UNIVERSITÀ DEGLI STUDI DI PADOVA

Dipartimento di Fisica e Astronomia "Galileo Galilei"

Corso di Laurea Magistrale in Fisica

Tesi di Laurea

How can cells count?

Robustness in self-assembly pathways

Relatore

Prof. Enzo Orlandini

Correlatore

Prof. Erwin Frey

Laureando

Michele Guerra

Anno Accademico 2018/2019

Acknowledgements

I would like to thank my advisor, Prof. Erwin Frey, and his whole group at Ludwig Maximilian University of Munich. They provided me, for my whole ERASMUS stay, with a lot of means and expertise that I probably wouldn't have had anywhere else. In particular I address a special thanks to Isabella Graf, who has been so generous to dedicate much of her time to me and guide me throughout this project.

Contents

Introduction

Cells are often mistakenly seen as simple bubbles of proteins, able to grow and divide. Instead, in recent years (see [7] and [2]), it has been highlighted that their content, such as organelles, is highly organized. If we consider more carefully the plenty of varieties of cells, we certainly realise that a lot of them are characterised by very peculiar structures, patterns or shapes. For example neurons stand out thanks to the presence of the dendritic tree and the axon, or the distinctive shape of cone and rod that characterises cells in the eye, or again those tails, called flagella, that many bacteria have for moving around. Of course, these elements are not merely aesthetic features, but are crucial for the functionality of the cell itself.

All the information available to the cell, required to build an organelle, is stored in the genome, but that's not enough. In fact, for two reasons, the genome can't be thought of as a mold that accurately encodes the plan for building a structure. The first reason is that living systems are dynamic, so they can't be portrayed by a fixed drawing, they rather need a flexible description of their form. The second reason is that genes can't store "vectorial" information, such as position or orientation, they just control concentrations of proteins. The general question is then "how can cells dictate **cell geometry**?", that is how can "scalar" information be translated into "vectorial" one?

In this thesis we will not tackle the problem in all its generality, because it is too wide, but we will focus on one particular aspect: how can a cell control the number of structures it produces? But first, does it? If there is no control at all one would expect a random process with a broad distribution in a cell population and, indeed, some organelles seems to vary randomly in a population, but others don't. In [6] the authors show that the chloroplasts in algae were not equally partitioned in daughter cells during cell division, but still, after some time, we find the same number of chloroplasts. So, there must be a control mechanism that compensates this initial imbalance. Another example is given by flagella in bacteria. Flagella appear in nature in different numbers and patterns. In fact, we generally speak of flagellation pattern, and it has been shown that the number of flagella is calibrated according to the environment conditions (e.g. viscosity) in order to ensure the best possible motility to the cell (reviewed in [1] and [2]).

As Marshall points out in [7], the problem of controlling the number of structures produced, which is just a narrower aspect of the original question as already explained, is again more complex than one could think. In fact, for many organelles it is not possible to separately consider their number and their size. Those two aspects are strictly related: if the cell produces a limited amount of "bricks" for the structure, either fewer but larger structures are produced, or more but smaller. To avoid this problem and focus on the number control only, we consider those organelles whose size is somehow fixed, for example if they have a ring shape completely determined by protein interactions. This level of simplification is still not enough, because it could happen that, starting from a given amount of building blocks, the cell ends up producing a bunch of incomplete structures instead of a complete one (see [4]).

This work focuses on a particular type of structure, the flagellum. The flagellar complex

is very similar in all species of bacteria: it can be schematically divided into a basal body, which lies in the cytoplasm, and the rod, hook and filament that protend into the extracellular space. Inside the basal body you find the motor machinery, in particular the **flagellar motor assembly**, which is responsible for the spinning of the whole tail and so allows the bacterium to move. Since the flagellar motor assembly is a ring, we can use the above argument and directly link the amount of proteins needed for its construction with the number of complete structures built. Moreover we will assume that the assembling process leads to complete structures only. In particular, this work studies the case of *Shewanella putrefaciens*, which is characterised by just one flagellum, so our question could be rephrased as "how can the cell count up to one?". It could seem a strange and too strict question to ask, but if the process is efficient (in the sense that the cell is able to produce only complete structures), then counting up to one is enough to count up to any number. All the biochemical information used was gently provided by Dr. Devid Mrusek.

The aim of the thesis is to analyse the behaviour of the system with respect to the production of the "bricks" of the flagellar motor assembly, both deterministically and stochastically. In a deterministic perspective concentration levels of proteins are studied using the law of mass action, while from a stochastic point of view it is assumed that proteins are too few to be described by concentrations, so they are modelled by a stochastic variable. We suppose that the system exhibits a reliable **counting mechanism** if it acts robustly. By **robustness** we mean essentially two things:

- we say that a system is deterministically robust if the final amount of proteins produced doesn't vary too much when the initial conditions vary;
- moreover it is stochastically robust if the number of "bricks" has small statistical fluctuations.

If the system stays robust (almost) regardless of all the parameter values, then we can safely assume that it is able to build the needed amount of assemblies.

We will proceed as follows. In Chapter 1 all the mathematical instruments are introduced, starting from the definition of a stochastic variable to the construction of the master equation that describes its dynamics. It will also present Gillespie's algorithm which will be heavily used in Chapter 2, where we first describe in detail all the reactions involved in the production of the flagellar motor assembly. Then we simulate the system trying to understand the role of each reaction from a robustness perspective. In the same chapter, we will also try some analytical studies. Last, in the final chapter, we will discuss the results and what could still be done.

Chapter 1

Methods

In this chapter, we briefly introduce all the notions and methods about stochastic processes we are going to use in the following sections. After summarising the basic concepts of probability theory, we proceed by exposing the theory of master equations and how to explore their predictions by means of an exact solution, if possible, or exact algorithms, like the one developed by Gillespie. This chapter is mainly based on the books of Gardiner [3], Ross [8] and Tijms [5].

1.1 Basics and notations

As commonly done in probability theory literature, we denote with Ω the **sample space**. The set of all possible events is a σ-algebra $\mathscr{F}$ on the sample space Ω. By introducing a **probability measure**, $\mathbb{P} : \mathscr{F} \to [\,0,\,1\,]$, we define the **probability space** $(\Omega, \mathscr{F}, \mathbb{P})$. A **random variable** $X : \Omega \to \mathbb{R}$ is then a $\mathscr{F}$-measurable function, that is

$$\{\,\omega \in \Omega : X(\omega) \leq x\,\} \in \mathscr{F}, \quad \forall x \in \mathbb{R}.$$

In this work we will always deal with discrete and positive random variables, which means that X will be an $\mathbb{N}$-valued function.

All the useful information about a random variable is contained in its **cumulative distribution function**, $F : \mathbb{R} \to [\,0,\,1\,]$, defined by

$$F(x) := \mathbb{P}(X^{-1}(-\infty, x)),$$

or by the **probability density function** $f(x)$, defined via $dF(x) = f(x)\,dx$. Given two random variables, X and Y, we can define their **joint cumulative distribution function** as

$$F(x, y) := \mathbb{P}(X^{-1}(-\infty, x) \cup Y^{-1}(-\infty, y)),$$

and their **joint probability density function** $f(x, y)$, similarly as before,

$$dF(x, y) = f(x, y)\,dxdy.$$

A **stochastic process** $X(t)$ is a family of random variables parametrised by a continuous or discrete parameter t, $\{\,X(t) : t \in T\,\}$, so all the quantities related to it will depend on t too. For example, a probability density function will be written as $f(x, t)$, a joint probability density as $f(x, t; x', t')$, and so on.

Let $X(t)$ (with $t \in \mathbb{R}$) be a stochastic process, and consider a joint probability density like $f(x,t;x',t')$, then the **conditional joint probability density** is defined as

$$f(x,t|x',t') := f(x,t;x',t')/f(x',t').$$

In the following sections, our physical system will be modelled with a stochastic process, where the parameter t will simply be time. In order to describe the time evolution of the system then, an equation for the dynamics will be needed. That will be the **master equation**.

Given a discrete random variable N we can define its **expected value**, or mean value, by

$$\mathrm{E}[N] := \sum_{n\in\mathbb{N}} nf(n),$$

where $f(n)$ is the probability density function, which in the discrete case is just $f(n) = \mathbb{P}(N = n)$. Sometimes it will be easier to use the notation with brackets $\langle n\rangle := \mathrm{E}[N]$. In a similar way we define the ℓ-th **moment** μ_ℓ of N

$$\mu_\ell = \mathrm{E}[N^\ell] = \sum_{n\in\mathbb{N}} n^\ell f(n),$$

so the expected value is the first moment. The idea behind this definition is that moments give information about the shape of the density function. A measure on how $f(n)$ is spread around the mean value is given by the **variance**

$$\mathrm{Var}[N] := \mathrm{E}[(N - \mathrm{E}[N])^2],$$

which is the second central moment of the distribution, where "central" means that it is computed around the mean value. In case we are studying two random variables, N_1 and N_2, their correlation is measured by the **covariance**

$$\mathrm{Cov}[N_1, N_2] = \mathrm{E}[(N_1 - \mathrm{E}[N_1])(N_2 - \mathrm{E}[N_2])],$$

computed using their joint density function.

1.2 Master equation

From now on we will be considering only random variables that take values in $\mathbb{N}$ and which are parametrised by a continuous parameter $t \in \mathbb{R}$, which we will just call time. In order to be consistent, a joint probability density function of a stochastic process $\{N(t), t \geq 0\}$ must satisfy

$$f(n_3,t_3;n_1,t_1) = \sum_{n_2\in\mathbb{N}} f(n_3,t_3;n_2,t_2;n_1,t_1).$$

It will be called a **Markov process** if the Markov assumption holds, namely that the conditional joint probability density depends only on the most recent condition. So, if we consider an ordered sequence of times $t_i > t_{i-1}$, the Markov assumption reads like

$$f(n_{m+\ell},t_{m+\ell};\ldots;n_{m+1},t_{m+1}|n_m,t_m;\ldots;n_1,t_1) = f(n_{m+\ell},t_{m+\ell};\ldots|n_m,t_m).$$

By combining these two properties, the former valid for a generic stochastic process and the last for a Markov process only, we get

$$f(n_3,t_3;n_1,t_1) = f(n_3,t_3|n_1,t_1)f(n_1,t_1)$$

and

$$\begin{aligned} f(n_3,t_3;n_1,t_1) &= \sum_{n_2\in\mathbb{N}} f(n_3,t_3;n_2,t_2;n_1,t_1) = \\ &= \sum_{n_2\in\mathbb{N}} f(n_3,t_3|n_2,t_2;n_1,t_1) f(n_2,t_2;n_1,t_1) = \\ &= \sum_{n_2\in\mathbb{N}} f(n_3,t_3|n_2,t_2) f(n_2,t_2|n_1,t_1) f(n_1,t_1), \end{aligned}$$

which lead to the **Chapman-Kolmogorov equation**

$$f(n_3,t_3|n_1,t_1) = \sum_{n_2\in\mathbb{N}} f(n_3,t_3|n_2,t_2) f(n_2,t_2|n_1,t_1). \tag{1.1}$$

We also define the **jump rate** as the following limit (if it exists)

$$w_{nm}(t) := \lim_{\Delta t\to 0} \frac{f(n,t+\Delta t|m,t)}{\Delta t}, \tag{1.2}$$

which represents the probability per unit of time to transit from m to n at time t.

If the previous condition holds, we can translate the Chapman-Kolmogorov equation into a differential equation involving the jump rates. Consider the time derivative of the following conditional probability density

$$\begin{aligned} \partial_t f(n,t|n_0,t_0) &= \lim_{\Delta t\to 0} \frac{1}{\Delta t} \left[f(n,t+\Delta t|n_0,t_0) - f(n,t|n_0,t_0) \right] \overset{(1.1)}{=} \\ &= \lim_{\Delta t\to 0} \frac{1}{\Delta t} \left[\sum_m f(n,t+\Delta t|m,t) f(m,t|n_0,t_0) - f(n,t|n_0,t_0) \right] = \\ &= \lim_{\Delta t\to 0} \frac{1}{\Delta t} \sum_m \big[\, f(n,t+\Delta t|m,t) f(m,t|n_0,t_0) + \\ &\qquad\qquad - f(m,t+\Delta t|n,t) f(n,t|n_0,t_0) \, \big] \overset{(1.2)}{=} \\ &= \sum_m \left[w_{nm}(t) f(m,t|n_0,t_0) - w_{mn}(t) f(n,t|n_0,t_0) \right], \end{aligned}$$

where $t > t_0$ and in the third line we used the fact that a probability density function is normalised, i.e. $\sum_m f(m,t+\Delta t|n,t) = 1$. By multiplying both sides by $f(n_0,t_0)$, which doesn't depend on t, and summing over n_0, we get the **master equation**

$$\partial_t f(n,t) = \sum_m \left[w_{nm}(t) f(m,t) - w_{mn}(t) f(n,t) \right], \tag{1.3}$$

which is a complicate differential equation for the probability density function $f(n,t)$.

It is straightforward to extend the previous derivation to the case where the stochastic process $X(t)$ takes values in a vectorial space, specifically $X(t) \in \mathbb{N}^N$ with a positive integer N. If we denote vectors with a bold character, like $\boldsymbol{n} \in \mathbb{N}^N$, then the master equation becomes

$$\partial_t f(\boldsymbol{n},t) = \sum_{\boldsymbol{m}\in\mathbb{N}^N} \left[w_{\boldsymbol{nm}}(t) f(\boldsymbol{m},t) - w_{\boldsymbol{mn}}(t) f(\boldsymbol{n},t) \right].$$

1.3 Gillespie's algorithm

1.3.1 Poisson process

Among the many possible definitions of a Poisson process, the one that gives most insight and will be most useful later is the following. First, we need to define what a counting process is.

Definition 1. *A stochastic process* $\{N(t), t \geq 0\}$ *is a* ***counting process*** *if it represents the total number of events that occurred up to time* t*, that is if it satisfies*

(i) $N(t) \geq 0$,

(ii) $N(t) \in \mathbb{N}$ *for all* $t \geq 0$,

(iii) *if* $s < t$*, then* $N(s) \leq N(t)$,

(iv) *for* $s < t$*,* $N(t) - N(s)$ *is the number of events that occured in the interval* $(s, t]$.

Definition 2. *A counting process* $\{N(t), t \geq 0\}$ *is a* ***Poisson process*** *with rate* $\lambda > 0$ *if*

(i) $N(0) = 0$,

(ii) *the numbers of events that occur in disjoint time intervals are independent (independent increments), and the number of events that occur in any interval of time depends only on the length of the time interval (stationary increments),*

(iii) $\mathbb{P}(N(h) = 1) = \lambda h + o(h)$,

(iv) $\mathbb{P}(N(h) \geq 2) = o(h)$.

