Gene Regulation in the Pi Calculus: Simulating Cooperativity at the Lambda Switch

Céline Kuttler, Joachim Niehren

Gene Regulation in the Pi Calculus: Simulating Cooperativity at the Lambda Switch

Céline Kuttler[1] and Joachim Niehren[2]

[1] Interdisciplinary Research Institute, Lille, France*
[2] INRIA Futurs, Lille, France**

Abstract. We propose to model the dynamics of gene regulatory networks as concurrent processes in the stochastic pi calculus. As a first case study, we show how to express the control of transcription initiation at the lambda switch, a prototypical example where cooperative enhancement is crucial. This requires concurrent programming techniques that are new to systems biology, and necessitates stochastic parameters that we derive from the literature. We test all components of our model by exhaustive stochastic simulations. A comparison with previous results reported in the literature, experimental and simulation based, confirms the appropriateness of our modeling approach.

1 Introduction

In living cells, genes and proteins interact in networks of gene regulation. All cells of a multicellular organism contain the same genetic material. Nevertheless, the use made of it varies widely between different tissues. The current state of a cell is determined by the proteins it contains; it changes when new proteins are produced by decoding genetic information.

Understanding the dynamic behavior of gene regulatory systems is a challenge to computational systems biology. The molecular actors within these networks interact *nondeterministically*. Given a particular condition, one can never tell with certainty which among several thinkable reactions will follow next. What occurs strongly depends on the identities of various proteins in the cell, their interaction capabilities, quantities – and random encounters. Such effects accumulate, making it difficult to predict the behavior of a system as a whole, even if its components are well characterized.

Informal descriptions of prototypical gene regulatory networks can be found in biological textbooks [13, 35]. These deal with qualitative aspects such as the possible reactions between molecular actors. They also address quantitative aspects as frequencies of such reactions, but usually remain rather vague on these. Precise quantitative parameters are more difficult to access. For well studied

* Interdisciplinary Research Institute, FRE 2963 of CNRS, in cooperation with the University of Lille 1 and supported by the Conseil Régional Nord-Pas de Calais.
** Mostrare Project of INRIA Futurs at the LIFL, in cooperation with the Universities of Lille 1 and 3.

systems they have been determined in series of experiments, and reported in the research literature.

Simulations can help understanding the dynamics of gene regulatory networks [7, 19]. This particularly holds for cases in which informal qualitative descriptions exist as well as quantitative characterizations. The question that remains is whether the available knowledge suffices to correctly predict the system's behavior. This can be shown by building a mathematical model, executing it, and comparing simulation and experimental results.

In this article, we propose to formally model gene regulatory networks as *communicating stochastic processes*, to our knowledge, for the first time. We draw inspiration from previous stochastic models of gene expression [2, 12, 21]. We follow Regev and Shapiro [38] in applying the *stochastic π-calculus* [33] as modeling language for systems biology. This is Milner's et. al. π-calculus [29, 28] – a fundamental model of concurrency – extended by stochastic control (see also [18]). Nondeterminism is inherent to concurrent computation, of which the π-calculus abstracts the essential features. Stochastic parameters control communication or *interaction* frequencies, and thus the evolution of the numbers of actors over time. Execution of stochastic π-calculus models yields *stochastic simulation* based on Gillespie's algorithm [14], using the BioSpi engine [34] or similar systems [32].

We investigate a prototypical instance of gene regulation in a bacterium, for which both qualitative and quantitative knowledge are available. As a first case study, we show how to model and simulate the control of *transcription initiation at the lambda switch* [35], a prototypical example where *cooperativity* is crucial. To be concrete, we model the molecular interactions at bacteriophage λ's right operator region, including positive control of transcription initiation and cooperativity in protein binding. This requires concurrent programming techniques that are new to systems biology:

1. we use *handshake protocols* in order to express *many-to-many communication* on same channel;
2. we use *alternative timer agents* in order to alternate stochastic rates associated to channels, this allows to express *cooperative enhancement* of the channel's activity.

We show how to compute the stochastic parameters from the literature, and integrate these parameters into our formal π-calculus models. We validate our models and parameters by running exhaustive simulation tests. One of the strengths of our approach is that we can easily simulate idealized subsystems, in order to observe distinguished phenomena independently from the system as a whole. We design a sequence of sub-models of different degrees of complexity, in order to simulate the many factors influencing transcription initiation at the λ switch. The simulation results we obtain convincingly confirm the appropriateness of our model[3].

[3] These simulation results are new compared to the presentation at the second international workshop on concurrent models in molecular biology (BioConcur 2004). They have permitted us to spot some flaws in the previous parameter sets.

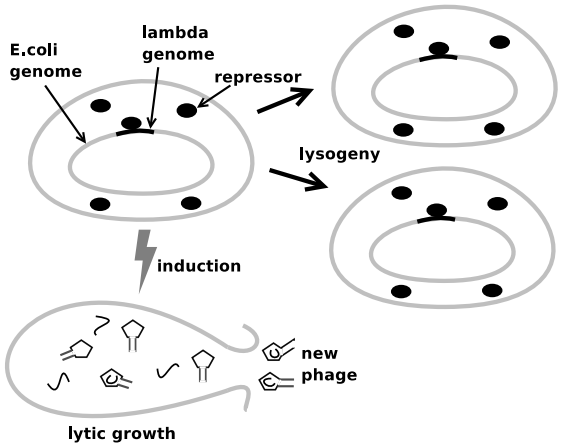


Fig. 1. Two pathways of λ infected E.coli bacterium: lysogeny and lytic growth.

The general mechanisms of cooperativity we model can be observed in many other places, ranging from the assembly of protein complexes, DNA looping to regulatory mechanisms in eukaryotes. For a discussion see [36]. Note that more aspects remain to be included in our stochastic π-calculus model of the λ switch in other to reflect recent observations that lead to the revision of long established assumptions [11, 40]. The modeling techniques proposed in this paper, however, should be sufficient there too.

Plan. We first describe the regulatory network at the λ switch informally (Sec. 2) and then distill the stochastic parameters from the literature (Sec. 3). We recall the stochastic π-calculus (Sec. 4) and apply it for modeling the network of the λ switch (Sec. 5). Finally, we present simulations for different scenarios obtained by implementation of our model in the BioSpi system (Sec. 6). We motivate the different set-ups with experiments or other simulations, and try to relate both to each other.

2 The λ Switch

Gene regulation at the lambda switch has remained a fruitful research area for decades [10, 31, 35]. It has served as a benchmark for testing simulation methods [15], and to reproduce or elucidate experimental knowledge [2, 3].

2.1 Pathways

Bacteriophage λ is a virus which infects the bacterium *Escherichia coli*. Injecting its genome into the bacterial cell, two developmental pathways as illustrated in Fig. 1 are possible. Either, in *lytic growth* the viral genome uses the molecular

machinery of the bacterial cell to produce new viruses and eventually burst the host. Alternatively, the viral genome gets integrated into the bacterial genome. Note the highlighted segment within the bacterial genome in Fig. 1. The only viral protein expressed is then the λ repressor, which disables the expression of all others through binding to dedicated segments of the viral genome. The host cell is now immune against further infections. The viral genome is subsequently transmitted to further bacterial generations in a passive way. This state called *lysogeny* is extremely stable, and usually maintained for generations. Spontaneous transitions from lysogeny to the state of lytic growth would occur about once every 5000 years for a single bacterial cell [10]. Considering that it takes the bacterium no longer than hour to divide into two daughter cells, the lysogenic state is extremely stable.

But surprisingly, upon an environmental signal the phage genome can efficiently become re-activated – this is called *induction*. Now, the bacterium switches from lysogeny into the phase of lytic growth. The viral genome is extracted from the host's, and uses the cell machinery to produce a fresh crop of viruses. This unavoidably leads to the *lysis*, or destruction of the host cell. What happens during induction, as well as the maintenance of lysogeny, crucially depends on the control of transcription initiation within O_R, the right operator region of phage λ's genome. O_R is commonly referred to as *the* λ switch.

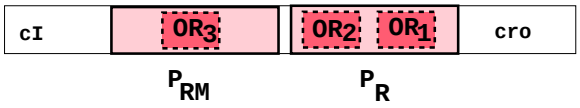
2.2 Network Controlling Transcription Initiation

The control of transcription initiation at the λ switch illustrates phenomena of *cooperativity*, which are even more important for gene regulation in higher forms of life [36]. Cooperative enhancement of a reaction between two molecular actors means that its strength is enhanced by a third, otherwise independent actor. We will see two instances of cooperative enhancement at the λ switch: *positive control* and *cooperative binding*.

Genes and promoters: The λ switch controls two *genes* cI and cro, illustrated Fig. 2. As for all other genes, transcription always starts at DNA segments called *promoters*, here P_{RM} and P_R respectively. Transcription of a gene eventually enables the production of the *protein* it encodes. RNA *polymerase* (RNAP) are molecules which can bind at promoters. Once bound to promoter P_{RM} a RNAP may initiate transcription of the cI gene, which subsequently allows for the production of new λ *repressor* proteins (Rep). An RNAP bound at P_R may start to transcribe the cro gene and thereby enables the production of Cro proteins[4].

Cro and Rep proteins appear in two forms, as *dimers* and *monomers* which can be distinguished in Fig. 3. When expressed they first appear as *monomers*. Subsequently these associate pairwise, and only in this form they can bind to DNA. Dimers are unstable, unless bound to DNA they soon dissociate back to

[4] P_{RM} stands for *promoter for synthesis of repressor during maintenance of lysogeny*. P_R simply stands for *right promoter*.



gene	promoter	protein
cI	P_{RM}	Rep
cro	P_R	Cro

Fig. 2. A spatial view on the λ switch, a segment of phage lambda's genome. The gene cI is transcribed from the promoter P_{RM}; it encodes the regulatory protein Rep. Its antagonist protein Cro is transcribed from promoter P_R. The operator regions O_{R1} and O_{R2} lie within P_R, while P_{RM} overlaps with O_{R3}.

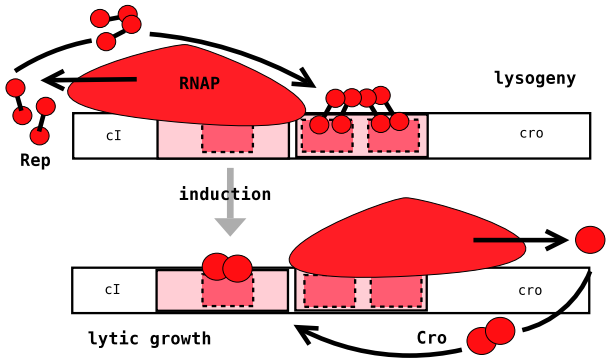


Fig. 3. Network states during lysogeny and lytic growth. In lysogeny, Rep attached to either or both of the binding sites O_{R1} and O_{R2} blocks recognition of the promoter P_R by RNAP, and thus prevents transcription of the gene *cro*. At the same time, interactions between Rep at O_{R2} and RNAP at P_{RM} stimulate transcription of the gene `cI`, which allows for the production of new Rep.

monomers. The higher the protein concentration in the cell, the higher is the degree of dimerization.

