–1186
47. Ozbudak EM, Thattai M, Kurtser I, Grossman AD, van Oudenaarden A (2002) Regulation of noise in the expression of a single gene. Nat Genet 31(1):69–73
48. Carrier GF (1968) Ordinary differential equations. A Blaisdell book in pure and applied mathematics. Blaisdell Pub. Co, Waltham, MA
49. Atkinson K, Han W, Stewart DE (2009) Numerical solution of ordinary differential equations. Pure and applied mathematics. Wiley, Hoboken, NJ
50. Coffey WT, Kalmykov YP, Waldron JT (2004) The Langevin equation. With applications to stochastic problems in physics, chemistry and electrical engineering. World Scientific series in contemporary chemical physics. World Scientific Publishing, Singapore
51. Gardiner CW (2004) Handbook of stochastic methods for physics, chemistry, and the natural sciences. Springer, Berlin
52. San Miguel M, Toral R (2000) Stochastic effects in physical systems. In: Instabilities and nonequilibrium structures VI. Springer, Netherlands, pp 35–127
53. Gillespie DT (1977) Exact stochastic simulation of coupled chemical reactions. J Phys Chem 81(25):2340–2361
54. Cai L, Friedman N, Xie XS (2006) Stochastic protein expression in individual cells at the single molecule level. Nature 440(7082): 358–362
55. McAdams HH, Arkin A (1997) Stochastic mechanisms in gene expression. Proc Natl Acad Sci USA 94(3):814–819
56. Loris R, Garcia-Pino A (2014) Disorder- and dynamics-based regulatory mechanisms in

toxin-antitoxin modules. Chem Rev 114 (13):6933–6947

57. Hayes F, Van Melderen L (2011) Toxins-antitoxins: diversity, evolution and function. Crit Rev Biochem Mol Biol 46(5):386–408
58. Nevozhay D, Adams RM, Van Itallie E, Bennett MR, Balazsi G (2012) Mapping the environmental fitness landscape of a synthetic gene circuit. PLoS Comput Biol 8(4):e1002480
59. Castro-Roa D, Garcia-Pino A, De Gieter S, van Nuland NA, Loris R, Zenkin N (2013) The Fic protein Doc uses an inverted substrate to phosphorylate and inactivate EF-Tu. Nat Chem Biol 9(12):811–817
60. Christensen SK, Gerdes K (2003) RelE toxins from bacteria and Archaea cleave mRNAs on translating ribosomes, which are rescued by tmRNA. Mol Microbiol 48(5):1389–1400