We can think of a Poisson process as a way to count events that happen with no memory (so it is a Markov process) and such that an event is unlikely to occur twice in a small interval of time.

Consider now the probability that n events occurred up to time t, $p_n(t) = \mathbb{P}(N(t) = n)$. Then we can derive its explicit form by means of Definition 2 only. Thanks to the assumptions we made, we are able to write a differential equation for $p_0(t)$

$$\begin{aligned} p_0(t+h) &= \mathbb{P}(N(t+h) = 0) = \\ &= \mathbb{P}(N(t) = 0, N(t+h) - N(t) = 0) = \\ &= \mathbb{P}(N(t) = 0)\mathbb{P}(N(t+h) - N(t) = 0) = \\ &= p_0(t)(1 - \lambda h + o(h)), \end{aligned}$$

where we used respectively the assumptions (ii) and (iii). Taking the limit for h going to zero gives us the differential equation

$$p_0'(t) = -\lambda p_0(t),$$

which, using as initial condition $p_0(0) = \mathbb{P}(N(0) = 0) = 1$ (according to assumption (i)), yields

$$p_0(t) = e^{-\lambda t}.$$

We can apply the same reasoning to find the explicit expression for $p_n(t)$, in particular

$$\begin{aligned} p_n(t+h) &= \mathbb{P}(N(t+h)=n) = \\ &= \mathbb{P}(N(t)=n, N(t+h)-N(t)=0)+ \\ &\quad + \mathbb{P}(N(t)=n-1, N(t+h)-N(t)=1)+ \\ &\quad + \mathbb{P}(N(t+h)=n, N(t+h)-N(t)\geq 2) = \\ &= (1-\lambda h)p_n(t) + \lambda h\, p_{n-1}(t) + o(h), \end{aligned}$$

where, in addition to what we did before, we used assumption (iv) and that, according to (iii), $p_0(h) = 1 - \lambda h + o(h)$. Taking the limit for h going to zero, we find the equation

$$p_n'(t) = -\lambda\, p_n(t) + \lambda\, p_{n-1}(t). \tag{1.4}$$

By induction, it's possible to prove that the solution of this equation is

$$p_n(t) = e^{-\lambda t}\frac{(\lambda t)^n}{n!}, \tag{1.5}$$

which is the probability density function for a Poisson process.

Notice that equation (1.4) has exactly the form of the master equation (1.3) with constant non zero jump rates $w_{nm}(t) = \lambda$ for $m = n-1$. This will be useful later on.

1.3.2 Properties of Poisson process

We can use (1.5) to compute the expected value and variance of the number of events as function of time

$$\mathrm{E}[N](t) = \sum_{n=0}^{+\infty} n p_n(t) = \lambda t$$

$$\mathrm{Var}[N](t) = \mathrm{E}[(n - \mathrm{E}[N])^2] = \lambda t,$$

which explains why λ is called "rate" of the process.

Consider a Poisson process $\{N(t), t \geq 0\}$ and call Z_k the k-th **interarrival time**, that is the interval of time between the $(k-1)$-th and the k-th event. So $S_n := \sum_{k=1}^{n} Z_k$ is the time of the n-th event. In the next section we will need to know the distribution of $\{Z_k, k \geq 1\}$. First, notice that

$$\begin{aligned} \mathbb{P}(Z_k > t | Z_{k-1} = s) &= \mathbb{P}(\text{no events in } (s, s+t] | Z_{k-1} = s) = \\ &= \mathbb{P}(\text{no events in } (s, s+t]) = \\ &= \mathbb{P}(Z_k > t), \end{aligned}$$

thanks to the independence of increments assumption. Moreover from the stationarity of increments it follows that

$$\mathbb{P}(\text{no events in } (s, s+t]) = \mathbb{P}(\text{no events in } (0, t]) = \tag{1.6}$$

$$= \mathbb{P}(Z_1 > t) = \mathbb{P}(N(t) = 0) = e^{-\lambda t}, \tag{1.7}$$

so the cumulative distribution for $\{Z_k, k \geq 1\}$ is

$$F(t) = \mathbb{P}(Z_k \leq t) = 1 - e^{-\lambda t}$$

which gives the probability density function

$$f(t) = \lambda e^{-\lambda t}. \tag{1.8}$$

A quantity slightly different from interarrival time is the **waiting time**, which is a random variable $\{\gamma_t, t \geq 0\}$ that measures the time elapsed from t to the next event. In order to find its distribution, we fix $t \geq 0$ and compute the probability of $\{\gamma_t > s\}$: this happens only if one of the mutually exclusive events $\{S_1 = Z_1 > t + s\}$, $\{S_1 = Z_1 \leq t, S_2 = Z_1 + Z_2 > t + s\}, \ldots,$ $\{S_n \leq 0, S_{n+1} > t + s\}$ occurs, so

$$\mathbb{P}(\gamma_t > s) = \mathbb{P}(Z_1 > t + s) + \sum_{n=1}^{+\infty} \mathbb{P}(S_n \leq t, S_{n+1} > t + s).$$

The last probability appearing in the previous expression can be rewritten as*

$$\begin{aligned}\mathbb{P}(S_n \leq t, S_{n+1} > t + s) &= \int_0^t \mathbb{P}(S_{n+1} > t + s | S_n = \tau)\mathbb{P}(S_n = \tau)\, d\tau = \\ &= \int_0^t \mathbb{P}(Z_{n+1} > t + s + \tau)\lambda^n \frac{\tau^{n-1}}{(n-1)!} e^{-\lambda\tau}\, d\tau,\end{aligned}$$

where we used the fact that

$$\mathbb{P}(S_n \leq t) = \mathbb{P}(N(t) \geq n) = 1 - \sum_{j=0}^{n-1} e^{-\lambda t} \frac{(\lambda t)^j}{j!}$$

and, by deriving with respect to t, we get the distribution of $\{S_n\}$

$$\mathbb{P}(S_n = t) = \lambda^n \frac{t^{n-1}}{(n-1)!} e^{-\lambda t}.$$

We can now put all together and complete the computation

$$\begin{aligned}\mathbb{P}(\gamma_t > s) &= e^{-\lambda(t+s)} + \sum_{n=1}^{+\infty} \int_0^t e^{-\lambda(t+s+\tau)} \lambda^n \frac{\tau^{n-1}}{(n-1)!} e^{-\lambda\tau}\, d\tau = \\ &= e^{-\lambda(t+s)} + e^{-\lambda(t+s)} \sum_{n=1}^{+\infty} \frac{\lambda^n}{(n-1)!} \int_0^t \tau^{n-1}\, d\tau = \\ &= e^{-\lambda(t+s)} \left(1 + \sum_{n=1}^{+\infty} \frac{(t\lambda)^n}{n!}\right) = e^{-\lambda s}.\end{aligned}$$

The cumulative distribution of $\{\gamma_t\}$ is then $\mathbb{P}(\gamma_t \leq s) = 1 - e^{-\lambda s}$, and by taking the derivative we get the waiting time distribution

$$w_t(s) = \lambda e^{-\lambda s},$$

which doesn't depend on t and is equal to the interarrival time distribution: we obtained the **memoryless property** of Poisson processes.

*Strictly speaking, for a random variable $X : \Omega \to \mathbb{R}$ the probability $\mathbb{P}(X = x)$ is null, since the measure of a single point in $\mathbb{R}$ is zero, but this abuse of notation is common in literature to indicate the probability density function.

1.3.3 Derivation of the algorithm

The process we are going to describe later is not a simple Poisson process as defined above, but it can be seen as the merging of two or more independent Poisson processes. What kind of process is it? The next theorem shows that it is again a Poisson process.

Theorem. *Consider two independent Poisson processes* $\{N_1(t), t \geq 0\}$ *and* $\{N_2(t), t \geq 0\}$*, with rates respectively* λ_1 *and* λ_2*, and let* $N(t) = N_1(t) + N_2(t)$*, then the merged process* $\{N(t), t \geq 0\}$ *is a Poisson process with rate* $\lambda = \lambda_1 + \lambda_2$*. Denote by* Z_k *the time elapsed between the* $(k-1)$*-th and* k*-th event of the merged process, and let* $I_k = i$ *if the* k*-th event is of type* i*, then*

$$\mathbb{P}(I_k = i | Z_k = t) = \frac{\lambda_i}{\lambda_1 + \lambda_2}, \qquad i = 1, 2,$$

for any $k = 1, 2, \ldots,$ *independently of* t*.*

Proof. We need first to demonstrate that Definition 2 holds for $N(t)$ too. Assumptions (i) and (ii) are easily verified since $N_1(t)$ and $N_2(t)$ are Poisson processes. Let's now deal with point (iii)

$$\begin{aligned}\mathbb{P}(N(h) = 1) &= \mathbb{P}(N_1(h) = 1, N_2(h) = 0) + \mathbb{P}(N_1(h) = 0, N_2(h) = 1) = \\ &= (\lambda_1 h + o(h))(1 - \lambda_2 h + o(h)) + (\lambda_2 h + o(h))(1 - \lambda_1 h + o(h)) = \\ &= (\lambda_1 + \lambda_2)h + o(h) = \lambda h + o(h),\end{aligned}$$

and point (iv) follows by noting that

$$\begin{aligned}\mathbb{P}(N(h) = 0) &= \mathbb{P}(N_1(h) = 0, N_2(h) = 0) = \\ &= (1 - \lambda_1 h + o(h))(1 - \lambda_2 h + o(h)) = \\ &= 1 - (\lambda_1 + \lambda_2)h + o(h),\end{aligned}$$

which implies that $\mathbb{P}(N(h) \geq 2) = o(h)$. So $\{N(t), t \geq 0\}$ is a Poisson process.

In order to prove the second assertion, call Z, Y_1 and Y_2 the random variables of interarrival times for respectively $N(t)$, $N_1(t)$ and $N_2(t)$. Then consider the following probability

$$\begin{aligned}\mathbb{P}(Z_k > t, I_k = 1) &= \mathbb{P}(Y_2 > Y_1 > t) = \\ &= \int_t^{+\infty} \mathbb{P}(Y_2 > Y_1 > t | Y_1 = s)\mathbb{P}(Y_1 = s)\, ds = \\ &= \int_t^{+\infty} \mathbb{P}(Y_2 > Y_1 > t | Y_1 = s)\lambda_1 e^{-\lambda_1 s}\, ds = \\ &= \int_t^{+\infty} e^{-\lambda_2 s}\lambda_1 e^{-\lambda_1 s}\, ds = \frac{\lambda_1}{\lambda_1 + \lambda_2} e^{-(\lambda_1+\lambda_2)t}\end{aligned}$$

where we used what we got in (1.6) and (1.8). Take $t = 0$ in the last expression

$$\mathbb{P}(I_k = 1) = \frac{\lambda_1}{\lambda_1 + \lambda_2},$$

and we already know that, according to (1.6), since $\{N(t)\}$ is a Poisson process,

$$\mathbb{P}(Z_k > t) = e^{-(\lambda_1+\lambda_2)t},$$

hence, considering also that $\mathbb{P}(Z_k > t, I_k = 1) = \mathbb{P}(Z_k > t)\mathbb{P}(I_k = 1)$, we get that $\mathbb{P}(I_k = 1 | Z_k = t) = \lambda_1/(\lambda_1 + \lambda_2)$, as we needed. Notice that the result is analogous when $I_k = 2$. □

Consider a process $\{N(t)\}$ resulting in the merging of many others $\{N_i(t)\}$, with different rates λ_i, $i = 1, \ldots, M$. It could be counting the changes of the configuration of a system that has M states at disposal. We know that $\{N(t)\}$ is a Poisson process with rate $\lambda = \sum_{i=1}^{M} \lambda_i$, so the waiting time distribution is $w_{t_0}(t) = \lambda e^{-\lambda t}$, for any t_0, and, according to the previous theorem, the probability that at time t the i-th process occurs, given that something actually happens (that is $Z_k = t$ for some $k > 0$), is just λ_i/λ.

The **stochastic simulation algorithm**, or Gillespie's algorithm, gives a recipe to generate a sample of the processes $\{N_i(t), t \geq 0\}$ that satisfies the above conditions.[†] In order to do so, it needs to tell us two things:

- the time between an event and the next, and
- what event occurs at that time.

We already know that the first is a random variable $t \in [\,0, +\infty)$ with exponential distribution $w_{t_0}(t) = \lambda e^{-\lambda t}$ which doesn't depend on the initial time t_0, so we just need to generate exponentially distributed random numbers. What we actually did is to map a new random variable ξ to t, via $t = \phi(\xi)$, such that ξ is uniformly distributed on $[\,0,\, 1\,]$. In this way we just need to draw random numbers ξ from a uniform distribution on the interval $[\,0,\, 1\,]$ and then map them to $\phi(\xi)$ which will be automatically distributed as we wish. In a formal way, the function ϕ needs to be an isometry between $[\,0,\, 1\,]$, with probability measure given by the cumulative distribution of ξ, $dF(\xi) = d\xi$, and $[\,0, +\infty)$, with the probability measure of t, $dF(t) = \lambda e^{-\lambda t} dt$, so we impose that

$$dP(\xi) = \phi^* dP(t),$$

where ϕ^* denotes the pullback via ϕ,

$$dP(\xi) = d\xi = \frac{\partial \phi}{\partial \xi} \lambda e^{-\lambda \phi(\xi)}\, d\xi = \left(-\frac{\partial}{\partial \xi} e^{-\lambda \phi(\xi)} \right) d\xi.$$

By integration between 0 and ξ, and imposing that $\lim_{t \to +\infty} \xi = 1$, we get

$$t = -\frac{1}{\lambda} \log(1 - \xi).$$

We can simplify it a little further by redefining $(1 - \xi) \to \xi$, as $1 - \xi$ is uniformly distributed if ξ is. As a result,

$$t = -\frac{1}{\lambda} \log(\xi).$$

Once we have a sample of times between an event and the next, we need to know at each time what event occurs among the M possibilities. To do so we use another uniformly distributed random number $\zeta \in [\,0,\, \lambda\,]$ and we pick the j-th event if

$$\sum_{i=1}^{j-1} \lambda_i < \zeta \leq \sum_{i=1}^{j} \lambda_i.$$

Using the steps we just described, we are able to computationally build a sample of the process $\{N(t)\}$ by means of just two uniformly distributed random numbers.