In lysogeny, the state of the network is characterized by a high number of Rep and negligible amount of Cro proteins; these frequencies are inverted during lytic growth. The environmental signal upon induction leads to a massive destruction of Rep proteins. P_R then becomes activated automatically, while transcription from P_{RM} ceases. These are consequences of the network controlling transcription initiation.

Repression of promoters by steric hindrance: The regulatory proteins Rep and Cro can bind to three neighboring *operator* regions O_{R1}, O_{R2}, and O_{R3}. By doing so, they control RNAP access to the promoters. As Fig. 2 indicates O_{R1} and O_{R2} both overlap the promoter P_R, while O_{R3} lies within P_{RM}. A protein bound within a promoter blocks recognition of the promoter by RNAP. This principle is called *steric hindrance*. The typical constellations are sketched in Fig. 3. Note that all bindings are *reversible*, i.e. the proteins dissociate from the DNA strand after some time. RNAP frequently falls off a promoter without initiating transcription.

The maintenance of lysogeny depends on the presence of a sufficient amount of repressor, that is predominantly bound at O_{R1} and O_{R2}. This impedes RNAP binding to P_R. As a consequence, Cro and all other viral genes are not expressed.

Cooperative enhancement of repressor binding at O_{R2} : The intrinsic binding affinity of Rep for O_{R1} is tenfold higher than for O_{R2} and O_{R3}. Thus, Rep is likely to be found at O_{R1}. Furthermore, Rep at O_{R1} significantly favors binding of another Rep to O_{R2} – this is what we call *cooperative binding*. One could say that the λ repressor at O_{R1} *recruits* another to O_{R2} [36].

Positive control of transcription initiation is needed for virtually all genes [36]. It refers to the fact that RNAP bound to a promoter needs the help of regulatory proteins in order to successfully initiate transcription. At P_{RM}, RNAP's frequency increases due to a direct contact with Rep bound at O_{R2}. This second instance of cooperative enhancement, called *positive control*, is decisive for maintaining the lysogenic state. Without it RNAP would rather fall off the inherently weak promoter P_{RM} than start to transcribe.

The production of Rep ceases once its level allows to fill not only O_{R1} and O_{R2}, but also the last site O_{R3}. At this point Rep inhibits its own production by steric hindrance of P_{RM} in a negative feedback loop.

Upon *induction*, the number of repressors rapidly decreases due to an external signal, so that O_{R1} and O_{R2} become more and more likely to remain vacant. Now polymerases find frequent opportunities to bind to P_R. As P_R is inherently a *strong promoter*, these bindings rapidly ensue transcription, followed by the production of Cro proteins.

3 Stochastic Parametrization

The stochastic π-calculus assumes rates that determine the speed of reactions. In this section, we discuss how to distill such rates from the literature. The resulting parameters are summarized and given mnemonic names in Figure 4.

In our system reversible binding reactions are frequent, such as by λ repressor to the operator O_{R1}:

$$\text{Rep} + \text{O}_{\text{R1}} \underset{k_d}{\overset{k_a}{\rightleftharpoons}} \text{Rep} \cdot \text{O}_{\text{R1}} \tag{1}$$

This bidirectional reaction converges to an equilibrium, in which the number of reactants on both sides remains constant. The *association* constant k_a determines the speed of the association reaction. It measures the number of Rep-O_{R1}-pairs that form complexes per mol and second. For the case of regulatory proteins the *association* rate constant k_a has been experimentally determined [6, 47, 44]. It is given by the net rate with which a protein locates its target site on DNA:

$$k_a = \frac{10^8}{\text{mol sec}} \tag{2}$$

We assume this value for all combinations of proteins and sites [5].

[5] This constant exceeds three dimensional diffusion by two orders of magnitude, and subsumes a number of mechanisms of target site location by proteins. In its search

event	rate	name	reference
dissociation of $\text{R}_{\text{NAP}} \cdot \text{P}_{\text{RM}}$	0.788	`Kd_RNAP_PRM`	[24]
dissociation $\text{R}_{\text{NAP}} \cdot \text{P}_{\text{R}}$	0.155	`Kd_RNAP_PR`	[16]
dissociation of $\text{Rep} \cdot \text{O}_{\text{R1}}$	0.155	`Kd_or1_rep`	[1]
dissociation of $\text{Rep} \cdot \text{O}_{\text{R2}}$	3.99	`Kd_or2_rep`	[1]
dissociation of $\text{Rep} \cdot \text{O}_{\text{R2}}$, coop.	0.155	`Kd_or2_rep_coop`	[41]
dissociation of $\text{Rep} \cdot \text{O}_{\text{R3}}$	20.22	`Kd_or3_rep`	[22]
dissociation of $\text{Cro} \cdot \text{O}_{\text{R1}}$	2.45	`Kd_or1_cro`	[41]
dissociation of $\text{Cro} \cdot \text{O}_{\text{R2}}$	2.45	`Kd_or2_cro`	[41]
dissociation of $\text{Cro} \cdot \text{O}_{\text{R3}}$	0.29	`Kd_or3_cro`	[41]
association of protein to operators	0.098	`Ka_protein`	[6]
association of R_{NAP} to promoters	0.098	`Ka_RNAP`	[47]
promoted transcription from P_{RM}	0.086	`Kf_prm_promoted`	[24]
transcription from P_{RM}	0.005	`Kf_prm`	[24]
transcription from P_{R}	0.05	`Kf_pr`	[16]
association of repressor monomers	0.048	`ka_repDimer`	[8]
dissociation of repressor dimers	0.5	`kd_repDimer`	[8]

Fig. 4. Stochastic parameters for molecular events at the λ switch

The *dissociation* constant k_d specifies the speed of the de-complexation. It measures the proportion of complexes that is resolved per second. As we will see, for the case of Rep binding to O_{R1} we can assume it to be $k_d = \frac{0.155}{\text{sec}}$.

However, it is less obvious to infer such dissociation rates the literature. What is determined experimentally for such reactions is mostly *Gibbs free energy* ΔG – a notion from thermodynamics. The value of ΔG quantifies the effort necessary for decomplexation. In the concrete example of O_{R1}, Shea and Ackers [1] provide $\Delta G = -12.5 \frac{kcal}{\text{mol}}$. This energy is negative, reflecting that binding requires an effort by the environment, while unbinding happens voluntarily. Non cooperative binding of Rep at the weaker binding site O_{R2} yields a value of $\Delta G = -10.5 \frac{\text{kcal}}{\text{mol}}$, for O_{R3} we obtain $\Delta G = -9.5 \frac{\text{kcal}}{\text{mol}}$. Note that a smaller value indicates *stronger* binding, and that a difference of $1kcal$ ensues a tenfold difference in binding strength.

Gibbs free energy correlates with the equilibrium constant K_{eq} of the binding reaction, which expresses the quantities of unbound pairs Rep and O_{R1} compared to complexes $\text{Rep} \cdot \text{O}_{\text{R1}}$ in equilibrium. The relationship is expressed through the equation:

$$K_{eq} = \exp(\frac{-\Delta G}{R \cdot T}) \tag{3}$$

where $R = 1.9872 \frac{cal}{\text{mol Kelvin}}$ is the universal gas constant and $T = 310.15$ Kelvin is the absolute temperature at which the experiments were performed (it corresponds to 37 Celsius).

process a protein first diffuses three-dimensionally trhough the cytoplasm, hits the DNA and subsequently slides along the DNA, rapidly scanning it for its specific site. A model explaining this has been proposed in [43].

The equilibrium constant K_{eq} represents the ratio of association and dissociation rate constants as shows the following kinetic equation:

$$K_{eq} = \frac{k_a}{k_d} \text{ mol} \tag{4}$$

The experimental data on Gibbs energy together with equations (2), (3), and (4) are sufficient to compute the dissociation rate k_d by straightforward arithmetics[6].

The rate constants k_a and k_d we have met so far are *macroscopic* – as in chemical kinetics. They do not depend on the actual numbers of molecules, but on concentrations. Gillespie's algorithm, however, and thus the biochemical stochastic π-calculus use *mesoscopic* rate constants as their stochastic rates. These refer to actual numbers of molecules and are determined from their macroscopic counterparts as follows:

$$k_a^{meso} = \tfrac{k_a}{A\ V}, \qquad k_d^{meso} = k_d,$$

where $A = 6.023 \cdot 10^{23}$ is Avogadro's number – i.e. number of molecules per mole – and $V = 1.7 \cdot 10^{-15} l$ is the E. coli cell volume. We need to divide by $A \cdot V$ for reactions involving two reactants, such as binding; for reactions that transform a single reactant as unbinding, the macroscopic and mesoscopic rates coincide. Note that we assume the cell volume to be constant while ignoring cell growth. Evaluating our equation yields the following final rates for the considered example reaction between O_{R1} and Rep:

$$k_a^{meso} = 0.098/\,\text{sec} \qquad k_d^{meso} = 0.155/\,\text{sec}$$

We can now quantify the effects of *cooperative binding* between repressors at O_{R1} and O_{R2}. Cooperativity adds a favorable term of $-2\frac{\text{kcal}}{\text{mol}}$ to the Gibbs binding energy of Rep at O_{R2} [41] [7]. Due to the exponential relation between free energies and equilibrium constants this massively strengthens the binding: the mesoscopic dissociation rate k_d for O_{R2} decreases from 3.99 to 0.155, the same value as for O_{R1}. Table 5 summarizes.

Finally, we need rates for transcription initiation, in which a complex of RNAP and promoter P undergoes an irreversible transition from a closed state into an open one, in which the two strands of DNA have locally been separated, and after which transcription proceeds [25]:

$$\text{RNAP} + P \rightleftharpoons^{K_{eq}} (\text{RNAP} \cdot P)_{\text{closed}} \ \rightarrow_{k_f} \ (\text{RNAP} \cdot P)_{\text{open}} \tag{5}$$

[6] The following set of equations determines the values of all rates for the example:

$$\begin{aligned} \Delta G &= -12.5 \cdot 10^3 \text{cal/mol} & K_{eq} &= \exp^{-\Delta G/(R\ T)} \\ R &= 1.9872 \text{ cal/(mol Kelvin)} & k_a &= 10^8/(\text{mol } sec) \\ T &= 310.15 \text{ Kelvin} & k_d &= k_a/K_{eq} \text{ mol} \end{aligned}$$

[7] Cooperativity also has a helping effect to binding at O_{R1}, however we chose to neglect this in our model as the effect at O_{R2} predominates.