[†]The algorithm also works with nonhomogeneus Poisson processes, that is if the rates are time dependent. However, this will not be the case for us, so we ignore it.

1.4 Chemical master equation

1.4.1 Combinatorial kinetics

As we will explain better in the next chapters, the system studied in this work consists of interacting particles of different species whose quantity evolves according to a network of chemical reactions, during which particles can be destroyed, created or combined. In a deterministic system, that is, when we are dealing with a large number of particles whose concentrations change incrementally, their dynamics during a chemical reaction is described using the **law of mass action**. Instead, if we are interested in the microscopic behaviour, we have to deal with a small number of particles subjected to high stochastic fluctuations, and the system can be modelled using what is usually called a "birth and death" process. Then, we assume that a specific number of particles for each species is created or destroyed in a given event.

Gardiner, in his book [3], calls this approach **combinatorial kinetics**, because we assume that the transition probability of a system due to a reaction between particles is proportional to the number of possible ways in which those particles could have met. Suppose we need to describe a system of S species interacting via R reactions. The number of particles $\{X_s(t)\}$ of the species s, with $s = 1, \dots, S$, is a stochastic process. In full generality, we can write the r-th reaction, with reaction rate k_r, as

$$\sum_{s=1}^{S} N_s^r \xrightarrow{k_r} \sum_{s=1}^{S} M_s^r \qquad \text{with } r = 1, \dots, R,$$

where N_s^r and M_s^r are the number of particles of the species s involved respectively on the left- and right-hand side of reaction r. This means that, if the reaction r occurs, the amount of X_s is changed by $M_s^r - N_s^r$.

Consider a single species, that is fix s, in a reaction r. To describe its dynamics, we need to know in how many ways N_s^r particles, out of x_s,[‡] can meet. This is given by the number of **combinations** of N_s^r items from a collection of x_s, that is how many subsets of N_s^r distinct elements we can extract from a collection of x_s elements, regardless of their order. This number is given by the binomial coefficient

$$\binom{x_s}{N_s^r} = \frac{x_s!}{N_s^r!(x_s - N_s^r)!} \qquad \text{if } N_s^r \leq x_s.$$

The previous reasoning applies for every species involved in the reaction, so the total number of ways in which all the required particles can meet and hence react is given by

$$\prod_{s=1}^{S} \frac{x_s!}{N_s^r!(x_s - N_s^r)!}.$$

According to the combinatorial kinetics assumption we can now write the transition probabilities for a system of particles of S species interacting via a reaction r as described above

$$t_r(x_1, \dots, x_S) := k_r \prod_{s=1}^{S} \frac{x_s!}{(x_s - N_s^r)!}, \tag{1.9}$$

where we absorbed the $N_s^r!$ factor of the denominator into the reaction rate k_r since it is just a constant that depends on the reaction only. Notice that t_r depends indirectly on time since the value of x_s changes with time.

[‡]I use a lower case symbol x_s to indicate a realization of the process X_s

Now we have all we need to write the master equation which describes the time evolution of the probability density function for the stochastic processes $\{X_s(t)\}$, $s = 1, \dots, S$, which is called **chemical master equation**

$$\begin{aligned}\partial_t f(\{x_s\}, t) = \sum_{r=1}^{R} \big[& t_r(\{x_s - (M_s^r - N_s^r)\}) f(\{x_s - (M_s^r - N_s^r)\}, t) + \\ & - t_r(\{x_s\}) f(\{x_s\}, t)\big],\end{aligned} \tag{1.10}$$

where we used a compact notation to indicate all the arguments.

1.4.2 Generating function and mean-field approximation

Solving the chemical master equation directly from (1.10) is very difficult. For this reason we introduce the **moment-generating function** for the probability density $f(\{x_s\}, t)$

$$G(\{\alpha_s\}, t) := \sum_{x_1=0}^{+\infty} \cdots \sum_{x_S=0}^{+\infty} \alpha_1^{x_1} \cdots \alpha_S^{x_S} f(\{x_s\}, t).$$

It's called "moment-generating" because its derivatives give the moments of f, for example for any $k = 1, \dots, S$

$$\partial_{\alpha_k} G|_{\{\alpha_s=1\}} = \sum_{x_k=0}^{+\infty} x_k f(x_k, t) = \mathrm{E}[X_k]$$

$$\partial^2_{\alpha_k} G|_{\{\alpha_s=1\}} = \sum_{x_k=0}^{+\infty} x_k(x_k - 1) f(x_k, t) = \mathrm{E}[X_k^2] - \mathrm{E}[X_k],$$

and so on for higher order partial derivatives. We can exploit this definition to rewrite equation (1.10) into a partial differential equation (in the following just "PDE") for G. In fact, the transition probabilities $t_r(\{x_s\})$ are polynomials of order N_s^r, so the corresponding terms in (1.10) can be replaced by an appropriate combination of partial derivatives (with maximum order N_s^r) of the moment-generating function. This will become clearer in the next chapter when we will consider some examples of chemical networks. The PDE for G will be of the form

$$\partial_t G(\{\alpha_s\}, t) = \sum_{i=1}^{S} A_i(\{\alpha_s\}) \partial_{\alpha_i} G + \sum_{i,j=1}^{S} B_{ij}(\{\alpha_s\}) \partial_{\alpha_i} \partial_{\alpha_j} G + \cdots, \tag{1.11}$$

where A_i and B_{ij} are polynomials, and it is a homogeneous linear partial differential equation (if it stops at the second order it is called Kolmogorov's equation). The initial condition is given by the number of particles at $t = 0$ for each species $x_{1,0}, \dots, x_{S,0}$, so that

$$f(x_1, \dots, x_S, t = 0) = \delta_{x_1, x_{1,0}} \cdots \delta_{x_S, x_{S,0}}$$

and the moment-generating function at $t = 0$ becomes

$$G(\{\alpha_s\}, t = 0) = \alpha_1^{x_{1,0}} \cdots \alpha_S^{x_{S,0}}.$$

Consider now the PDE for G (1.11). By taking the derivatives of both sides with respect to α_k (with $1 \le k \le S$) and then setting all the $\{\alpha_s\}$ to 1 we get a differential equation that describes the time evolution of mean values

$$\partial_t \langle x_k \rangle = \sum_{i=1}^{S} \tilde{A}_i \langle x_i \rangle + \sum_{i,j=1}^{S} \tilde{B}_{ij} \langle x_i x_j \rangle + \cdots,$$

with $\tilde{A}_i$ and $\tilde{B}_{ij}$ constants. Iterating this procedure, such an equation for the time evolution of every needed combination of x_k's can be written down. If lucky, one ends up with a closed linear system of ordinary differential equation; by solving it, we get the time evolution of the mean values of all the variables $\{X_k\}$. Similarly one can proceed for higher order moments: we just need to take higher order derivatives of (1.11).

Apart from the simplest cases, the procedure described above will not return a closed system of equations: usually, by computing an equation for the n-th moment, the $(n+m)$-th moment appears. For this reason, in order to get a finite set of equations, some kind of approximation is required. The crudest one is the **mean-field approximation**, which assumes that the mean value of a product of variables is the product of the mean values

$$\mathrm{E}[XY] \simeq \mathrm{E}[X]\mathrm{E}[Y].$$

Notice that, by the definition of covariance,

$$\mathrm{Cov}[X,Y] = \mathrm{E}[XY] - \mathrm{E}[X]\mathrm{E}[Y],$$

so the mean-field approximation consists of neglecting the covariance, which, as we will see, is not always negligible.

By means of this approximation we are able to close, by force, the system of equations for mean values or higher moments. In particular, the equations we get for mean values correspond to the deterministic equation obtained using the law of mass action,[§] so we define the deterministic motion to be the one resulting as a solution of the equations for mean values using the mean-field approximation.

As we will see, the methods explained above to analytically extract information about the processes $\{X_s(t)\}$ are not always usable. For this reason we would like to explore the behaviour of the chemical system by building samples using Gillespie's algorithm. At first glance it seems that Gillespie's algorithm can't be applied in this scenario: the algorithm works for a compound Poisson process resulting from the merging of many Poisson processes with constant rates, while in our case the transition rates of the chemical system are described by (1.9) which are state-dependent. In order to avoid this problem notice that if we fix the state of the system, that is the values of $\{X_s(t)\}$, the transition rates $t_r(x_1, \dots, x_S)$ are, of course, fixed as well; this tells us that between two successive events the system is indeed described by a Poisson process, so we can use the Gillespie's algorithm to determine when the next event will happen and what it will be. Then we have to update the state of the system accordingly and consider a new Poisson process, with the updated rate, which describes the system until the next event, and so on.

[§]There is a small difference, actually. Consider, for example, a reaction of the type $2\mathrm{A} \to \mathrm{B}$. The law of mass action will produce an equation of motion quadratic with respect to the concentration of A. Combinatorial kinetics, instead, since the particles are indistinguishable, leads to a term proportional to $c(c-1)$, where c is the amount of A. This subtlety is not important for our purposes.

Chapter 2

Flagellar motor assembly

The aim of this chapter is to describe in detail the system we are studying, the flagellar motor assembly (FMA), part of the flagellum of *Shewanella putrefaciens*. We first introduce all the reactions involved and assumptions we made, then we start using the theory developed in Chapter 1 and present our results. We want to gain a deeper understanding of how the system behaves and how it manages to solidly control the production of resources needed to build the flagellum. The biochemical information used in this chapter were gently provided to us by Dr. Devid Mrusek.*

2.1 Chemical network

2.1.1 Interactome

The key element of the interactome that leads to the production of the proteins which the FMA is made of is the master regulator A: it activates the transcription of the gene X and thus starts the production of the required proteins. In order to trigger the gene transcription, A first needs to be activated itself: it transforms into its dimeric form (A_2), then the dimers gather around the gene hexamerize (A_6-X) and trigger the protein production while decaying again to their dimeric state.

So, once A is activated, it allows the transcription of proteins G, M and FliG and then it detaches from the gene returning to its dimeric form. These proteins are the ones directly involved in the production of the building block of FMA: M and G interact and give MG, then FliG binds to MG and frees G while producing the "brick" of the FMA, M-FliG.

An important role is played by the protein G. In fact, G, which is also produced by other chemical networks of the cell, gives rise to a negative feedback mechanism. Similarly to what happens for protein A, G dimerizes into G_2. The opposite reaction, in which G_2 is dismantled back to G, is enhanced by the presence of M and A. Dimers G_2 can then attach to A in an irreversible way causing a consumption of resources: the more G is present, the less A triggers the gene transcription.

2.1.2 Assumptions

In describing this system we want to keep the analysis at a simple level, neglecting some of the specific biological features that could conceal a deeper understanding of the general mechanism.

*Philipps University Marburg, LOEWE Center for Synthetic Microbiology and Department of Chemistry, Prof. Dr. Gert Bange laboratory.

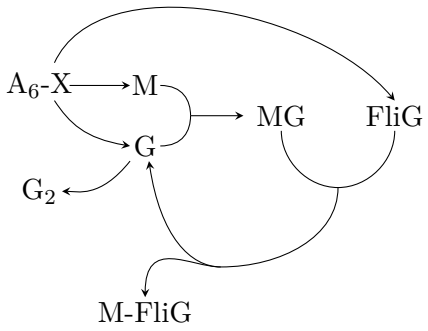


Figure 2.1: A scheme depicting part of the interactome described in Section 2.1.1, from the gene transcription to the production of the building block M-FliG.

We know, for instance, that all proteins involved are subjected to degradation, the rate of which should be similar for all the different species. In this work, we will neglect degradations, which means that we are assuming the process to be faster than degradation processes.

At the end of all the chemical steps depicted above, the building blocks are involved in the construction of the FMA itself. This means that, as the FMA ring grows, the number of M-FliG decreases. We assume that the process produces the whole needed amount of resources before starting to build the FMA, so we will not study the assembling process itself. Alternatively, we assume that the number of M-FliG does not influence the interactome we describe. For the same reason questions about the efficiency of the assembling process, still very important and strictly correlated to our work, will not be addressed here. Even when the FMA is complete, the degradation of its constituents is still going on. So, in the biological system, the cell needs to continuously produce further M-FliG to balance the degradation.

As it is explained in the following paragraph, we avoid to model the negative feedback via a choice of state-dependent reaction rate like an Hill function, we rather use simple reactions with constant rates.

2.1.3 Reactions system

More in detail, the specific reactions involved in the previous description, embodying our assumptions, are described below. For each of them we compute the transition probability according to equation (1.9). We will use the letter λ (k) to indicate the reaction rate of a forward (reverse) reaction, with a subscript to differentiate the different cases. The nomenclature for the random variable which describes each protein species will be clarified later.

As we said before, the activation procedure of the master regulator A consists of three steps:

- first A dimerizes according to an irreversible reaction

$$2\mathrm{A} \xrightarrow{\lambda_d} \mathrm{A}_2$$
$$t_d = \lambda_d n^A (n^A - 1),$$

  where d stands for "dimerization", and n^A is the number of A particles;

- then the dimers A_2 gather around the three free sites on the gene X

$$
\begin{aligned}
&A_2 + X \underset{k_X}{\overset{\lambda_X}{\rightleftharpoons}} A_2\text{-}X,\\
&t_1^{\rightharpoonup} = \lambda_X n_u^{A_2} n_u^X\\
&t_1^{\leftharpoondown} = k_X n_b^{A_2} n_1^X,\\
&A_2 + A_2\text{-}X \underset{k_X}{\overset{\lambda_X}{\rightleftharpoons}} (2A_2)\text{-}X,\\
&t_2^{\rightharpoonup} = \lambda_X n_u^{A_2} n_1^X\\
&t_2^{\leftharpoondown} = k_X n_b^{A_2} n_2^X,\\
&A_2 + (2A_2)\text{-}X \underset{k_X}{\overset{\lambda_X}{\rightleftharpoons}} (3A_2)\text{-}X,\\
&t_3^{\rightharpoonup} = \lambda_X n_u^{A_2} n_2^X\\
&t_3^{\leftharpoondown} = k_X n_b^{A_2} n_3^X,
\end{aligned}
$$

  where the superscript $\rightharpoonup$ ($\leftharpoondown$) stands for "forward reaction" ("reverse reaction"), and we use $n_u^{A_2}$ to indicate the number of unbound A_2, and similarly $n_b^{A_2}$ for the bound ones; $n_u^X = 1$ if the gene has nothing attached to it, otherwise it is zero, and the variable n_i^X is 1 if there are i A_2 on the gene X, otherwise $n_i^X = 0$; notice also that we are assuming the chemical rates for the attachment and detachment of one, two or three dimers to be the same;

- last, the hexamerization of A onto the gene X

$$
\begin{aligned}
&(3A_2)\text{-}X \underset{\lambda_h}{\overset{k_h}{\rightleftharpoons}} A_6\text{-}X,\\
&t_h^{\rightharpoonup} = \lambda_h n_b^{A_2}(n_b^{A_2} - 1)(n_b^{A_2} - 2)\\
&t_h^{\leftharpoondown} = k_h n^{A_6},
\end{aligned}
$$

  where h stands for "hexamerization" and n^{A_6} is the number of A_6-X.

In terms of reactions, what is pictured in Figure 2.1 becomes:

- gene transcription and production of G, M and FliG

$$
\begin{aligned}
&A_6\text{-}X \xrightarrow{\lambda_G} X + G + 3A_2\\
&t_G = \lambda_G n^{A_6},\\
&A_6\text{-}X \xrightarrow{\lambda_M} X + M + 3A_2\\
&t_M = \lambda_M n^{A_6},\\
&A_6\text{-}X \xrightarrow{\lambda_F} X + \text{FliG} + 3A_2\\
&t_F = \lambda_F n^{A_6};
\end{aligned}
$$

- G and M combine into MG

$$
\begin{aligned}
&G + M \xrightarrow{\lambda_{MG}} MG\\
&t_{MG} = \lambda_{MG} n^G n^M,
\end{aligned}
$$

  where n^G and n^M are the amount of G and M respectively;

- production of the building block of FMA

$$\text{FliG} + \text{MG} \xrightarrow{\lambda_b} \text{M-FliG} + \text{G}$$
$$t_b = \lambda_b n^F n^{MG},$$

where b stands for "building block" (or "brick"), n^F represents the number of FliG and n^{MG} the amount of MG. Note that the amount of building blocks is denoted by n^C.

Last, we need to specify the reactions involved in the negative feedback:

- first, G dimerizes via

$$2\text{G} \xrightarrow{\lambda_{GG}} \text{G}_2$$
$$t_{GG} = \lambda_{GG} n^G (n^G - 1),$$

but G_2 degrades in presence of A or M

$$\text{G}_2 + \text{A} \xrightarrow{k_A} 2\text{G} + \text{A}$$
$$t_{k_A} = k_A n^A n^{G_2},$$
$$\text{G}_2 + \text{M} \xrightarrow{k_M} 2\text{G} + \text{M}$$
$$t_{k_M} = k_M n^M n^{G_2},$$

where n^{G_2} is the amount of G_2;

- then G_2 binds to A_2, preventing it to hexamerize

$$\text{G}_2 + \text{A}_2 \xrightarrow{\lambda_f} \text{GA}$$
$$t_f = \lambda_f n^{G_2} n^{A_2},$$

where f stands for "feedback" and the amount of GA is n^{GA}.

2.2 Stochastic analysis

2.2.1 Robust counting mechanism

The set of reactions in the previous section describes how the basic unit of the FMA, the protein M-FliG, is produced. After that, the assembling of the FMA ring starts. How many of them will be formed? One could think that a constraint on the number of structures is easily achieved by limiting the production of building blocks. Inversely, it's not that simple. In principle, the amount of units produced could be proportional not only to the number of structures, but also to their size. We can avoid this problem by assuming that the size of the FMA is fixed by its ring shape. Moreover, we can't just be satisfied by whatever dynamics produces the required amount of M-FliG, because we need it to be **robust**: that quantity should vary slowly when changing initial conditions (that is how many proteins are already present at the beginning) and parameters (chemical rates), and, if it is described by a stochastic process, it should have small fluctuations. For the last requirement, we will search for a dynamics that leads to a final amount of building blocks whose variance is smaller than its mean value, that is we want it to be described by a "better" distribution than the Poisson one.

2.2.2 Chemical master equation

Thanks to the transition probabilities we found in Section 2.1.3 using combinatorial kinetics, we now know the chemical master equation for the whole system according to equation (1.10). It describes the evolution of the probability density function of all the random variables involved in the system. Since we can't directly solve it and it doesn't give any better insight about the problem, we avoid to write it explicitly. What we can do is to transform it into a PDE for a generating function and show that it can't be solved without approximations, as discussed in Section 1.4.2, because it gives rise to an infinite set of ordinary differential equations for mean values.

We will try to overcome these problems by relying on stochastic simulations of the system in order to understand, at least qualitatively, what happens if we change initial number of proteins and chemical rates. In particular, we want to learn what the role of each component of the reactions network is. Hopefully, this will help us in making more solid assumptions to simplify the analysis.

Gillespie's algorithm, derived in Section 1.3.3, is able to produce a trajectory for each species of proteins subject to the transition rates we derived. Since it is a numeric computation, we need to fix the values of all the constants involved. With regard to the starting amount of proteins, we will assume that no protein except A is present at the beginning. Later on, we will also study what happens if we change quantities of A and G. We can study how the robustness of the system evolves as function of the parameters by exploring the parameter space using an appropriate spacing. In the following section, we will compute the mean value and the variance of the final amount of building blocks produced for each value of the parameters, that is for each point of the sampled parameter space. In order to compute averages and variances, we will iterate the algorithm at least 5000 times for each point of the sampled parameter space, so that our estimates are statistically meaningful.

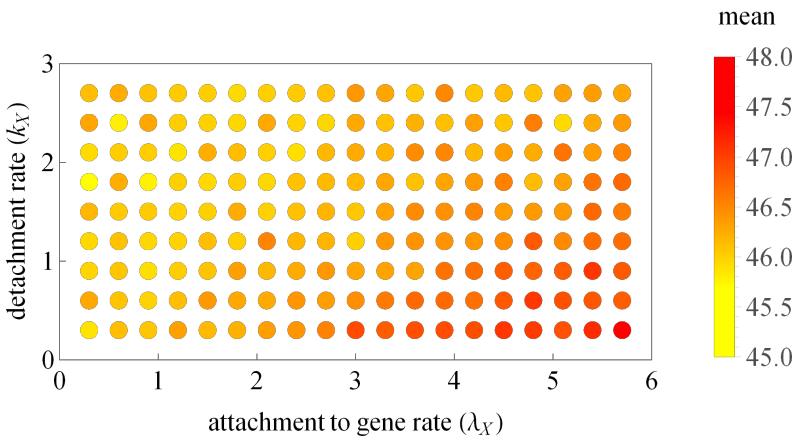
2.2.3 Role of chemical rates

The sampled parameter space described before lives in a space of dimensions given by the number of rates we used, that is 14. Since it's impossible to visualize at the same time the whole space, we will study first each step of the network separately, considering only two parameters at a time, that is a two dimensional section of the whole parameters space. The aim is to infer if some of the parameters don't influence the results from a robustness perspective. For example, we can already guess that the dimerization step of A only affects the time needed to reach the end of the process and not how many (or how robustly) building blocks are produced, because all the A's will eventually dimerize. The choice of the range of values of parameters studied with our simulations is made considering the computation time (using, for instance, very high values of k_X would make the system needlessly slow) and aiming to depict regions of parameters space where meaningful variations of the system's behaviour could have been observed.

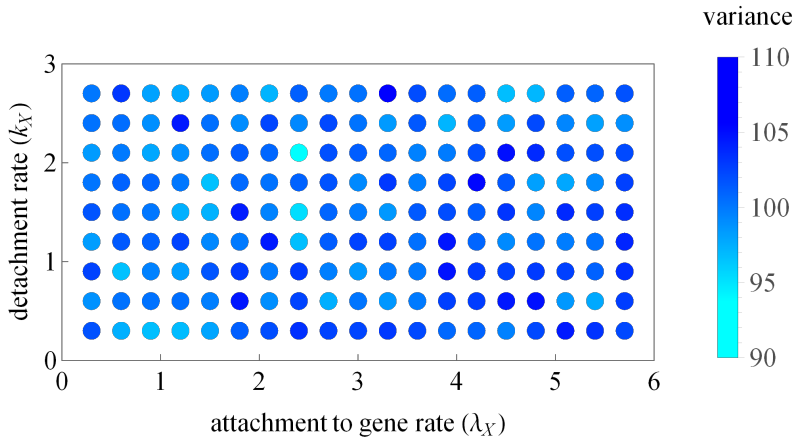
Activation of the master regulator A

Let's first take a look at the process of attachment and detachment of dimers A_2 to the gene X. All the reactions are ruled by the two rates λ_X and k_X. In Figure 2.2 the result of our simulations is presented. In particular, one can see how the mean value (red gradation) and the variance (blue gradation) of the final amount of M-FliG scale when varying λ_X and k_X. While the variances seems to be independent of the processes of attachment and detachment, one observes a different behaviour for mean values: the higher λ_X is with respect to k_X,

the more building blocks will be produced. This was to be expected since the system stops when all the dimers A_2 are consumed by the feedback mechanism. Hence, the longer A_2 stays linked to the gene, the less it will be affected by the feedback. However, it is also clear that the variation for mean values is very small (about 3 units), clearly not enough to overcome the huge stochastic fluctuations quantified by the variance. For this reason the key step that makes the system robust should be somewhere else, but we can argue that, if robustness is reached, at least the attachment and detachment to the gene process is not able to break it since its contribution to the matter is negligible.



(a) Mean value of building blocks produced.

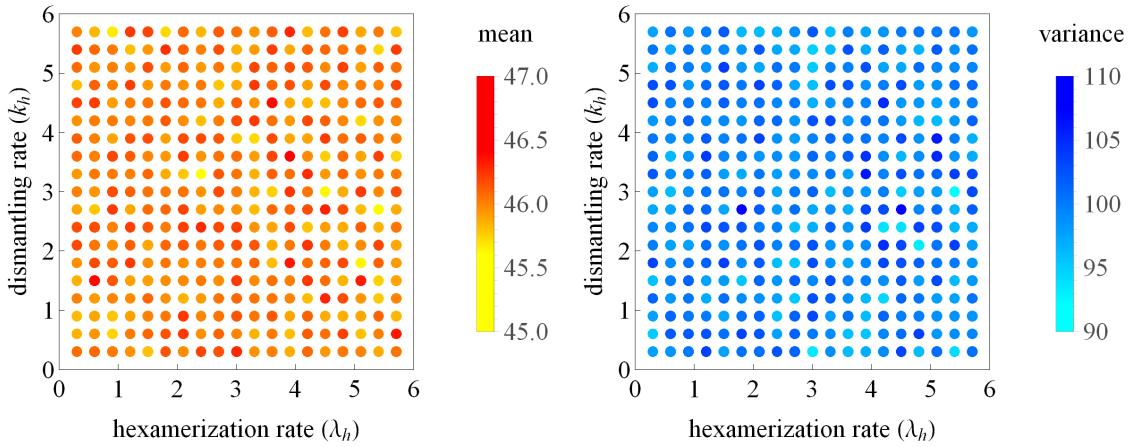


(b) Variance of building blocks produced.

Figure 2.2: Mean value and variance of the final amount of building blocks produced as function of the attachment and detachment rate of A_2 to X. All the other parameters are fixed at 0.5. We used 50 as initial number of A proteins, while the other species are initially absent.

The second and last step involved in the activation of A is the hexamerization onto the gene, controlled by the rate λ_h. Once hexamerized, it can either start the reactions sequence

that leads to the building blocks, or dismantle back to its dimer form. This last process is ruled by the rate k_h. As done before, in Figure 2.3 we plot the average and variance of the final amount of M-FliG studying a range of values for the rates between 0 and 6: no dependency is visible. In order to highlight more precisely if the mean value and variance depend on λ_h or k_h, we compute the slopes of slices of the previous graphs obtained by fixing one of the two parameters. The results are shown in Figure 2.4 (corresponding to the average plot in Figure 2.3(a)), and in Figure 2.5 (for the variance plot in Figure 2.3(b)). All the slopes are randomly distributed around zero, so there is indeed no clear dependency. This comes with no surprise since, when A is in its activated form, it contributes to both the building block production and feedback mechanism.

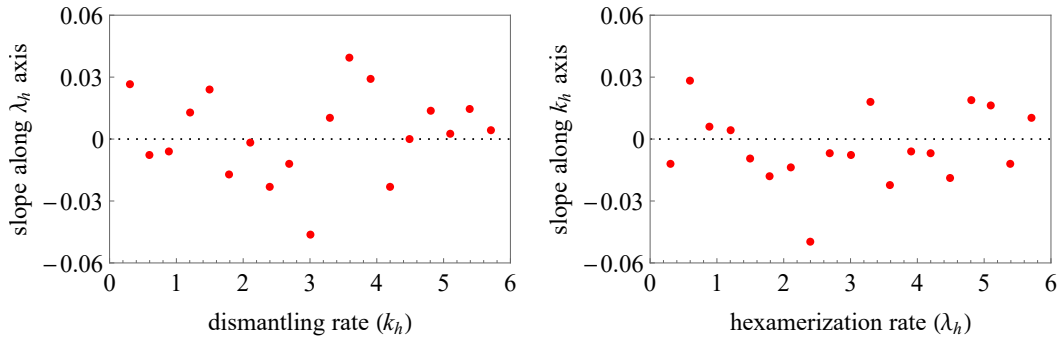


(a) Mean value of building blocks produced.

(b) Variance of building blocks produced.

Figure 2.3: Mean value and variance of the final amount of building blocks produced as function of the hexamerization and dismantling rate of A_6-X. All the other parameters are fixed at 0.5. We used 50 as initial number of A proteins, while the other species are initially absent.

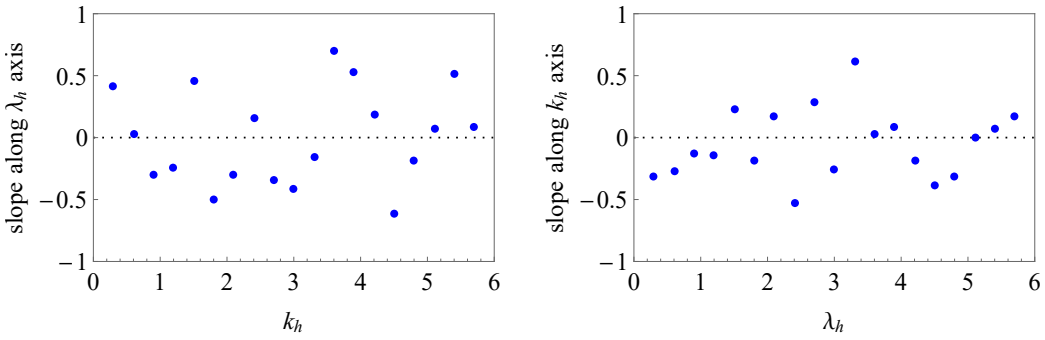
Figure 2.4: Slopes of sections of the average graph in Figure 2.3(a) parallel to...