	ΔG	k_d	binding strength
O_{R1}	-12.5	0.155	strongest
O_{R2} (coop)	-12.5	0.155	
O_{R2} (isolated)	-10.5	3.99	
O_{R3}	-9.5	20.22	weakest

Fig. 5. Parameters for binding of λ repressor to the three operator regions.

Processes	P	$::= \mathtt{0}$	idle
		$\mid P_1 \mid P_2$	concurrent composition
		$\mid (\mathtt{new}\ x(r))\ P$	channel creation
		$\mid A(\overline{y})$	parametric process
		$\mid \pi_1, P_1 + \ldots + \pi_n, P_n$	choice
		$\mid \mathtt{if}\ x{=}y\ \mathtt{then}\ P_1\ \mathtt{else}\ P_2$	conditional
Prefixes	π	$::= x!\{\overline{y}\}$	polyadic output
		$\mid x?\{\overline{z}\}$	polyadic input
Definitions	D	$::= A(\overline{y})\ \mathtt{::=}\ P.$	

Fig. 6. Syntax of the stochastic π-calculus, where $\overline{y} = y_1, \ldots, y_n$ and $\overline{z} = y_1, \ldots, y_n$.

The k_f rates for the promoter P_R and P_{RM} can be found in [16, 24]. *Positive control* of RNAP by repressor binding at O_{R2} increases the k_f rate of P_{RM} roughly tenfold. Note that the dissociation rate of RNAP binding at P_{RM} is not affected, which distinguishes this mechanism from cooperative binding of regulatory proteins.

Throughout this paper we assume a constant RNAP concentration of $c = 30 \cdot 10^{-9}$ mol according to [41]. This corresponds to a population of circa 30 RNAP molecules via the simple calculation $\#RNAP = c \cdot V \cdot A = 30.7$, with A and V as above.

Finally, we assume the rate at which repressor monomers associate to dimers to be $0.025\,\mathrm{sec}^{-1}(nM)^{-1}$ while setting the dissociation rate to $0.5/\,\mathrm{sec}$ following [8].

4 Stochastic Pi Calculus

We now recall the variant of the stochastic π-calculus [33] that is the core language underlying the *BioSpi* simulation engine [34].

Figure 6 lists the syntax of our stochastic π-calculus. The vocabulary consists of an infinite set of *channel names* x, y, z, an infinite set of *process names* A, B, C and *stochastic rates* r that are nonnegative floating point numbers. We write $\overline{y}$ for finite, possibly empty sequences of channels.

Parallel compositions $P_1 | \ldots | P_n$ are processes with parallel subprocesses P_1, $\ldots, P_n$. The composition operator is associative and commutative, so that the ordering of composition is irrelevant. The empty parallel composition where $n = 0$ is the idle process $\mathtt{0}$, the neutral element of composition. Processes $(\mathtt{new}\ x(r))P$

$$A(\overline{z}) \quad \rightarrow \quad P[\overline{z}/\overline{y}] \qquad \text{with respect to } A(\overline{y}) \texttt{ ::= } P$$

$$\left.\begin{array}{l} \ldots + x?\{\overline{y}\}, P + \ldots \mid \\ \ldots + x!\{\overline{z}\}, P' + \ldots \end{array}\right\} \quad \rightarrow \quad P[\overline{z}/\overline{y}] \mid P' \qquad \text{if } \overline{z} \text{ free for } \overline{y} \text{ in } P$$

$$\texttt{if } x{=}y \texttt{ then } P_1 \texttt{ else } P_2 \quad \rightarrow \quad \begin{cases} P_1 \text{ if } x = y \\ P_2 \text{ if } x \neq y \end{cases}$$

Fig. 7. Reduction rules, where $\overline{y} = y_1, \ldots, y_n$, $\overline{z} = y_1, \ldots, y_n$, and $n \geq 0$.

define a new channel x with scope P, similarly to an existential quantifier $\exists x.P$; this new channel x is associated with the stochastic rate r.

Definitions of parametric processes $A(\overline{y})\texttt{::=}P$ associate the name A to a process P with free channel names $\overline{y}$, the parameters of A. Parametric definitions are universally valid for all parameter choices. They may be recursive, i.e. contain self applications. An application of a parametric process $A(\overline{z})$ calls the process named A with channels $\overline{z}$. Formally, $A(\overline{z})$ preceeds by unfolding the definition of A while substituting $\overline{z}$ for $\overline{y}$, according to the first reduction rule in Fig. 7.

A conditional[8] $\texttt{if } x{=}y \texttt{ then } P_1 \texttt{ else } P_2$ tests for equality between channels x and y; if equality holds it reduces to P_1, otherwise to P_2. As an example, consider the definition: $A(x, y) \texttt{ ::= if } x{=}y \texttt{ then 0 else } A(y, x)$ which is valid for all channels x, y. With respect to this definition, we can reduce the process $A(z, z) \rightarrow \texttt{0}$ for all z, while if $z_1 \neq z_2$ we have infinite reduction chains $A(z_1, z_2) \rightarrow A(z_2, z_1) \rightarrow A(z_1, z_2) \rightarrow \ldots$.

Choices $\pi_1, P_1 + \ldots + \pi_n, P_n$ offer synchronous communication and nondeterminism. Two choices composed in parallel can communicate with each other if one of them contains an output capacity $x!\{\overline{z}\}, P'$ and the other some input capacity $x?\{\overline{y}\}, P$ for the same channel x. The result of this communication act will be $P' \mid P[\overline{z}/\overline{y}]$ where $\overline{z}$ is substituted for $\overline{y}$ in P. Communication over the channel x lets an output capacity for x send a tuple of channels $\{\overline{z}\}$ to an input capacity for x, which waits for such data to replace its tuple of formal parameters $\{\overline{y}\}$.

A complete *program* consists of a set of `public` channel declarations, a set of definitions, and an initial process P that is to reduced with respect to these declarations and definitions.

Let us express chemical reactions for illustration. We consider two competing reactions of type x_1 and x_2 with rates r_1 and r_2:

$$x_1 : A + B \rightarrow^{r_1} C_1$$
$$x_2 : A + B \rightarrow^{r_2} C_2$$

[8] BioSpi supports conditionals as sums of match prefixes. For better readability we adopt an alternative notation with keywords `if then else`.

We encode the two rules types x_1 and x_2 as global channel with rates r_1 and r_2 and define A, B as parametric processes without parameter:

$$\begin{array}{l}\texttt{public}\ (x_1(r_1), x_2(r_2)).\\ A\ \texttt{::=}\ x_1!\{\}, C_1\ + x_2!\{\}, \texttt{0}.\\ B\ \texttt{::=}\ x_1?\{\}, 0 + x_2?\{\}, \texttt{C}_2.\end{array}$$

We now compose many molecules of types A and B in parallel. Each A-B-pair can decide to react, either according to rule x_1 which reduces in the following manner:

$$A \mid A \mid B \mid B \quad \rightarrow \quad C_1 \mid A \mid \texttt{0} \mid B \quad \rightarrow \quad \ldots$$

If alternatively rule x_2 happened to be applied, one could observe:

$$A \mid A \mid B \mid B \quad \rightarrow \quad 0 \mid A \mid \texttt{C}_2 \mid B \quad \rightarrow \quad \ldots$$

All channels are associated with a stochastic rate, that is either introduced by `public` declaration or the `new` operator. Such rates define exponential distributions that characterise the communication activity of the channel (see [33, 34]). Communications over channels with `infinite` rate are executed instantaneously, as are conditionals and channel creations. Channels with finite rates communicate only afterwards. The scheduling of communication acts over these channels is based on Gillespie's algorithm [14].

```
public (pro(ka_protein), release(kd_OR_A)).

OR_vacant ::= pro ? {}, OR_bound.
OR_bound  ::= release ! {}, OR_vacant.
A_unbound ::= pro ! {}, A_bound.
A_bound   ::= release ? {}, A_unbound.

System ::= OR_vacant | A_unbound | A_unbound | A_unbound.
```

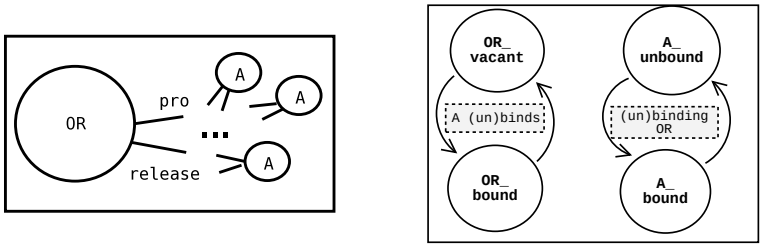


Fig. 8. An operator region and three regulatory proteins: expressing many-to-one communication over global channels. The topology of the system is shown on the left, and the state transition diagrams of molecular actors of type A and OR are given on the right.

5 Modeling the Network of Transcription Initiation

We formally model the network controlling transcription initiation at the λ switch in the stochastic π-calculus. We start with three simpler subsystems, before turning to modeling the λ switch system as described in Section 2. Following Regev and Shapiro's guidelines [39], we represent members of the biomolecular population as processes, and biomolecular events as communication.

5.1 Modeling techniques in system components

We start with a case of many-to-one communication, which is the simplest subsystem to model. We consider a network with a unique operator region on DNA of whatever type `OR` and many proteins of the same type `A` that can attach to it. The operator has two states `vacant` and `bound`; the possible states of the proteins are `bound` and `unbound`.

We use the four possible combinations of molecule types with their states as names of parametric processes: `OR_vacant`, `OR_bound`, `A_unbound`, `A_bound`. We introduce two global channels, `pro` for reactions of protein binding to the operator, and `release` for unbinding events. The rate of `pro` is the association rate `ka_protein` that is invariant for all types of operators and proteins. The rate of `release` is the dissociation rate `kd_OR_A`, which depends on the specificity between protein and operator. Figure 8 presents the definitions of all agents in the system in the stochastic π-calculus, the topology and state transitions.

As simple as this example may seem, it is already sufficient for simulating binding and unbinding of either Rep or Cro at *isolated* operator sites O_{R1}, O_{R2}, or O_{R3} which are then distinguished by their dissociation rates.