(a) ... the λ_h axis.

(b) ... the k_h axis.

Figure 2.5: Slopes of sections of the variance graph in Figure 2.3(b) parallel to...



(a) ... the λ_h axis.

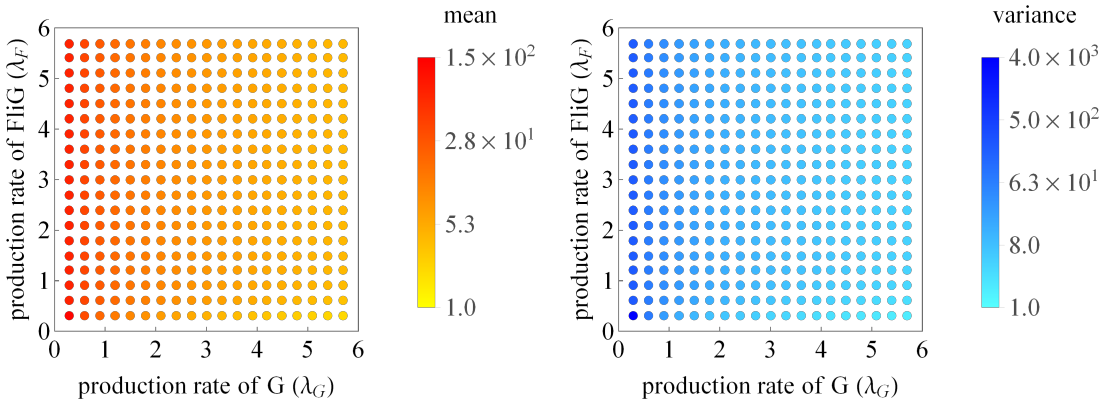
(b) ... the k_h axis.

Production of building blocks

Once the master regulator appears as a hexamer bound to the gene, that is in its activated form, the network core of the interactome can start producing M-FliG as depicted in Figure 2.1. The first steps are the irreversible reactions for the production of G, M and FliG, respectively, ruled by the rates λ_G, λ_M and λ_F. In particular, Figure 2.6 shows the behaviour of the system under variations of λ_G and λ_F. Here, something different with respect to the previous reactions happens. First, notice that there is a huge dependency of both the mean value and variance on the production rate of G, rapidly decreasing of two or three orders of magnitude. This seems obvious, since G is involved in the consumption of the master regulator. So, when its production rate is small, the feedback is weaker and more building blocks are produced. Another new behaviour is highlighted in Figure 2.7 which shows a section of the plots in Figure 2.6 obtained by fixing $\lambda_G = 5.7$. For large value of λ_G, the average of the final amount of building blocks is able to overcome the variance, but there are still two problems: the amount of M-FliG produced is very small, and the region of values of λ_G and λ_F that leads to this kind of robustness is very narrow.

Until now, the major effect we have seen is that of the production rate of G, λ_G. For this reason, it makes sense to compare the role of the other reaction rates to that of λ_G. Let's make a step further into the reactions network and consider the production of MG ruled by λ_{MG}. Figure 2.8 shows the dependency on the production rate of G, but with an important difference: if we take the ratio at each point between the variance and the mean value (that is we compute the **Fano factor**) we get Figure 2.9, where we observe that this time the region of parameters that makes the system stochastically robust is larger than in the previous case. As already done before, we check the dependency on the production rate of MG by computing the slope along the λ_{MG} axis, of both the mean values and variances. The result is plotted in Figure 2.10: for small values of λ_G, there is a non-trivial role of λ_{MG}, because in that case it crucially influences whether G should participate in the feedback mechanism or the production of MG itself, while for larger values of λ_G the dependency on λ_{MG} disappears.

The last reaction we need to check is the production of the actual building blocks M-FliG, controlled by λ_b. As explained above, we compare it with the production rate of G. The results presented in Figure 2.11 are very similar to the previous ones: as expected (see Figure 2.13), λ_b is irrelevant for the system, because at that stage of the interactome the only reaction that can occur is, indeed, the production of building blocks. Figure 2.12 shows again that the region that leads to stochastic robustness is quite large and depends solely on λ_G.



(c) Mean value of building blocks produced.

(d) Variance of building blocks produced.

Figure 2.6: Mean value and variance of the final amount of building blocks produced as function of the production rate of G and FliG. All the other parameters are fixed at 0.5. We used 50 as initial number of A proteins, while the other species are initially absent. Due to the huge variation, we used a logarithmic color scale.

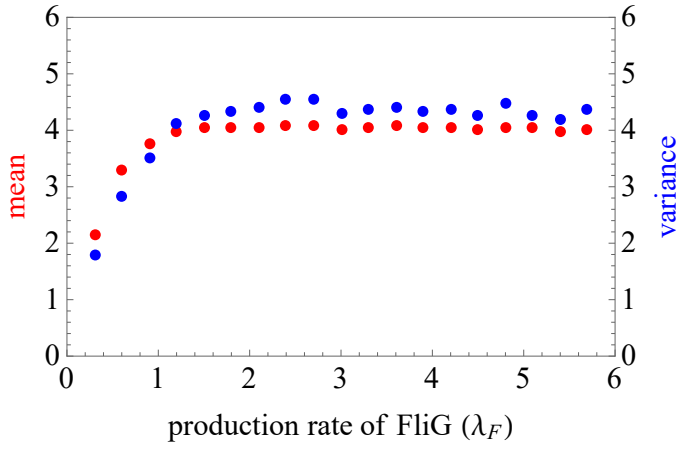
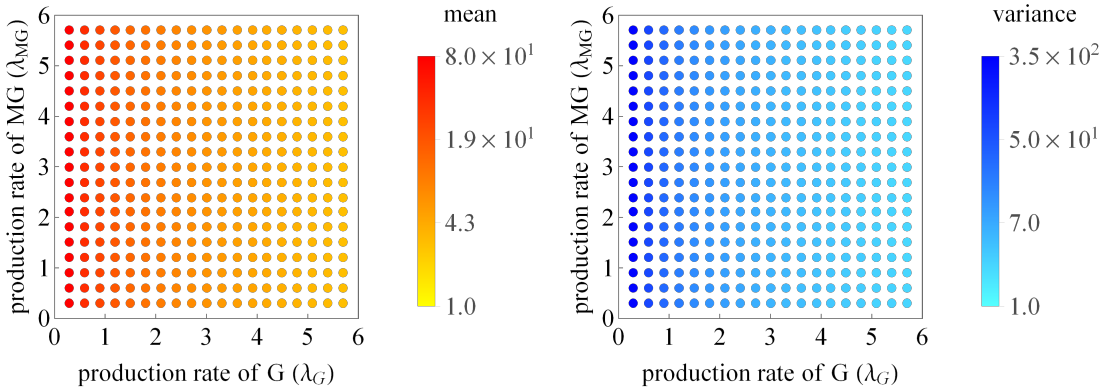


Figure 2.7: Section of Figure 2.6 fixing $\lambda_G = 5.7$.



(a) Mean value of building blocks produced.

(b) Variance of building blocks produced.

Figure 2.8: Mean value and variance of the final amount of building blocks produced as function of the production rate of G and MG. All the other parameters are fixed at 0.5. We used 50 as initial number of A proteins, while the other species are initially absent. Due to the huge variation, we used a logarithmic color scale.

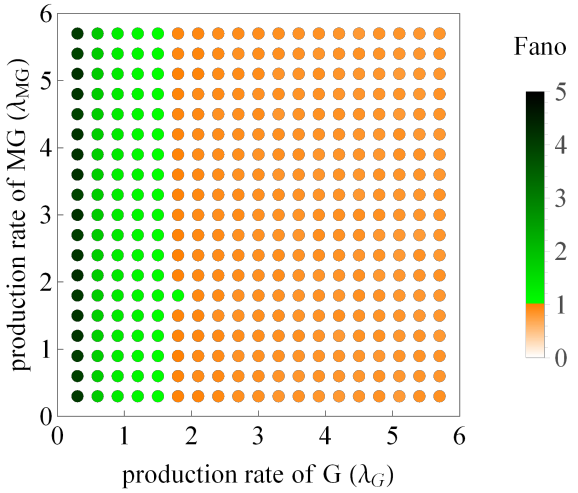


Figure 2.9: Fano factor corresponding to Figure 2.8. The different color scale highlights the region where the Fano factor is smaller than 1, that is, where the variance is smaller than the mean value.

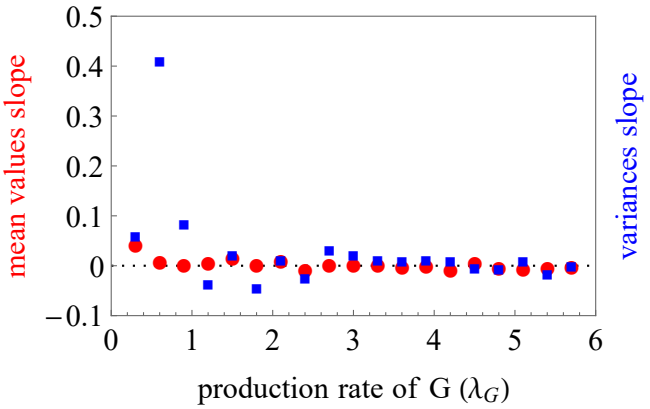
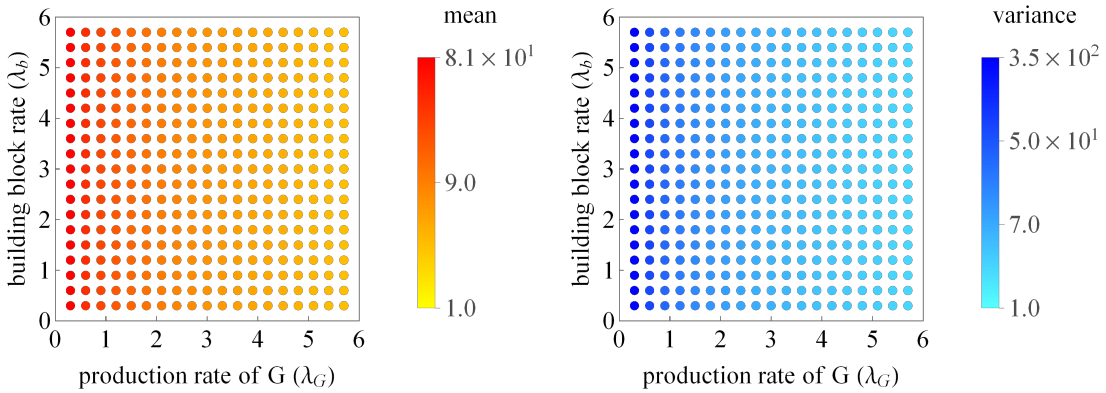


Figure 2.10: Slopes of the graphs for the average (red) and variance (blue) of Figure 2.8 parallel to the λ_{MG} axis.



(a) Mean value of building blocks produced.

(b) Variance of building blocks produced.

Figure 2.11: Mean value and variance of the final amount of building blocks produced as function of the production rate of G and M-FliG. All the other parameters are fixed at 0.5. We used 50 as initial number of A proteins, while the other species are initially absent. Due to the huge variation, we used a logarithmic color scale.

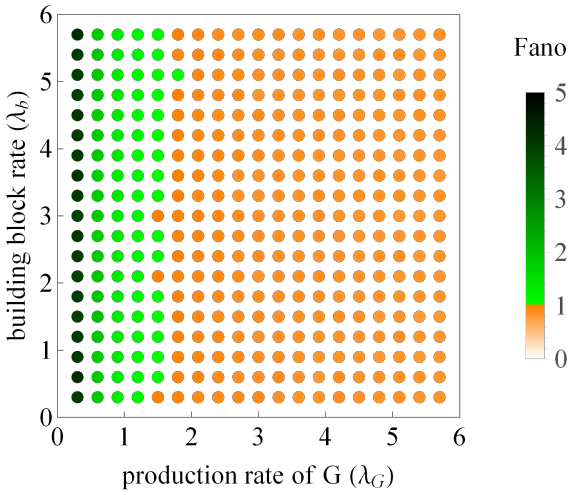


Figure 2.12: Fano factor corresponding to Figure 2.11. The different color scale highlights the region where the Fano factor is smaller than 1, that is, where the variance is smaller than the mean value.

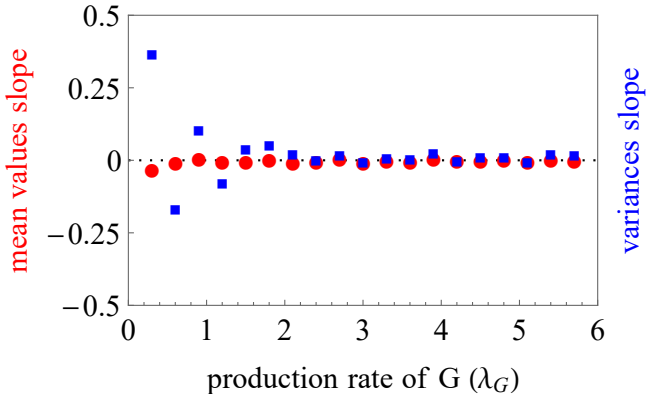


Figure 2.13: Slopes of the graphs for the average (red) and variance (blue) of Figure 2.11 parallel to the λ_b axis.

Taken together, all these simulations show that λ_G has a crucial role for reaching robustness. In particular, λ_G must be larger than λ_F and λ_M. In this way, the channel that produces G is preferred to the reactions producing FliG and M, thus it favours the feedback loop.