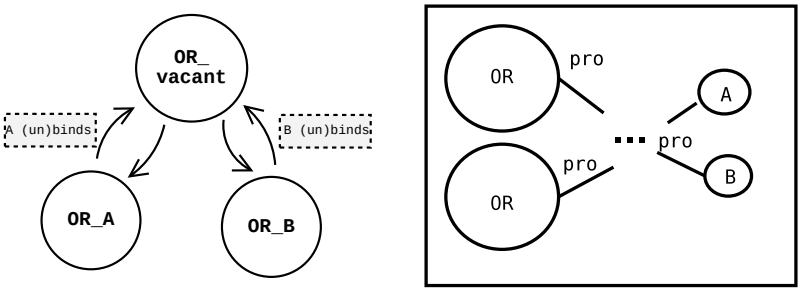


Fig. 9. Left: Operator region with distinct states when binding different proteins, right: two operator regions alongside with two proteins of different types

Many-to-many communication and handshakes: We next consider a case of many-to-many communication, for which we introduce a less simple handshake protocol. We study a system with two operators of the same type `OR` that can be bound by proteins of two different types `A` and `B`. We wish to design our

```
public (pro(ka_protein), a, b).

OR_vacant ::= pro ? {type,init},
              if type=a then (new release(kd_OR_A))
                              init ! {release}, OR_A(release)
                        else (new release(kd_OR_B))
                              init ! {release}, OR_B(release).
OR_A(release) ::= release ! {}, OR_vacant.
OR_B(release) ::= release ! {}, OR_vacant.

A_unbound ::= new(init(infinite))
              pro ! {a,init}, init ? {release}, A_bound(release).
B_unbound ::= new(init(infinite))
              pro ! {b,init}, init ? {release}, B_bound(release).
A_bound(release) ::= release ? {}, A_unbound.
B_bound(release) ::= release ? {}, B_unbound.

System ::= OR_vacant | OR_vacant | A_unbound | B_unbound.
```

Fig. 10. Modeling two operator regions with two proteins of different types.

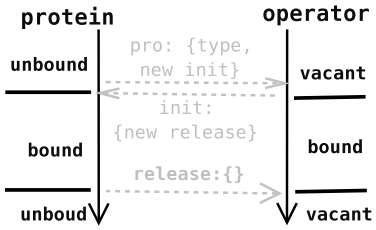


Fig. 11. Handshake protocol for protein binding to operator. Solid black arrows denote the time flow; dotted grey arrows denote communication, the annotations indicate the names of channel used and exchanged. Horizontal lines indicate the time point of state changes of corresponding molecular actor.

model such that all knowledge about binding parameters is localized within operator sites. With this all proteins can bind operators in the same generic manner, only depending on their types. We thus define operators of type `OR` with three possible states: `vacant`, `A`, and `B`. Fig. 9 illustrates their state transitions. The states of proteins of types `A` and `B` remain as previously introduced: `unbound` and `bound`.

We obtain seven names for parametric processes, when building all possible pairs of molecular types with their states: `A_bound, A_unbound, B_bound, B_unbound, OR_vacant, OR_A, OR_B`. We again use a unique global channel `pro` for protein binding. The stochastic rate of `pro` is the association rate `ka_protein` that is invariant for all types of proteins and operators. In addition, we introduce two channels `a` and `b` with arbitrary rates that encode the protein types.

Recall that dissociation rates are specific for each combination of proteins and operators. In addition to that, the rate of a channel is fixed upon its creation. Hence we need a dedicated `release` channel of appropriate rate per possible combinations of protein and operator. Furthermore, these channels should be introduced by the operators, where the knowledge on all interaction parameters is localized. The biological motivation for this is that the specificity of bindings depends on the operator's sequence. The better a protein matches this, the higher the specificity of binding. Fig. 10 presents the definitions of all agents in the system.

We deploy a handshake protocol illustrated in Fig. 11: binding is initiated by the protein, which transmits to the operator its type and a freshly created private channel of name `init`. The operator creates a new `release` channel upon each binding, that bears the suitable stochastic rate depending on the protein type - and hands it over to the protein using `init`. Subsequent dissociation occurs with the specific rate.

This generic models needs only slight generalization to apply to the interactions of multiple proteins Cro and Rep with different kinds of operator regions O_{R1}, O_{R2}, and O_{R3}. What it doesn't reflect yet are cooperative interdependencies between binding events, or mutual exclusion of binding at spatially overlapping sites.

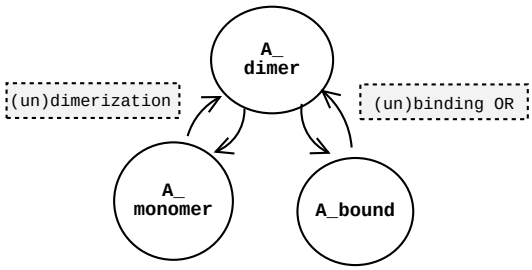


Fig. 12. State transition diagram for generic protein A with dimerization and binding

Timers: In our third case, we utilize timers as proposed by Regev [37]. Timers serve for auxiliary purposes, they don't have a biological equivalent.

Their sole purpose is to trigger an activity performed by a single molecular actor. Timers wait until a partner is ready to communicate over some specific channel. Such catalysts are needed for modeling first-order reactions in the π-calculus, where all actions necessitate precisely *two* participants.

An example is to apply timers for dissociating complexes, here in the case of *dimers*. We consider a system with modified protein `A`. The protein `B` and operator site `OR` remain as introduced previously. We distinguish between `A` monomers and `A` dimers, and enable only dimers to bind to `OR`. Proteins of type `A` hence have states `monomer`, `dimer`, and `bound`, Fig. 12 illustrates the transitions between these. Operator regions of type `OR` are either `vacant` or in bound states `A` or `B` as previously. The definition of the system is given in Fig. 13.

```
public (pro(ka_protein),
       dimerize_A(ka_A_Dimer), undimerize_A(kd_A_Dimer)).

A_monomer ::= dimerize_A ! {}, A_dimer
            + dimerize_A ? {}, 0.
A_dimer   ::= (new init(infinite))
              pro ! {a,init}, init ? {release}, A_bound(release)
            + undimerize_A ? {}, A_monomer | A_monomer.
A_bound(release) ::=  release ? {}, A_dimer.

OR_vacant ::= pro ? ... # rules for OR_A, OR_B as before

Timer(c) ::= c ! {}, Timer(c).

System := Timer(undimerize_A) |
          A_monomer | A_monomer | A_monomer | OR_vacant.
```

Fig. 13. Modified system with timer for dissociation of dimers into monomers

For every type of protein `A` we use two global channels `dimerize_A` and `undimerize_A`. Every `A` monomer has the choice to read or write on the channel `dimerize_A`. We have chosen somehow arbitrarily that the writer continues as a dimer, while the reader dies. In order to undimerize, a dimer of type `A` interacts with `Timer(undimerize_A)`, and dissociates back into two monomers of type `A`.

5.2 Modeling the λ Switch

The molecular population is summarized in Fig. 14. In a λ infected E. coli cell, we assume precisely one copy of each P_{RM}, P_R, O_{R1}, O_{R2}, and O_{R3}, disregarding replication. Alongside with these reside a large number of RNAP, and variable numbers of the regulatory proteins Rep and Cro.

Molecular actors are connected by public channels in Fig. 14 that are declared and assigned stochastic rates in Fig. 15. The channel `rnap` is used both for RNAP docking to P_{RM} and P_R, and is assigned the rate `Ka_RNAP`. Regulatory

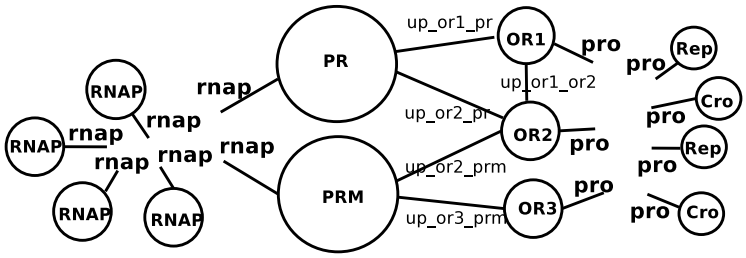


Fig. 14. The molecular population with its communications channels.

```
public (rnap(Ka_RNAP), pro(Ka_protein),
       up_or1_pr(infinite), up_or2_pr(infinite),
       up_or2_prm(infinite), up_or3_prm(infinite),
       up_or1_or2(infinite),
       vacant, blocked, inhibited, cro, rep, polymerase).
```

Fig. 15. Public channels with their stochastic rates

proteins establish connections to operator regions over the channel `pro` with association rate `Ka_protein`. Since both interactions are of many-to-many type, all establishment of bindings follow our handshake protocol.

P_{RM}	P_R	O_{R1}, O_{R3}	O_{R2}	`Rep`, `Cro`	R_{NAP}
vacant	vacant	vacant	vacant	unbound	unbound
rnap_high	rnap	cro	cro	bound	bound
rnap_low	inhibited	rep	rep_high	monomer	
inhibited		blocked	rep_low		
			blocked		

Fig. 16. States of molecular actors.

The possible *states of molecular actors* are listed in Table 16. Particular channels are used to communicate *state updates*. We declare these update channels `up_A_B` with rate `infinite`. They will be used to transmit *state update messages*, whose names indicate the new state the sender is switching to: `vacant`, `blocked`, and alike. These last are encoded by channels of arbitrary rates.

We use update channels for synchronization purposes, modeling cooperativity for instance between O_{R1} and O_{R2}, as well as for implementing mutual inhibition. Consider P_R, where binding is mutually exclusive with O_{R2}. The frequency of transcription initiation at P_{RM} depends on O_{R2}'s state - Rep present there exerts positive control on it. Binding of Rep to O_{R2} in turn can be cooperatively strengthened, which depends on whether another Rep is bound to O_{R1}.

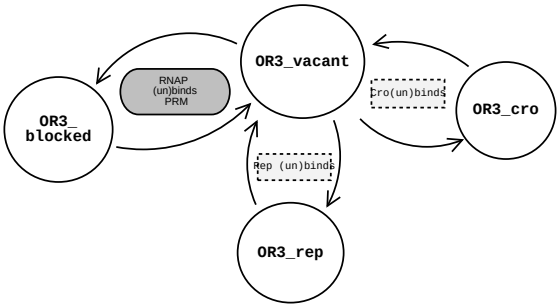


Fig. 17. State transitions of O_{R3}. Transitions can be caused locally, or follow as side effects of events at other molecular actors.

```
OR3_rep(release) ::= release ? {}, up_or3_prm ! {vacant}, OR3_vacant.
OR3_cro(release) ::= release ? {}, up_or3_prm ! {vacant}, OR3_vacant.
OR3_blocked  ::=  up_or3_prm ? {new_prm}, OR3_vacant.
OR3_vacant ::=
pro ? {type,init},
      if type=rep then (new release(Kd_or3_rep))
         init ! {release}, up_or3_prm ! {rep}, OR3_rep(release)
      else             (new release(Kd_or3_cro))
         init ! {release}, up_or3_prm ! {cro}, OR3_cro(release)
+ up_or3_prm ? {new_prm},
       if new_prm=polymerase then OR3_blocked else OR3_vacant.
```

Fig. 18. Specification of O_{R3} module

Handshake and state updates at O_{R3}: the model for the generic operator region from Fig. 10 was sufficient for a first interaction with Rep and Cro. However, we need specializations in order to appropriately reflect any of the actual operator regions at the λ switch.