Feedback mechanism

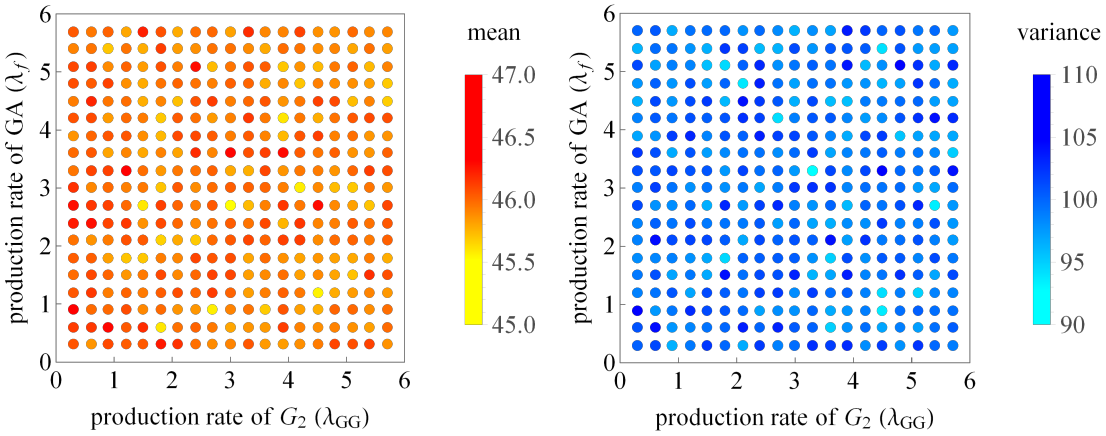
The system stops producing building blocks when all the resources are consumed. The reactions in the interactome responsible for stymieing the growth of M-FliG are the dimerization of G (which is reversible, as we have seen) and the binding of G_2 to A_2. Since the last reaction is not reversible, all A_2 are doomed to be bounded to a G_2 and can't be freed: this happens when all reactions stop occurring. In Figure 2.14, we study the role of the production rate of G_2 (λ_{GG}) comparing it with the production rate of GA (λ_f). It seems there is no visible dependency on these two parameters, but we can check it better by computing the slopes of the two graphs along sections parallel to the axes, as pictured in Figure 2.15 and Figure 2.16: the mean value of building blocks produced slightly decreases when λ_{GG} increases since there are more negative slopes than positive ones. This makes sense because when λ_{GG} is larger more G could be involved in the feedback loop. The other graphs, however, suggest no dependency on the two analysed parameters.

In principle, it could also be interesting to compare the role of the production of MG (ruled by λ_{MG}) and the dimerization of G (ruled by λ_{GG}), because they compete for the same resource, that is, the protein G. Figure 2.17 indicates that there is again no dependency. A deeper insight, as already done in similar cases, is given in Figure 2.18 and Figure 2.19: most of the slopes in Figure 2.18(a) are positive, even though very small, and that's exactly what was expected since a greater production of MG means that more G will be subtracted from the feedback loop. It is not a big dependency, though, because the transition probability for dimerization depends quadratically on the number of G, while the one for the production of MG only does so linearly.

We are still missing the role of the two degradation reactions of G_2, ruled by the parameters k_A and k_M. Since they are involved in the dynamics of G, we compare their behaviour with respect to the production rate of MG again.

First, in Figure 2.20, we consider the degradation of G_2 caused by A. No dependency is visible, so we refer again to the analysis of the slope of sections as pictured in Figure 2.21 and 2.22: we see, in Figure 2.21(a), that an increase in the value of λ_{MG} enhances the production of building blocks regardless of the degradation of G_2, similarly to what we have seen in the previous case.

Last, we repeat the same procedure for the degradation of G_2 caused by the presence of M in Figures 2.23 to 2.25. Here something new happens: there seems to be a positive correlation between the role of λ_{MG} and k_M. It's easy to understand why: if the degradation of G_2 is stronger, a larger amount of G is available for MG to be produced and, if the production of MG is strengthened too, that excess of G will be deployed for MG rather than G_2. So why wasn't this effect visible for k_A too? That's because in the first case the degradation of G_2 was prompted by A, and at that stage of the process all proteins A have already been dimerized. In contrast, in the second case, the degradation is triggered by M, which is much more abundant.

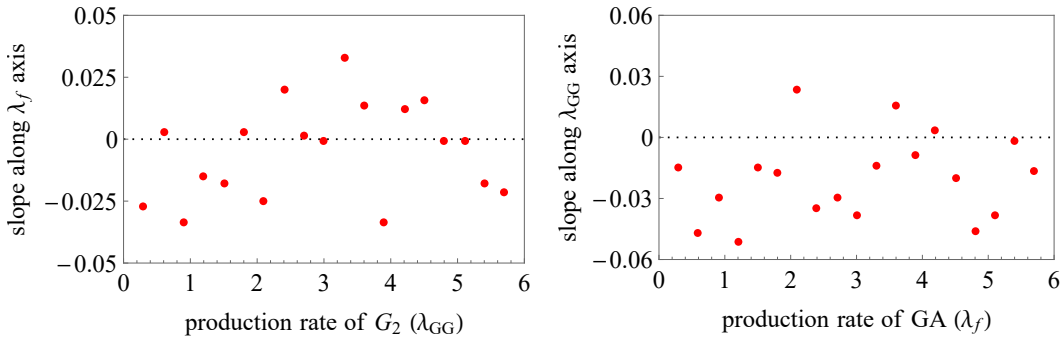


(a) Mean value of building blocks produced.

(b) Variance of building blocks produced.

Figure 2.14: Mean value and variance of the final amount of building blocks produced as function of the production rate of G_2 and GA. All the other parameters are fixed at 0.5. We used 50 as initial number of A proteins, while the other species are initially absent.

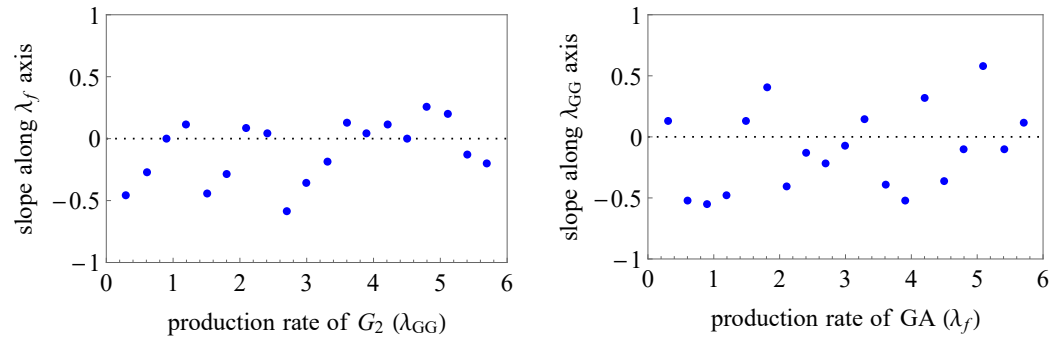
Figure 2.15: Slopes of sections of the average graph in Figure 2.14(a) parallel to...



(a) ... the λ_f axis.

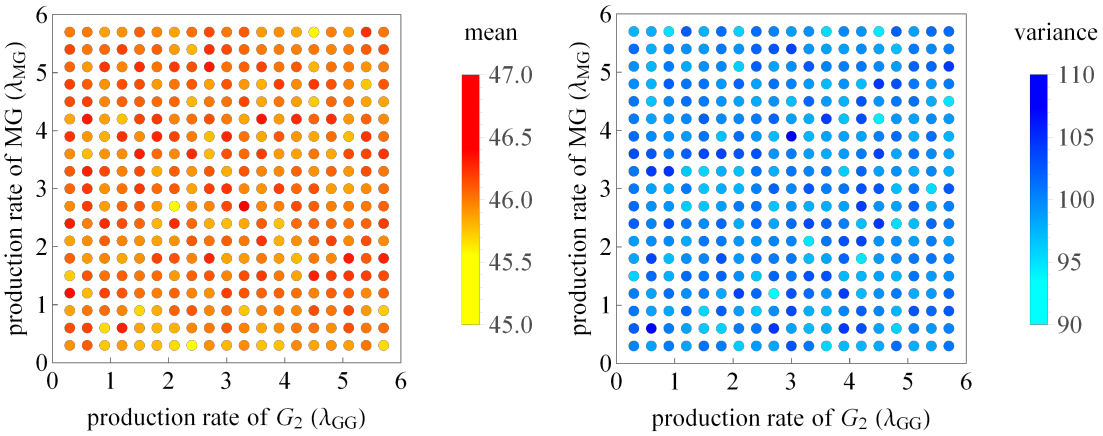
(b) ... the $lambda_{GG}$ axis.

Figure 2.16: Slopes of sections of the variance graph in Figure 2.14(b) parallel to...



(a) ... the λ_f axis.

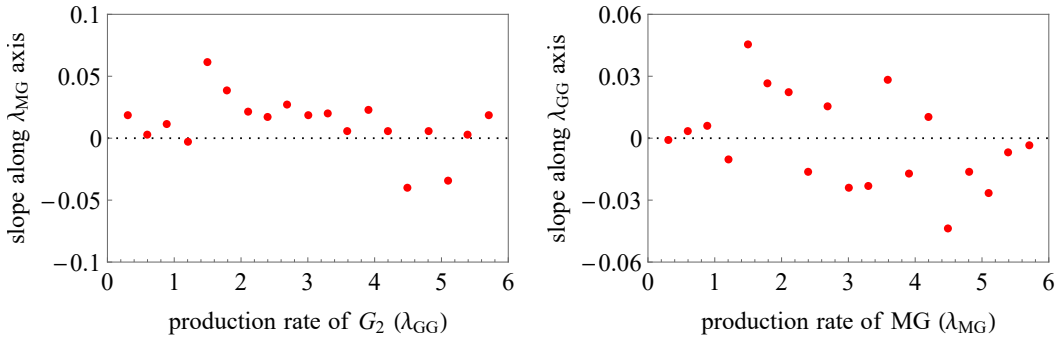
(b) ... the λ_{GG} axis.



(a) Mean value of building blocks produced.

(b) Variance of building blocks produced.

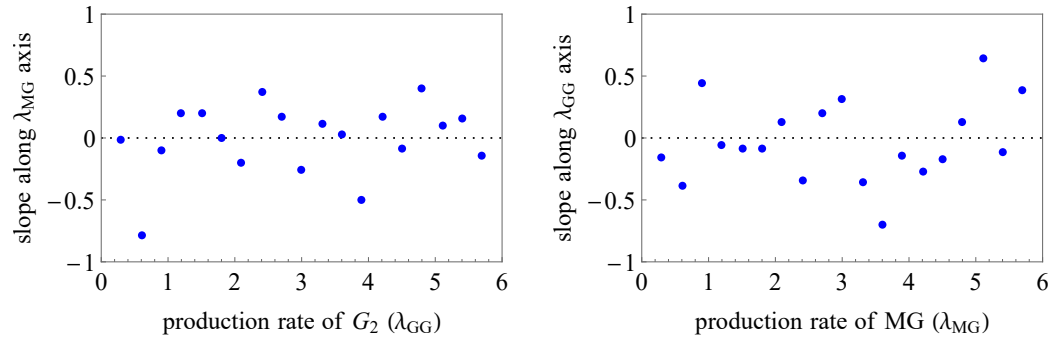
Figure 2.17: Mean value and variance of the final amount of building blocks produced as function of the production rate of G_2 and MG. All the other parameters are fixed at 0.5. We used 50 as initial number of A proteins, while the other species are initially absent.



(a) ... the λ_{MG} axis.

(b) ... the λ_{GG} axis.

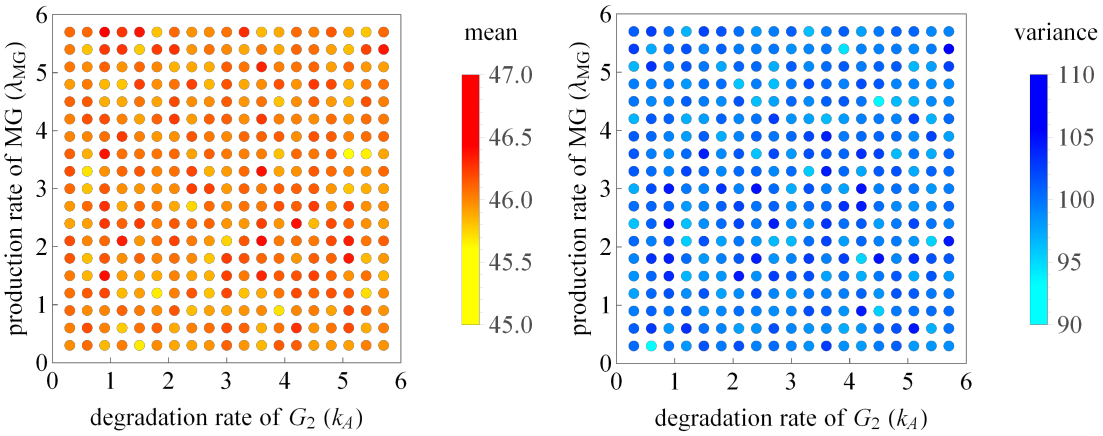
Figure 2.18: Slopes of sections of the average graph in Figure 2.17(a) parallel to...



(a) ... the λ_{MG} axis.

(b) ... the λ_{GG} axis.

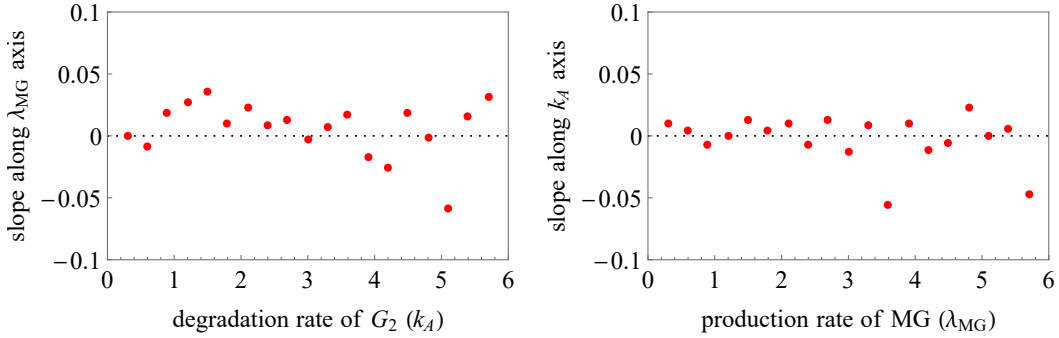
Figure 2.19: Slopes of sections of the variance graph in Figure 2.17(b) parallel to...



(a) Mean value of building blocks produced.

(b) Variance of building blocks produced.