Consider the operator site O_{R3}. It can be either `vacant`, `blocked` by a RNAP at P_R, occupied by a Cro protein (state `cro`), or Rep. Figure 17 illustrates O_{R3}'s transitions. Note the introduction of a second category of annotations, not yet present in former examples: transitions to and from `inhibited` are triggered by events at P_{RM}. Recall that bindings of RNAP to P_{RM} and regulatory proteins at O_{R3} mutually exclude each other. We thus keep the states of O_{R3} and P_{RM} consistent by instantaneous state updates, for which we reserve a dedicated *update channel* `up_or3_prm`.

The π-calculus implementation of O_{R3} is given in Fig. 18. Besides being one of the possible counterparts for protein binding and implementing the handshake protocol, the code comprises state updates: when `OR3` releases a bound protein and is about to depart from either of its states `rep` or `cro`, it notifies `PRM` (which is currently in state `inhibited`) via `up_or3_prm`, and only after this continues as `OR3_vacant`. In analogy upon protein docking, `OR3_vacant` again updates P_{RM}. Alternatively, as soon as `PRM` gets docked by a polymerase and communicates this `up_or3_prm`, `OR3_vacant` switches to `OR3_blocked`.

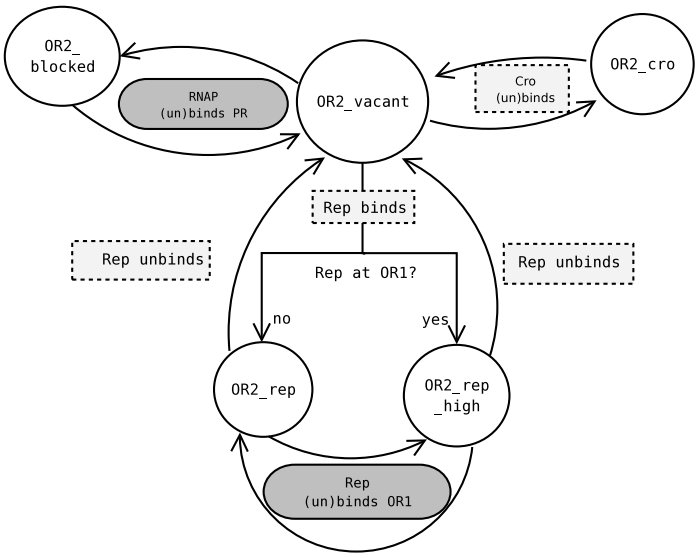


Fig. 19. State transitions of the O_{R2} model

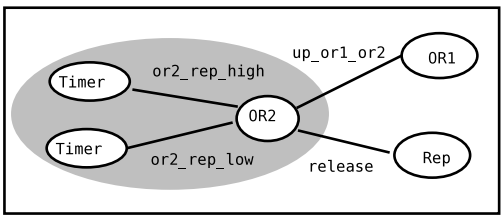


Fig. 20. `OR2` with auxiliary timers for the adjustment of binding strength for repressor, depending on `OR1`'s state as notified over `up_or1_or2`.

```
public(or2_rep_high(Kd_or2_rep_coop),
    or2_rep_low(Kd_or2_or2)).

OR2_vacant(or1) ::=
  pro ? {id,init},
  if   id=cro
  then up_or2_prm ! {cro}, up_or2_pr ! {cro},
       (new release(Kd_or2_cro)) init ! {release},
       OR2_cro(or1,release)
  else up_or2_prm ! {rep}, up_or2_pr ! {rep},
       (new release(infinite)), init ! {release},
       if or1=rep
       then OR2_rep_high(release)
       else OR2_rep_low(or1,release)
+ up_or1_or2 ? {new_or1},
       OR2_vacant(new_or1)
+ up_or2_pr ? {new_pr},
       if new_pr=polymerase
       then up_or2_prm ! {blocked}, OR2_blocked
       else OR2_vacant(or1).

OR2_cro(or1,release) ::=
  release ? {},
       up_or2_prm ! {vacant},
       up_or2_pr ! {vacant},
       OR2_vacant(or1)
+ up_or1_or2 ? {new_or1},
       OR2_cro(new_or1, release).


OR2_rep_high(release) ::=
  or2_rep_high ? {},
       release ? {},
       up_or2_prm ! {vacant},
       up_or2_pr ! {vacant},
       OR2_vacant(rep)
+ up_or1_or2 ? {new_or1},
       OR2_rep_low(new_or1, release).

OR2_rep_low(or1,release) ::=
  or2_rep_low ? {},
       release ? {},
       up_or2_prm ! {vacant},
       up_or2_pr ! {vacant},
       OR2_vacant(or1)
+ up_or1_or2 ? {new_or1},
       if   new_or1=rep
       then OR2_rep_high(release)
       else OR2_rep_low(new_or1, release).

OR2_blocked ::= up_or2_pr ? {c}, OR2_vacant(vacant).

Module ::= Timer(or2_rep_low)
       |   Timer(or2_rep_high)
       |   OR2_vacant(vacant).
```

Fig. 21. Specification of the O_{R2} module

Cooperative repressor binding at O_{R2} necessitates more sophisticated control than seen so far. Recall that we assigned it to the operator the task to determine the unbinding time point. In cases without cooperativity, the operator simply associates the appropriate dissociation rate to the `release` channel upon creation. For the case of cooperative repressor binding to `OR2`, we delay the reading offer on the `release` channel. This is performed by associating an `infinite` rate to `release`. `OR2` applies a delay determined differentially for cases with or without cooperativity. For this, `OR2` makes use of *two alternative timer processes*. When repressor binding is cooperative and `OR2` thus is in state `rep_high`, dissociation is triggered over channel `or2_rep_high`. Otherwise, it is determined by channel `or2_rep_low`, both bear a distinct rates determined in Sec. 3. `OR2` works by selecting appropriate states and switching between these as necessary. Figure 19 shows `OR2`'s state transitions, Fig. 20 illustrates the topology, for the full π-calculus specification see Fig. 21.

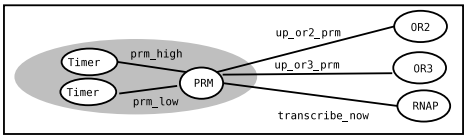


Fig. 22. Module abstracting P_{RM} with its two auxiliary timer processes. The frequency of transcription initiation by `RNAP` is controlled by channel `prm_high` for `PRM_rnap_high`; or via `prm_low` for state `PRM_rnap_low`. Transitions between these two states in turn depend on changes of state of `OR2`, and are synchronized over `up_or2_prm`.

Positive control at P_{RM}: switchable timers. We now come to modeling cooperative enhancement of transcription initiation of RNAP bound to P_{RM} by repressors at O_{R2}. The technique introduced at O_{R2} is useful again. This time, we model variation of the rate of transcription initiation with the help of switchable timers. We use two auxiliary timer processes communicating with P_{RM} over public channels `prm_high` and `prm_low` with rates from Table 4.

```
public(prm_high(Kf_prm_promoted),prm_low(Kf_prm)).
```

Fig. 22 illustrates our idea. For the full specification of the P_{RM} module see Fig. 23. Let us highlight the last paragraph, which defines the P_{RM} module as the concurrent composition of four processes: two timers, the promoter and the gene it controls.

```
Module ::=
Timer(prm_high) | Timer(prm_low) | Gene_cI | PRM_vacant(vacant).
```

Assume polymerase being docked to P_{RM} in our model. Transcription is then triggered over the instantaneous channel `transcribe_now`. Depending on the occupancy of `OR2`, `PRM` switches to one of two states representing the complex with `RNAP`. In the first, `PRM_rnap_low`, there is no `Rep` at `OR2` and P_{RM} hence

```
public(prm_high(Kf_prm_promoted),
       prm_low (Kf_prm),
       gene_cI(infinite).

PRM_vacant(or2) ::=
  rnap ? {init},
    if   or2=rep
    then (new transcribe_now(infinite), release(Kd_rnap_prm))
         init ! {transcribe_now, release, gene_cI},
         up_or3_prm ! {polymerase},
         PRM_rnap_high(transcribe_now,release)
    else (new transcribe_now(infinite), release(Kd_rnap_prm))
         init ! {transcribe_now, release, gene_cI},
         up_or3_prm ! {polymerase},
         PRM_rnap_low(transcribe_now,release,or2)
+ up_or2_prm ? {new_or2},
         PRM_vacant(new_or2)
+ up_or3_prm ? {new_or3},
         PRM_inhibited(or2) .

PRM_rnap_low(transcribe_now,release,or2) ::=
  prm_low ? {},
        transcribe_now ! {},
        up_or3_prm ! {vacant},
        PRM_vacant(or2)
+ up_or2_prm ? {new_or2},
       if new_or2=rep
       then PRM_rnap_high(transcribe_now, release)
       else PRM_rnap_low(transcribe_now, release, new_or2)
+ release ? {}, up_or3_prm ! {vacant}, PRM_vacant(or2).

PRM_rnap_high(transcribe_now,release) ::=
  prm_high ? {},
       transcribe_now ! {},  up_or3_prm ! {vacant},
       PRM_vacant(rep)
+ up_or2_prm ? {new_or2},
       PRM_rnap_low(transcribe_now, release, new_or2)
+ release ? {},
       up_or3_prm ! {vacant},
       PRM_vacant(rep).

PRM_inhibited(or2) ::=
   up_or2_prm ? {new_or2}, PRM_inhibited(new_or2)
+  up_or3_prm ? {new_or3}, PRM_vacant(or2).

Gene_cI ::= gene_cI ? {},
       mRNA_cI | Gene_cI.

Timer(c) ::= c ! {}, Timer(c).

Module ::=  Timer(prm_high)
       |    Timer(prm_low)
       |    Gene_cI
       |    PRM_vacant(vacant).
```

Fig. 23. Specification of P_{RM} module with cI gene

works only at basal rate. This is indicated by waiting for the `prm_low` timer to shoot. Alternatively `PRM` is in state `PRM_rnap_high` and listens to `prm_high` in order to trigger transcription initiation. Switching between both timers follows instantaneously upon update from `OR2`. In either case, the polymerase may also unbind.

```
RNAP_unbound ::=
  (new init(infinite)) rnap ! {init},
  init ? {transcribe_now, release, toGene},
  RNAP_bound(transcribe_now, release, toGene).

RNAP_bound(transcribe_now, release, toGene) ::=
  transcribe_now ? {}, toGene ! {}, RNAP_unbound
+ release ! {}, RNAP_unbound.
```

Fig. 24. Specification of `RNAP`

RNAP: To conclude we give our π-calculus model of RNAP in Fig.24. We introduce only two states, `bound` and `unbound`. Note that the handshake protocol deployed for RNAP binding to promoters slightly differs from that for regulatory proteins. This is the case because `RNAP` needs a channel `release` for simple unbinding, another `transcribe_now` for transcription to be kicked off, and `toGene` to be pointed at the right gene to transcribe. The behavior of RNAP as sketched here is simplified. For `RNAP`'s behavior beyond the initiation of transcription, which is out of this present paper's scope, the model needs to be extended (see [23]). The same holds for the specification of the genes, e.g. the process `Gene_cI` within the `PRM` module needs to be replaced in order to obtain an appropriate model of transcription.

6 Stochastic Simulation

We next validate our π-calculus model of the dynamics at the λ switch by exhaustive stochastic simulation. These are performed by execution with the BioSpi system [34].

We use a sequence of models of distinguished subsystems in order to evaluate the different components independently. Given its complexity, it does not make sense to directly start with the complete system. From the software engineering perspective, this is necessary for debugging reasons. From the biological standpoint, it is a current practice to isolate subsystems in order to observe their aspects as independently as possibly. The π-calculus programming approach is advantageous in that perspective, in that it allows to freely design and compose subsystems of interest.

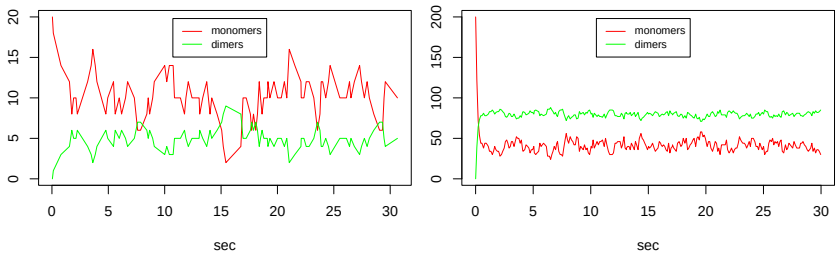


Fig. 25. Dynamics of formation and breakage of λ repressor dimers over 30 simulated seconds. Initiation with 20 monomers (left) or 200 monomers (right).

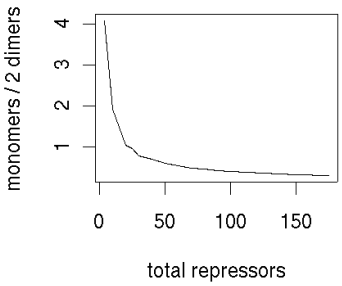


Fig. 26. Shift of concentration dependent equilibrium between monomers and dimers

6.1 Simulating Components

We thus perform simulations of subsystems that can be compared against existing knowledge, either experimental or from established other simulation studies. Our strategy is incremental and bottom up. First we present simulations of repressor dimerization. We then move over binding of Rep to DNA, and the impact of dimerization on binding patterns to that of cooperative interaction between Rep *on* DNA. Finally we investigate interactions between RNAP, DNA, and Rep's positive control thereof. The control of transcription initiation from P_{RM} is highly relevant; it has been subject to a number of theoretical and experimental studies [4, 5, 24, 41].

Dimerization: The essential point to remind about repressor dimerization was the concentration dependent equilibrium [35]; we can observe this in simulations. Figure 25 visualizes the dynamics of the dimerization process starting with different numbers of monomers. As an example, for the case of three monomers the starting point would be the parallel composition of three instances of `Rep_monomer` with a timer for dimer dissociation:

```
System ::= (new rep_undimerize(kd_repDimer),
            rep_dimerize(ka_repDimer))
            Timer(rep_undimerize) |
            Rep_monomer | Rep_monomer | Rep_monomer.
```

When launched with 20 monomers, such a system tends towards a mean setting in which around around half the total repressors can be found as monomers, while the others are present as dimers. Note that in this case, one observes strong fluctuations (see Fig.25 left). Only a rough quarter of initially 200 monomers are present as such in average, while around 75% are dimer-bound - with less important fluctuations. The *shift* of the equilibrium towards dimers becomes more obvious as we plot the average ratio of repressors present as monomers to that of dimer-bound ones over a long time range for various levels of repressors, see Fig. 26.

Fig. 27. Binding to isolated operator sites, assuming 100 repressor monomers.

	ΔG	mean sojourn	bound
O_{R1}	−12.5	6.4	96 %
O_{R2}	−10.5	0.25	46 %
O_{R3}	−9.5	0.05	15 %

6.2 Repressor binding to DNA

We now consider the binding of repressor dimers to operator sites on DNA. The following set up allows to simulate site O_{R1} and three repressor dimers that can reversibly attach to it, or dissociate back to monomers:

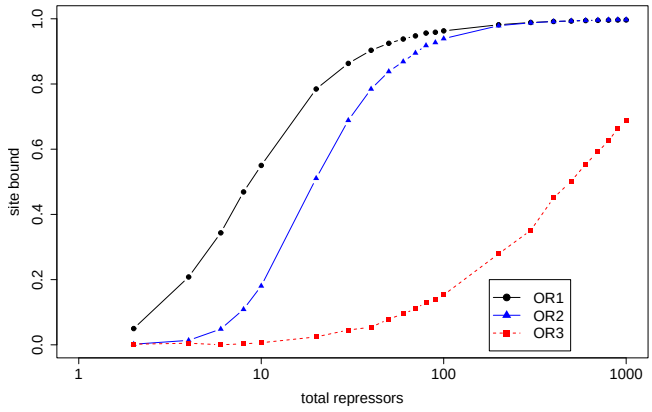


Fig. 28. Occupancy of sites O_{R1}, O_{R2} and O_{R3} in the presence of dimerization and cooperative binding. Each point represents the relative occupancy of a site over 5000 simulated seconds, for a given number of total repressors. In a lysogen, one can expect 100-200 total repressors.

```
System ::= (new dimerize(ka_repDimer), rep_undimerize(kd_repDimer),
            bind(Ka_protein), release(Kd_or1_rep))
            OR | Timer(rep_undimerize) |
            Rep_unbound | Rep_unbound | Rep_unbound .
```

By adjustment of the rate for channel `release`, we can simulate binding to the *isolated* sites O_{R1}, O_{R2} and O_{R3}. Figure 27 summarizes corresponding simulations, emphasizing the impact of different binding site site affinities. Recall from Sec. 3 that a smaller value of the Gibbs free energy ΔG indicates a stronger binding.

We make two corresponding observations. The complex of repressor and operator site is most stable at O_{R1}, where we observe an average sojourn time of Rep of 6.4 seconds (this value is the mean of an exponential distribution not shown here). This is consistent with [35], reporting that binding of repressor to O_{R1} persists in the order of up to 10 seconds. For O_{R2} and O_{R3} less favorable ΔGs lead to drastic drops of complex stability.

The efect is also visible when considering not individual binding events, but average behaviour. For a given concentration and a long time scale, O_{R1} is better saturated with repressor than any of the other sites. The last column in Tab. 27 reports the fraction of time the respecitve sites are bound when 100 Rep are included and dimerization activated.

Binding of repressor to the right operator O_R: Figure 28 shows the saturation of sites O_{R1}, O_{R2} and O_{R3} as they arise in our simulations of the λ switch when both repressor dimerization and cooperative repressor binding between O_{R1} and O_{R2} are enabled. Each of the curves summarize a series of

experiments for varying Rep levels. Before discussing the full system, we will investigate the underlying components and mechanisms one by one.

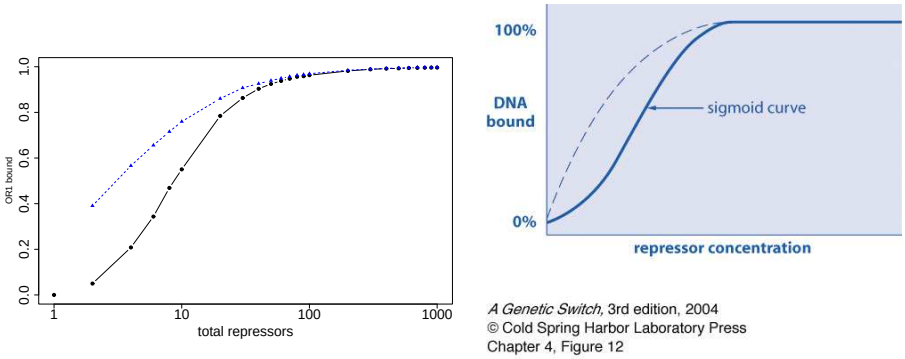


Fig. 29. Occupancy of operator region O_{R1} as a function of repressor concentration and dimerization. Our results (left), benchmark from Ptashne's book [35] (right). Dashed lines: all repressors are found as dimers regardless of concentration. Solid lines: dimerization of repressors is included, hence the concentration-dependent equilibrium affects the binding curve.

Dimerization sharpens response at O_{R1}: Figure 29 (left) illustrates the saturation of O_{R1} as a function of repressor level. Each of the two curves summarizes a series of experiments. For the solid line dimerization is enabled, i.e. only part of the total repressors are present as dimers and thus able to bind the operator. The dashed line assumes that dimers are stable at all concentrations, meaning that 100% of total repressors are found as dimers regardless of the concentration. The x axis indicates the number of total repressors on a logarithmic scale, while the y axis gives the relative occupancy of O_{R1}. Each data point represents the relative occupancy of O_{R1} for an experiment simulating the full dynamics of docking to DNA with or without dimerization over 5000 seconds.

Over this time scale, we can compare our results based on a stochastic discrete event approach against other's from deterministic continuous models, which compute only averages: one sees both qualitative and quantitative agreement with results from [35] reproduced in Fig. 29 (right). Dimerization has the effect to change the shape of the binding curve, namely to give a sharper response in terms of site occupancy as the amount of repressor increases.