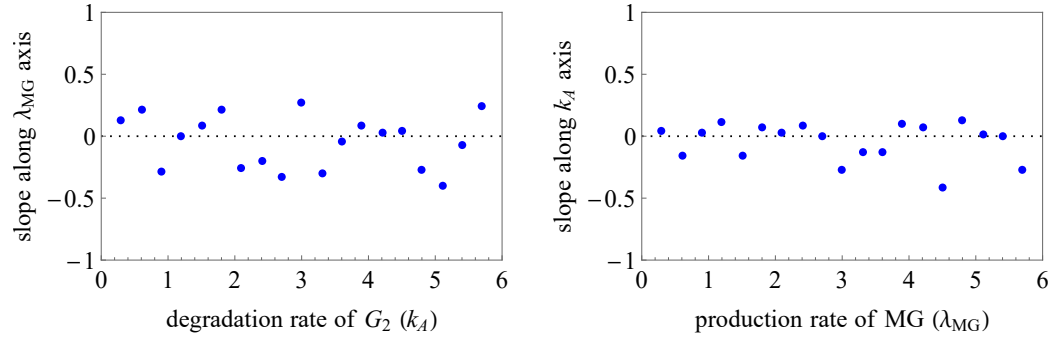
Figure 2.20: Mean value and variance of the final amount of building blocks produced as function of the degradation rate of G_2 triggered by A and the production rate of MG. All the other parameters are fixed at 0.5. We used 50 as initial number of A proteins, while the other species are initially absent.



(a) ... the λ_{MG} axis.

(b) ... the k_A axis.

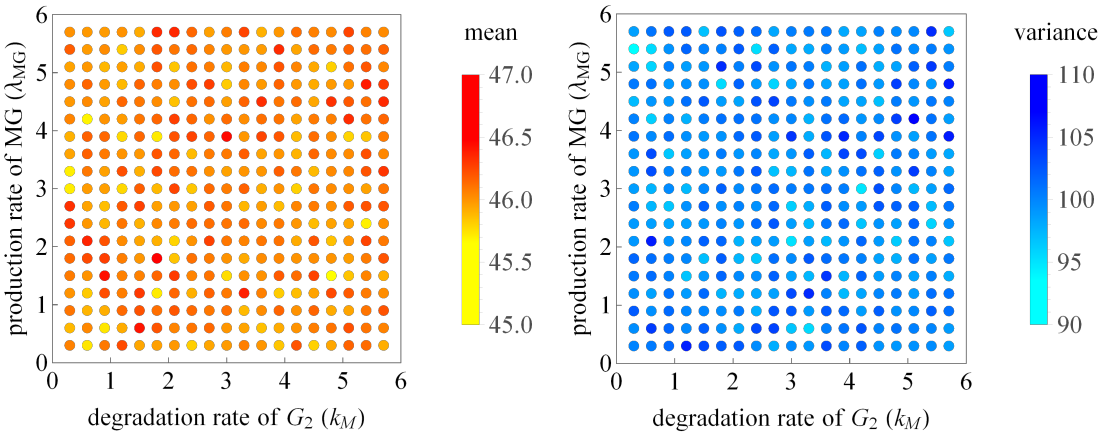
Figure 2.21: Slopes of sections of the average graph in Figure 2.20(a) parallel to...



(a) ... the λ_{MG} axis.

(b) ... the k_A axis.

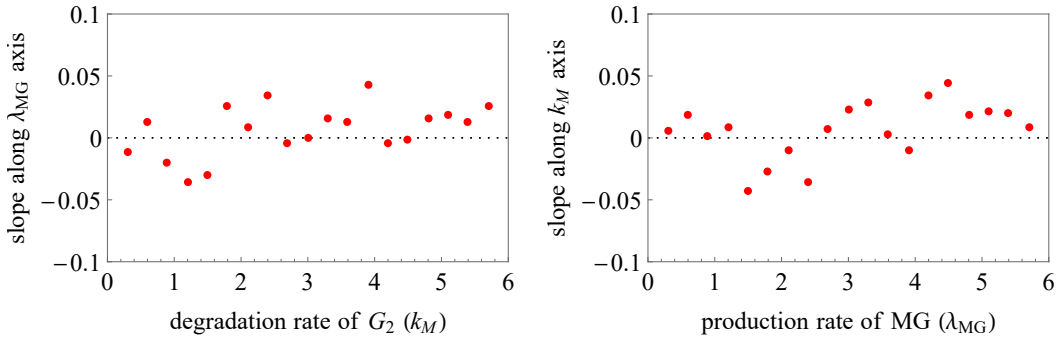
Figure 2.22: Slopes of sections of the variance graph in Figure 2.20(b) parallel to...



(a) Mean value of building blocks produced.

(b) Variance of building blocks produced.

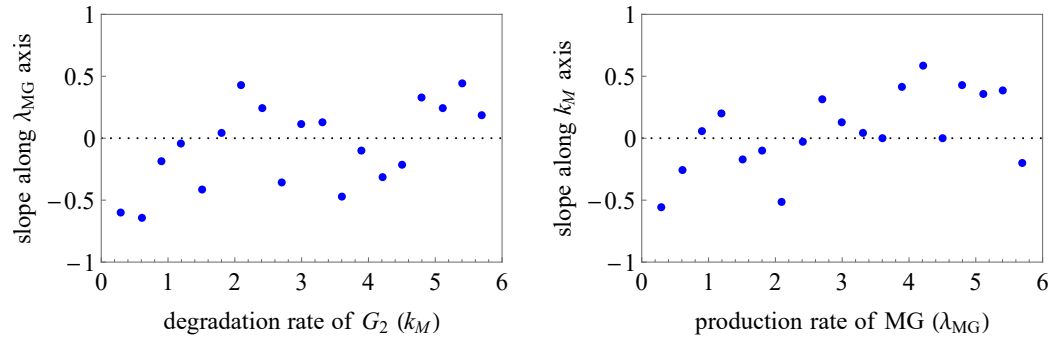
Figure 2.23: Mean value and variance of the final amount of building blocks produced as function of the degradation rate of G_2 triggered by M and the production rate of MG. All the other parameters are fixed at 0.5. We used 50 as initial number of A proteins, while the other species are initially absent.



(a) ... the λ_{MG} axis.

(b) ... the k_M axis.

Figure 2.24: Slopes of sections of the average graph in Figure 2.23(a) parallel to...



(a) ... the λ_{MG} axis.

(b) ... the k_M axis.

Figure 2.25: Slopes of sections of the variance graph in Figure 2.23(b) parallel to...

2.2.4 Role of initial amount of proteins

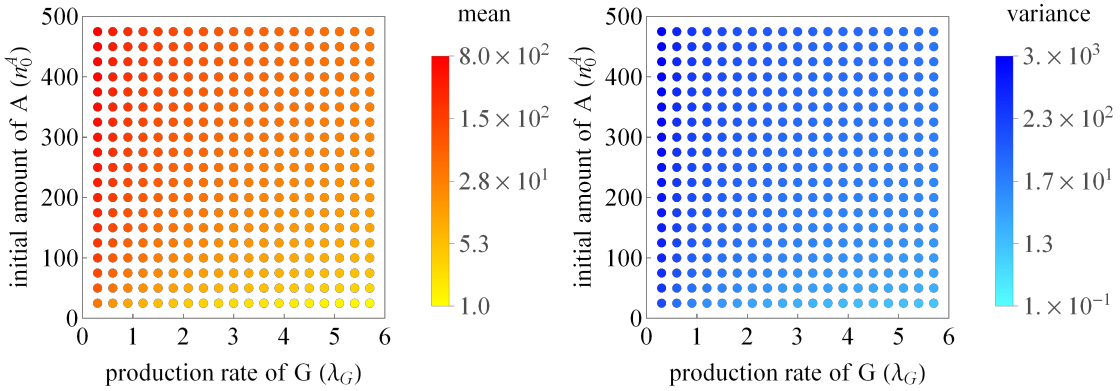
For all the simulations in the previous section we explored the behaviour of the system when changing the reaction rates. Those are not the only parameters the system is subjected to. In fact, we also fixed the amount of proteins that are initially supplied to the system, in particular, we always used 50 initial proteins A and all the other species were initially absent. Now, we take into account these initial conditions too, because protein A is not produced by the system itself. Moreover experiments suggest that protein G isn't produced by our interactome only, so we should study what happens when the initial amount of G changes.

Master regulator

Let's first consider the master regulator A. Since, according to our analysis, the production rate of G is responsible for the major dependency, we study how the building block production changes when we vary both the initial number of A at disposal, n_0^A, and the value of λ_G. In Figures 2.26 and 2.27 you can again see that an increase in λ_G leads to a stronger feedback, so a smaller amount of M-FliG is produced, but more robustly. Moreover, as expected, more master regulator yields more building blocks, but it doesn't affect so much the robustness: the Fano factor shows a much more pronounced dependency on λ_G. Something that is missing, or not visible enough, in these figures is how many building blocks are actually produced. In fact, it is true that we want the system to be robust, but that's not enough since it should be able to robustly produce the *required amount* of building blocks. With this in mind, we first control in Figure 2.28(a) what the variation (in percentage) of the average final amount of building blocks is if there's a change in n_0^A: as λ_G increases, not only stochastic fluctuations become smaller, but the dependency on n_0^A does too. Also, it could be useful to compare Figure 2.28(a) with Figure 2.28(b), which shows as lines the variation of the mean value of the produced quantity of building blocks if n_0^A varies from 25 to 500: when λ_G is large the lines are short (this is another way to express Figure 2.28(a)), but you also see that if you fix the final amount of building blocks you need, there's a large interval of values for λ_G that manage to fulfil that requirement while still acting robustly against variation of n_0^A.

Protein G

Experiments suggest that G is involved in other processes inside the bacterium, so we should check whether a variability in its amount at disposal for the system we are studying affects its robustness. For this reason we compare in Figure 2.29 the role of n_0^G and that of λ_G. Unfortunately the system seems to be very fragile in this perspective, since when there are too many proteins G the feedback mechanism is too favored and the system is not able to produce any building block.



(a) Mean value of building blocks produced.

(b) Variance of building blocks produced.

Figure 2.26: Mean value and variance of the final amount of building blocks produced as function of the production rate of G and the initial amount of A. All the other parameters are fixed at 0.5 and all the other species, but A, are initially absent. Due to the huge variation, we used a logarithmic color scale.

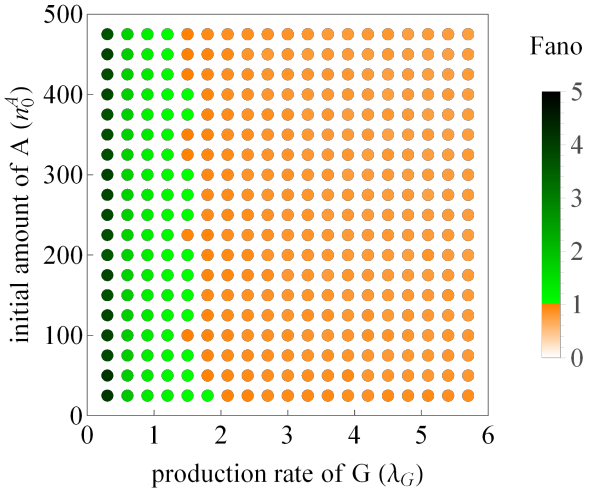
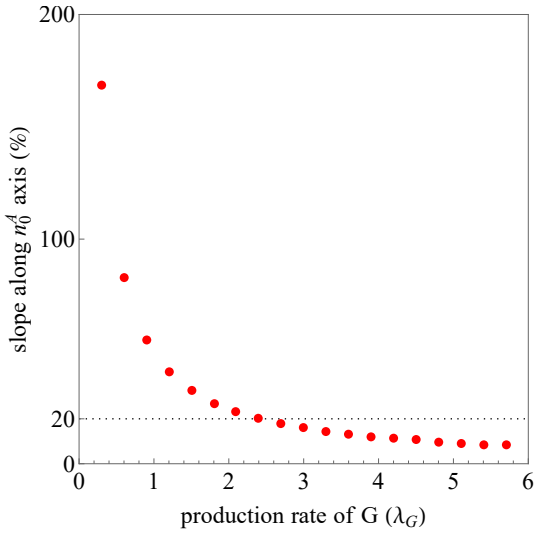
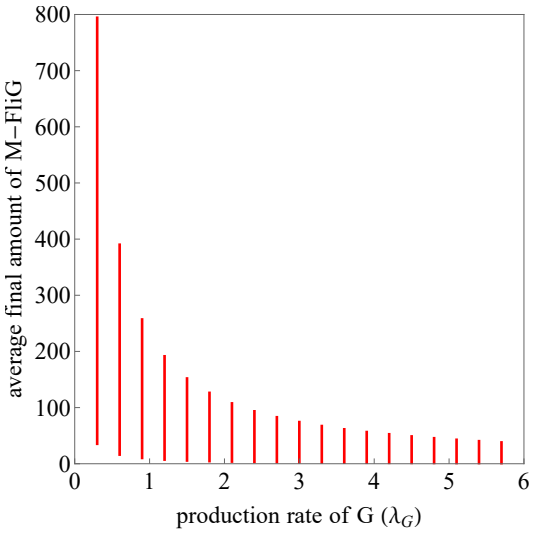


Figure 2.27: Fano factor corresponding to Figure 2.26. The different color scale highlights the region where the Fano factor is smaller than 1, that is, where the variance is smaller than the mean value.

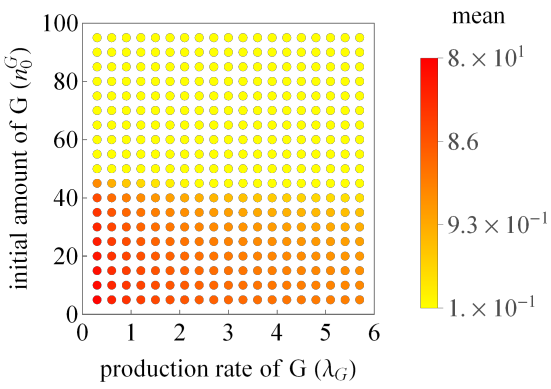


(a) This graph represents the slope of the mean in Figure 2.26 in percentage as function of the production rate of G.

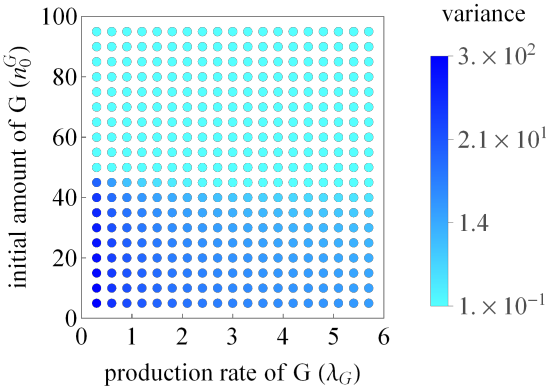


(b) Projection of Figure 2.26 on the mean-λ_G plane.