Superimposing dimerization and cooperative binding at O_{R2}: Figure 30 summarizes how the second operator site fills with Rep for three scenarios. The dashed curve illustrates binding to O_{R2} in presence of O_{R1} and dimerization. We contrast this with binding to the isolated O_{R2} with and without dimerization. Note that the effect of *dimerization* is far less pronounced at an isolated O_{R2} than it was O_{R1}, where dimerization lead to a sharp increase of

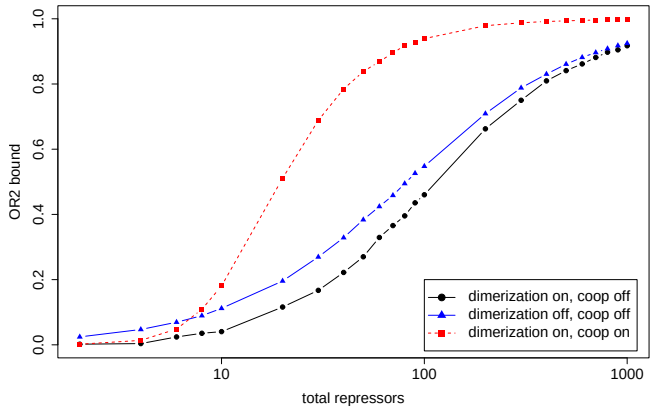


Fig. 30. Occupancy of site O_{R2}, as a function of repressor level, dimerization and cooperativity. We consider *isolated* O_{R2} for the curves 'with'/'without' dimerization, and a system comprising both O_{R1} and O_{R2} for the curve 'O_{R2}, coop and dimerization'. Because at very low concentrations, binding occurs mainly at O_{R1}, the cooperative advantage only becomes visible at a certain level.

sensitivity in the lower concentration range. This can be explained because the isolated weaker O_{R2} only fills notably at higher Rep concentrations, when the equilibrium is heavily biased toward dimers. However, now the *combined effect of cooperativity and dimerization* becomes prominent. Recall that binding at O_{R2} is cooperatively strengthened as O_{R1} is placed next to it. As can be seen from the dashed curve in Fig. 30 the predominant cause of O_{R2}'s saturation at lysogenic repressor concentration levels is cooperative binding with O_{R1}. This cooperativity propagates O_{R1}'s stronger sensitivity to O_{R2}.

Negative auto-regulation of Rep at O_{R3}: Our results on Rep binding to the third operator O_{R3} are included in Fig. 28. The binding curve is again based on the observation of the isolated operator, under variation of repressor level while disregarding interfering traffic between RNAP and P_{RM}. The most striking effect when comparing the binding curve with the other operators' is that the site fills to a significantly lower degree. It reaches around 30% when 200 repressors are included in the simulation. Even with an amount of 1000 repressors the isolated O_{R3} remains unsaturated. As we will report later the saturation further decreases to 4% as RNAP docking to P_{RM} interferes.

These results agree with recent experimental findings. Dodd and co-workers have demonstrated that an *additional layer of cooperativity* is needed for effective repression of P_{RM} [9] at lysogenic repressor concentrations. Revet and co-workers first observed a long-range DNA loop between the right operator and another distal region in λ's genome [40]. As was subsequently understood, this loop is stabilized by an assembly of eight repressor proteins, in which the two repressor

dimers cooperatively bound to O_{R1} and O_{R2} participate. They cooperatively interact with another repressor tetramer bound to λ's *left operator* region O_L - while looping the DNA between the two regions. The large assembly further stabilizes all participants. More importantly, it juxtaposes O_{R3} with a third site at the left operator, O_{L3}. This allows for cooperative binding of repressor at O_{R3} and O_{L3}.

This additional level of cooperativity is a recent finding and not, as yet, fully characterized. However its importance is clearly seen [46]. It allows to repress P_{RM} and maintain a low level of Rep, that is not yet ensured by binding to O_{R3} alone. Thus in a bacterium hosting phage λ, the lysogenic repressor concentration never surpasses a range allowing to return to the level in which O_{R1} and O_{R2} can be vacated. This is the key to induction back from lysogeny to the state of lytic growth [11].

coop	pos. control	repressors	transcription initiations from P_{RM}	RNAP at P_{RM} (sec)	P_{RM} repressed (sec)	P_{RM} vacant (sec)	Rep at O_{R2} (sec)
on	on	100	58	873	46	280	1111
			76	870	46	279	1120
			71	873	46	279	1120
			58	889	44	266	1110
			74	903	43	253	1111
		50	67	901	19	281	990
			69	921	14	265	985
			70	899	18	283	972
		25	46	917	5	278	728
			52	929	5	266	730
			51	922	5	273	727
		10	19	927	< 1	272	305
			26	932	< 1	267	323
			28	922	< 1	277	300
off	on	100	38	886	41	273	570
			35	867	45	285	587
			31	890	43	267	575
on	off	100	5	899	41	260	1108
			6	915	39	246	1118
			4	898	44	258	1101

Fig. 31. Simulations over 1200 seconds: P_{RM} activity (absolute number of transcription initiations). Values are in units of simulated seconds for the following columns: P_{RM} repression, P_{RM} vacancy, and occupancy of O_{R2} by Rep. First block: results for varied repressor levels when both cooperative repressor binding and positive control are enabled. Second and third block: simulation results under elimination of either of the two mechanisms, for a level of 100 repressors.

6.3 RNAP binding and transcription initiation

We further increase the scope of the model, by adding P_{RM} and RNAP to the three operator sites and repressors simulated so far. We study this system over 20 minutes simulated time, comparable with the life span of an individual bacterium. This allows to observe the occupancy patterns of both P_{RM} and the operators, as well as the initiation of transcription for the cI gene. We first consider the impact of varying repressor levels in a model including the essential cooperative features - dimerization, cooperative repressor binding and positive control of transcription initiation. Next we perturb our model and study the consequences.

Table 31 summarizes a series of simulations. For each set-up we indicate the number of repressors included, and whether cooperative binding and positive control are enabled. Each line of the table summarizes one simulation run, for which we report the following quantities:

- absolute number of transcription initiations from P_{RM} observed. This should be related to the theoretical upper bound of 103, estimated form P_{RM}'s maximal rate of 0.086 initiations per second and the simulated time of 1200 seconds.
- absolute time that RNAP is bound to P_{RM} (in sec),
- absolute time P_{RM} is repressed as a consequence of Rep binding to O_{R3} (in sec),
- vacancy of P_{RM} (in sec),
- occupancy of O_{R2} by repressor (in sec).

To begin with, we mimic the system's behavior under *lysogenic repressor concentrations* with 100 repressors. The first block in Tab. 31 summarizes five executions of our model. For all runs P_{RM} is bound by RNAP in approximately 70 % of the time, P_{RM} is inhibited via competing Rep binding to O_{R3} for 4 % of the time, and otherwise vacant. The second operator site O_{R2} is bound by Rep around 92 % of the time for all five runs. The numbers of transcription initiations range between 58 and 76. Note that the variability of this figure is significantly higher than that of any other considered.

The actual number of transcript initiations from P_{RM} in a single lysogenic bacterium is difficult to determine experimentally. In the past, a precise level was deemed necessary for the maintenance of lysogeny [20]. This was rectified by recent experiments showing that repressor levels varies widely from cell to cell [4]. This phenomenon is known as *transcriptional noise*. Our result are in agreement with an estimated average number of transcripts per cell cycle of 70.

Reactions to partial and near-total Rep depletion: Our next step is to preserve the system's essential characteristics - dimerization, positive control and cooperative binding - but to thin out the repressor pool. In the remainder of Tab. 31's first block we report the outcomes of each three simulations with 50, 25 and 10 total repressors.

The primary effect is that saturation of O_{R2} drops nonlinearly. This has important secondary effects on transcription initiation from P_{RM}. A first reduction

to 50 repressors de-represses P_{RM} and seems to favor initiation, at least in these runs. As a reaction to further depletion transcription visibly reduces, while the actual P_{RM} saturation by RNAP increases: in the presence of 10 repressors, initiations drop to around a third of those seen with 50 repressors. This should be related to the increasing vacancy of O_{R2}. It gives a first impression of how the system of positive auto-regulation breaks down.

Examining this question in detail seems promising for two reasons. First, the λ switch is known to be extremely robust. It needs to cope with transient fluctuations of Rep level. Nevertheless, *induction* relies on the system's ability to escape from the lysogenic state when repressor falls below a critical threshold. Recall that as both O_{R1} and O_{R2} are vacated, P_{RM}'s antagonist P_R becomes likely to take over. A detailed investigation remains beyond the current paper's scope.

The impact of cooperative repressor binding on P_{RM} activity is another point of interest. After we have observed its immediate impact at O_{R2}, we move on to a larger perspective. We perturb our π-calculus model by lowering the cooperative dissociation rate of Rep at O_{R2} to the basal one. This lowers O_{R2}'s saturation to half the previous amount, see Tab. 31 (second block, last column). And it has consequences for RNAP at P_{RM}. The binding itself is not lowered - our simulations even indicate a slightly higher promoter saturation. Nevertheless, the number of transcription initiations drops to half that of the wild type.

Impact of positive control: We last eliminated the positive control of transcription initiation in our model. This is reached by lowering the parameter for promoted transcription the basal one, i.e. by manipulating the stochastic parameters of the channels illustrated in Fig. 22. Our resulting in silico experiments are summarized in the last block in Tab. 31. The number of initiations in presence of 100 total repressors dramatically decreases from an average of 67 in the initial system to an average of 5. All the while, immediate RNAP bindings as well as all other features are not affected, when compared to the original setting. This underlines the importance of positive control.

This simulation scenario was motivated by wet lab experiments with modified λ repressors. These mutants bind cooperatively but fail to stimulate transcription. As [17] reported the λ switch was no longer functional. Our simulation outcomes seem in rough agreement with this, even though we can not directly compare the results. Most recently Michalowski and Little [27] suggested that positive auto-regulation may be a dispensable feature altogether. They experimentally observed that the λ switch remains functional if positive control is eliminated, but at the same time P_{RM}'s intrinsic initiation rate k_f increased.

Conclusion and Future Work

We have presented a detailed model for the mechanism of transcription initiation control at the λ switch in the stochastic π-calculus. We have distilled the stochastic parameters from the literature, implemented the model in the BioSpi

	our prediction	reference value	
P_{RM} activity, (full system)	67 initiations in 1200 sec	estimate: 70 per cell cycle	[4]
P_{RM} activity, (pos. control off)	$>$ 90% reduced	observation: positive control can be eliminated, but this needs to be compensated by up-regulation of basal rate in order to maintain system functional	[27]
P_R repressed ($>$ 45 Rep)	$>$ 98.1 %	98.5 %	[26]
O_{R1}-O_{R2} cooperativity	(reproduced)	necessary to repress P_R efficiently	[35]
P_{RM} repression at lysogenic Rep levels, considering O_R only	4%	repression is ineffective; transcription initiations lowered 5-20% as a consequence of O_{R3} binding	[26, 9]

Fig. 32. Overview of our predictions and results from other studies.

system, and obtained confirming simulation results. Figure 32 summarizes our simulations.

In follow up work we have already extended our approach to modeling the dynamics of transcript elongation itself [23]. Work on translation is under way, again the most subtle aspects include appropriate parameter choices [9]. This will permit us to close the complex feedback loops at the λ switch.

We aim to obtain a simulation of *induction*. Recent experiments suggest a new ambiguity for Cro's role. The common assumption that its presence at intermediate repressor levels rapidly leads to induction has been falsified. DNA looping between the right and left operator region seems to render the switch insensitive to the presence of Cro [45]. Therefore, we also plan to refine our approach in order to account for long-distance cooperativity in repressor binding. The techniques developed in this paper should be helpful.

Finally, we plan to study more effects of parameter variation systematically.

Acknowledgments Our view of the λ switch has been sharpened in discussions with Erik Aurell, Isabella Moll, and Mark Ptashne. We thank Keith Shearwin for feed-back on the manuscript, and Denys Duchier, Cédric Lhoussaine, and Bernard Vandenbunder for discussions. This work was partly carried out during a research visit of CK to the *Linnaeus Centre for Bioinformatics* in Uppsala (Sweden), enabled through a European Commission grant HPRI-CT-2001-00153. Last but not least, she gratefully acknowledges the LIFL for its hospitality.

[9] The parameter set for cI translation in [2] and adopted by several others should be revised in order to reflect newer findings [30, 42].

References

1. Gary K. Ackers, Alexander D. Johnson, and Madeline A. Shea. Quantitative model for gene regulation by λ phage repressor. *Proceedings of the National Academy of Sciences USA*, 79(4):1129–1133, February 1982.
2. Adam Arkin, John Ross, and Harley H. McAdams. Stochastic kinetic analysis of developmental pathway bifurcation in phage λ-infected Escherichia coli cells. *Genetics*, 149:1633–1648, 1998.
3. Erik Aurell, Stanley Brown, Johan Johanson, and Kim Sneppen. Stability puzzles in phage λ. *Physical Review E*, 65:051914, 2002.
4. Kristoffer Baek, Sine Svenningsen, Harvey Eisen, Kim Sneppen, and Stanley Brown. Single-cell analysis of λ immunity regulation. *Journal of Molecular Biology*, 334(3):363–372, 2003.
5. Audun Bakk. Transcriptional activation mechanisms of the P_{RM} promoter of λ phage. *Biophysical Chemistry*, 114(2–3):229–234, 2005.
6. Otto G. Berg, Robert B. Winter, and Peter H. von Hippel. Diffusion-driven mechanisms of protein translocation on nucleic acids: 1 - models and theory. *Biochemistry*, 20:6929–6948, 1981.
7. J. M. Bower and H. Bolouri, editors. *Computational Modeling of Genetic and Biochemical Networks*. MIT Press, 2001.
8. Ralf Bundschuh, F. Hayot, and C. Jayaprakash. The role of dimerization in noise reduction of simple genetic networks. *Journal of Theoretical Biology*, 220:261–269, 2003.
9. Ian B. Dodd, A.J. Perkins, D. Tsemitsidis, and J.B. Egan. Octamerization of CI repressor is needed for effective repression of P_{RM} and efficient switching from lysogeny. *Genes & Development*, 15:3013–3022, 2001.
10. Ian B Dodd, Keith E Shearwin, and J Barry Egan. Revisited gene regulation in bacteriophage λ. *Current Opinion in Genetics & Development*, 15(2):145–152, 2005.
11. Ian B. Dodd, Keith E. Shearwin, Alison J. Perkins, Tom Burr, Ann Hochschild, and J. Barry Egan. Cooperativity in long-range gene regulation by the lambda CI repressor. *Genes Dev.*, 18(3):344–354, 2004.
12. Johan Elf and Mans Ehrenberg. What makes ribosome-mediated trascriptional attenuation sensitive to amino acid limitation? *PLoS Computational Biology*, 1(1):14–23, 2005.
13. Harley Lodish et al. *Molecular Cell Biology*. Freeman, 2003.
14. Daniel T. Gillespie. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *Journal of Computational Physics*, 22:403–434, 1976.
15. Jeff Hasty, David McMillen, Farren Isaacs, and James J. Collins. Computational studies of gene regulatory networks: In numero molecular biology. *Nature Reviews Genetics*, 2:268–279, 2001.
16. DK Hawley, AD Johnson, and WR McClure. Functional and physical characterization of transcription initiation complexes in the bacteriophage lambda O_R region. *J. Biol. Chem.*, 260(14):8618–8626, 1985.
17. D.K. Hawley and W.R. McClure. The effect of a lambda repressor mutation on the activation of transcription initiation from the lambda P_{RM} promoter. *Cell*, 32:327–333, 1983.
18. Jane Hillston. *A Compositional Approach to Performance Modelling*. PhD thesis, University of Edinburgh, 1995. Distinguished Dissertations Series. Cambridge University Press, 1996.

19. T. Ideker, T. Galitski, and L. Hood. A new approach to decoding life: systems biology. *Annual Review of Genomics and Human Genetics*, 2(343), 2001.
20. A D Johnson, A R Poteete, G Lauer, R T Sauer, G K Ackers, and M Ptashne. λ repressor and Cro – components of an efficient molecular switch. *Nature*, 294(5838):217–223, 1981.
21. Andrzej M. Kierzek, Jolanta Zaim, and P. Zielenkiewicz. The effect of transcription and translation initiation frequencies on the stochastic fluctuations in prokaryotic gene expression. *Journal of Biological Chemistry*, 276:8165–8172, 2001.
22. Kenneth S. Koblan and Gary K. Ackers. Site-specific enthalpic regulation of DNA transcription at bacteriophage λ O_R. *Biochemistry*, 31:57–65, 1992.
23. Céline Kuttler. Bacterial transcription in the pi calculus. In *3rd International Workshop on Computational Methods in Systems Biology*, 2005.
24. Mei Li, W.R. McClure, and M. M. Susskind. Changing the mechanism of transcriptional activation by phage λ repressor. *Proceedings of the National Academy of Sciences USA*, 94:3691–3696, 1997.
25. William R. McClure. Mechanism and control of transcription initiation in prokaryotes. *Annual Review Biochemistry*, 54:171–204, 1985.
26. B. Meyer and M. Ptashne. Gene regulation at the right operator O_R of bacteriophage lamba. I. O_{R3} and autegeneous negative control by repressor. *J. Mol. Biol.*, pages 19–205, 1980.
27. Christine B. Michalowski and John W. Little. Postitive autoregulation of cI is a dispensable feature of the phage λ gene regulatory circuitry. *J. Bacteriol.*, 187(18):6430–6442, 2005.
28. Robin Milner. *Communicating and Mobile Systems: the π-calculus.* Cambridge University Press, 1999.
29. Robin Milner, Joachim Parrow, and David Walker. A calculus of mobile processes (I and II). *Information and Computation*, 100:1–77, 1992.
30. Isabella Moll, Go Hirokawa, Michael C. Kiel, Akira Kaji, and Udo Bläsi. Translation initiation with 70S ribosomes: an alternative pathway for leaderless mRNAs. *Nucl. Acids Res.*, 32(11):3354–3363, 2004.
31. Amos B. Oppenheim, Oren Kobiler, Joel Stavans, Donald L. Court, and Sankar Adhya. Switches in bacteriophage lambda development. *Annual Reviews Genetics*, 39:409–429, 2005.
32. Andrew Phillips and Luca Cardelli. A correct abstract machine for the stochastic pi-calculus. *Transactions on Computational Systems Biology*, 2005. to appear.
33. Corrado Priami. Stochastic π-calculus. *Computer Journal*, 6:578–589, 1995.
34. Corrado Priami, Aviv Regev, Ehud Shapiro, and William Silverman. Application of a stochastic name-passing calculus to representation and simulation of molecular processes. *Information Processing Letters*, 80:25–31, 2001.
35. Mark Ptashne. *A Genetic Switch: Phage Lambda Revisited.* Cold Spring Harbor Laboratory Press, 3rd edition, 2004.
36. Mark Ptashne and Alexander Gann. *Genes and Signals.* Cold Spring Harbor Laboratory Press, 2002.
37. Aviv Regev. *Computational Systems Biology: A Calculus for Biomolecular Knowledge.* Tel Aviv University, 2002. PhD thesis.
38. Aviv Regev and Ehud Shapiro. Cells as computation. *Nature*, 419:343, 2002.
39. Aviv Regev and Ehud Shapiro. The π-calculus as an abstraction for biomolecular systems. In Gabriel Ciobanu and Grzegorz Rozenberg, editors, *Modelling in Molecular Biology.* Springer, 2004.

40. Bernard Révet, Brigitte von Wilcken-Bergmann, Heike Bessert, Andrew Barker, and Benno Müller-Hill. Four dimers of lambda repressor bound to two suitably spaced pairs of lambda operators form octamers and DNA loops over large distances. *Current Biology*, 9(3):151–154, 1999.
41. Madeline Shea and Gary K. Ackers. The O_R control system of bacteriophage lambda: A physical-chemical model for gene regulation. *Molecular Biology*, 181:211–230, 1985.
42. C. S. Shean and M. E. Gottesman. Translation of the prophage λ cI transcript. *Cell*, 70(3):513–522, 1992.
43. Michael Slutsky and Leonid A. Mirny. Kinetics of protein-DNA interaction: Facilitated target location in sequence-dependent potential. *Biophys. J.*, 87(6):4021–4035, 2004.
44. Kim Sneppen and Giovanni Zocchi. *Physics in Molecular Biology*. Cambridge University Press, 2005.
45. Sine L. Svenningsen, Nina Costantino, Donald L. Court, and Sankar Adhya. On the role of Cro in λ prophage induction. *Proceedings of the National Academy of Sciences USA*, 102(12):4465–4469, 2005.
46. Jose MG Vilar and Leonor Saiz. DNA looping in gene regulation: from the assembly of macromolecular complexes to the control of transcriptional noise. *Current Opinion in Genetics & Development*, 15:1–9, 2005.
47. Rolf Wagner. *Transcription Regulation in Prokaryotes*. Oxford University Press, 2000.