Figure 2.28: Analysis of the behaviour of the system under variations of the initial amount of A.



(a) Mean value of building blocks produced.



(b) Variance of building blocks produced.

Figure 2.29: Mean value and variance of the final amount of building blocks produced as function of the production rate and initial amount of G. All the other parameters are fixed at 0.5 and all the other species, but A and G, are initially absent. Due to the huge variation, we used a logarithmic color scale.

2.3 Equivalent system

The previous section gave us a deeper insight into how the system behaves and how it manages to produce the building blocks for the flagellar motor assembly. We also checked what conditions allow the system to behave robustly: the feedback mechanism plays a key role for robustness while there is a partial fragility when changing initial conditions for protein G. Our target now is to use this information and try to simplify the system, by reducing the number of reactions, without affecting its overall qualitative behaviour. In this way, we hope to get an intuitive but still more quantitative understanding of the system. That is, we are looking for a more effective description of the system.

2.3.1 Simplifying the interactome

We will now consider separately each step that doesn't seem to affect the overall behaviour of the system. First we analyse the subnetwork that activates A. A second candidate is the production of MG. Finally we focus on the role of G.

Activation of A

As a first attempt, we reduce the number of reactions involved in the activation of A. Consider a more straightforward process: A directly produces G, M and FliG

$$\begin{aligned}
&\mathrm{A} \xrightarrow{\lambda_G} \mathrm{A} + \mathrm{G} \\
&t_G = \lambda_G n^A, \\
&\mathrm{A} \xrightarrow{\lambda_M} \mathrm{A} + \mathrm{M} \\
&t_M = \lambda_M n^A, \\
&\mathrm{A} \xrightarrow{\lambda_F} \mathrm{A} + \mathrm{FliG} \\
&t_F = \lambda_F n^A,
\end{aligned}$$

and it is consumed by G_2 via the feedback

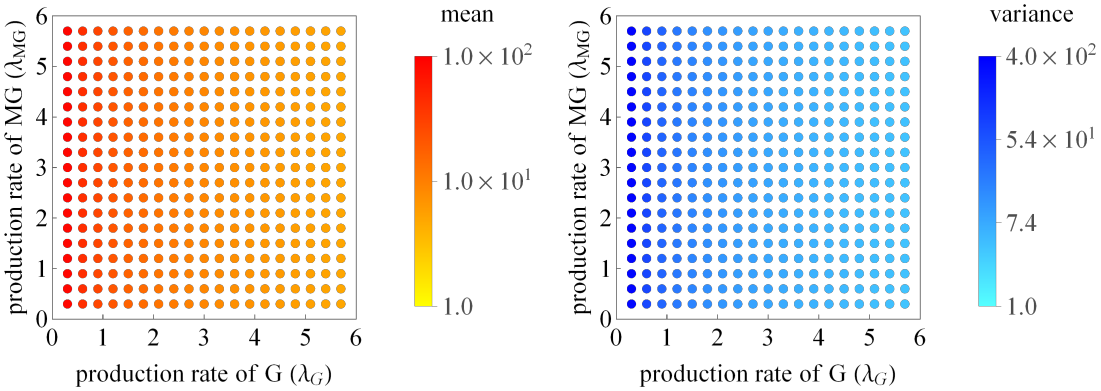
$$\begin{aligned}
&\mathrm{A} + \mathrm{G}_2 = \mathrm{GA} \\
&t_f = \lambda_f n^A n^{G_2}.
\end{aligned}$$

Essentially what we are doing is neglecting the oligomerization of A. We now want to check whether some of the features of the full system disappeared and if its qualitative behaviour changed.

Consider, for example, the role of productions of G and MG that we studied for the complete system in Figures 2.8 and 2.9, we can compare that analysis with the one of the simplified system in Figures 2.30 and 2.31. There are, of course, quantitative differences but the overall qualitative behaviour is the same: an increase of the production rate of G makes the system more robust.

Another important aspect, according to our previous analysis, is the role of initial conditions. Figures 2.32 and 2.33 show that not much has changed with respect to Figures 2.26 and 2.27: the lack of oligomerization of A is irrelevant from a robustness perspective.

A small difference is revealed by a comparison of Figure 2.34 and Figure 2.29: the absence of the gene activation process has smoothed out the dependency on the initial amount of G, but, again, qualitatively nothing changed.



(a) Mean value of building blocks produced.

(b) Variance of building blocks produced.

Figure 2.30: Mean value and variance of the final amount of building blocks produced as function of the production rate of G and MG for the system with **no oligomerization of A**. All the other parameters are fixed at 0.5. We used 25 as initial number of A proteins (because there's no dimerization, we use 25 instead of 50), while the other species are initially absent. Due to the huge variation, we used a logarithmic color scale.

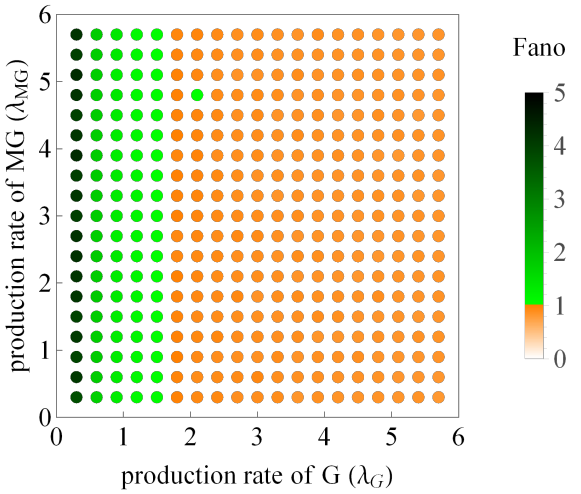
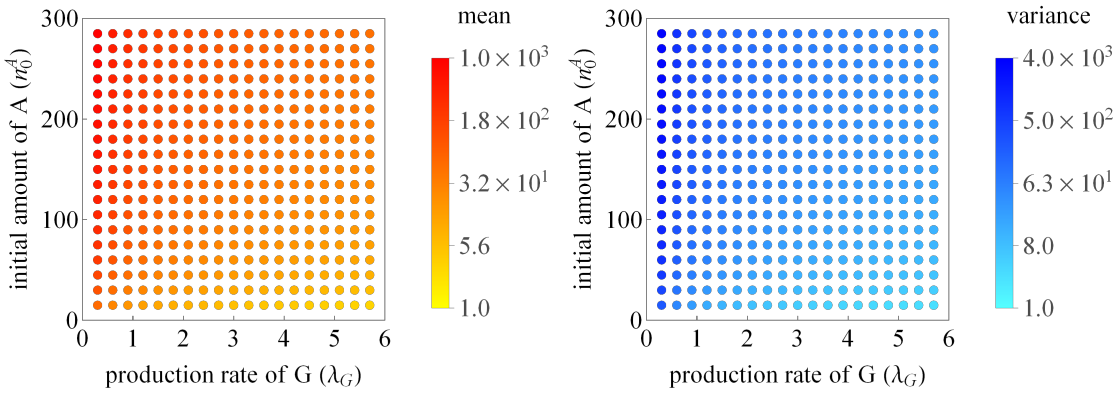


Figure 2.31: Fano factor corresponding to Figure 2.30. The different color scale highlights the region where the Fano factor is smaller than 1, that is, where the variance is smaller than the mean value.



(a) Mean value of building blocks produced.

(b) Variance of building blocks produced.

Figure 2.32: Mean value and variance of the final amount of building blocks produced as function of the production rate of G and the initial amount of proteins A for the system with **no oligomerization of A**. All the other parameters are fixed at 0.5. All the other species but A are initially absent. Due to the huge variation, we used a logarithmic color scale.

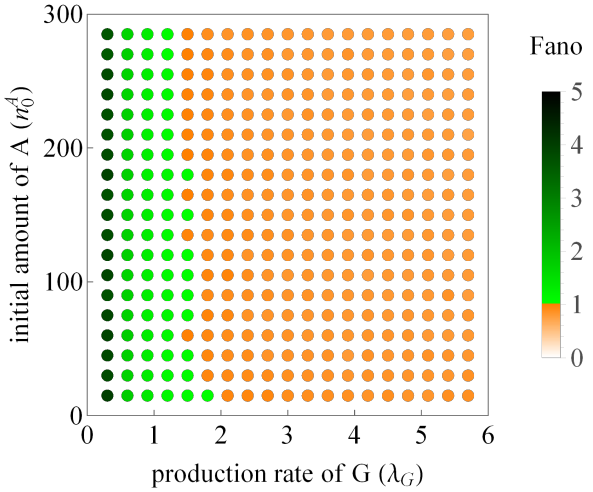
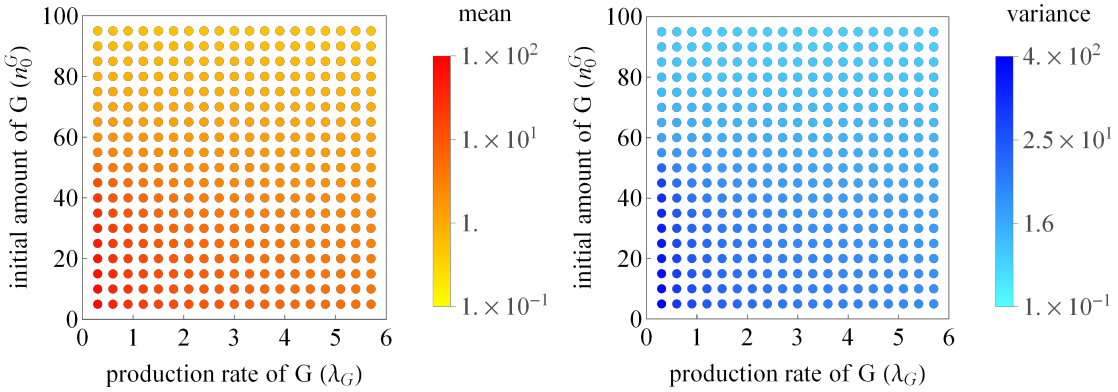


Figure 2.33: Fano factor corresponding to Figure 2.32. The different color scale highlights the region where the Fano factor is smaller than 1, that is, where the variance is smaller than the mean value.



(a) Mean value of building blocks produced.

(b) Variance of building blocks produced.

Figure 2.34: Mean value and variance of the final amount of building blocks produced as function of the production rate of G and the initial amount of proteins G for the system with **no oligomerization of A**. All the other parameters are fixed at 0.5 and all the other species but A (which is fixed at 25) are initially absent. Due to the huge variation, we used a logarithmic color scale.

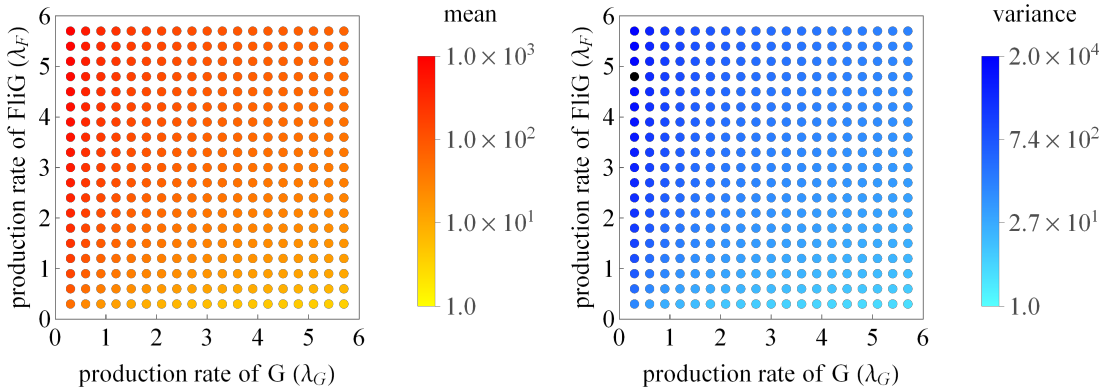
Production of MG

One more step that could be unnecessary for the qualitative description we are aiming for is the production of MG. In fact, its role for the dynamics is to counter the feedback mechanism together with production of FliG. Precisely, the proposed changes are

$$\begin{aligned}
&\mathrm{A} \xrightarrow{\lambda_G} \mathrm{A} + \mathrm{G} \\
&t_G = \lambda_G n^A, \\
&\mathrm{A} \xrightarrow{\lambda_F} \mathrm{A} + \mathrm{FliG} \\
&t_F = \lambda_F n^A, \\
&\mathrm{FliG} + \mathrm{G} \xrightarrow{\lambda_b} \text{M-FliG} + \mathrm{G} \\
&t_b = \lambda_b n^F n^G.
\end{aligned}$$

In this way, G is still involved both in the feedback mechanism (via dimerization) and in the building block production.

In Figures 2.35 and 2.36 we check if the production rate of G is still crucial for the robustness of the system. The mean value and variance of the building blocks produced scale similarly as in Figure 2.8. Notice, however, that we are comparing λ_G with λ_F, because now the production of FliG is responsible for balancing the feedback mechanism, while in the complete system that role was shared between the production of MG and FliG. Furthermore, even though a large value of λ_G corresponds to a smaller value of the Fano factor exactly like what happened in the complete system, now we are not able to reach the region where the variance is smaller than the mean value (at least in the interval of values considered).



(a) Mean value of building blocks produced.

(b) Variance of building blocks produced.

Figure 2.35: Mean value and variance of the final amount of building blocks produced as function of the production rate of G and FliG for the system with **no oligomerization of A** and **no production of MG**. All the other parameters are fixed at 0.5. All the other species but A (which is fixed at 25) are initially absent. Due to the huge variation, we used a logarithmic color scale.

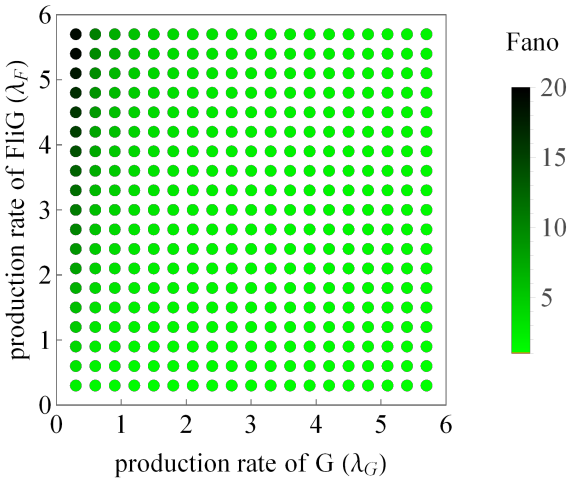
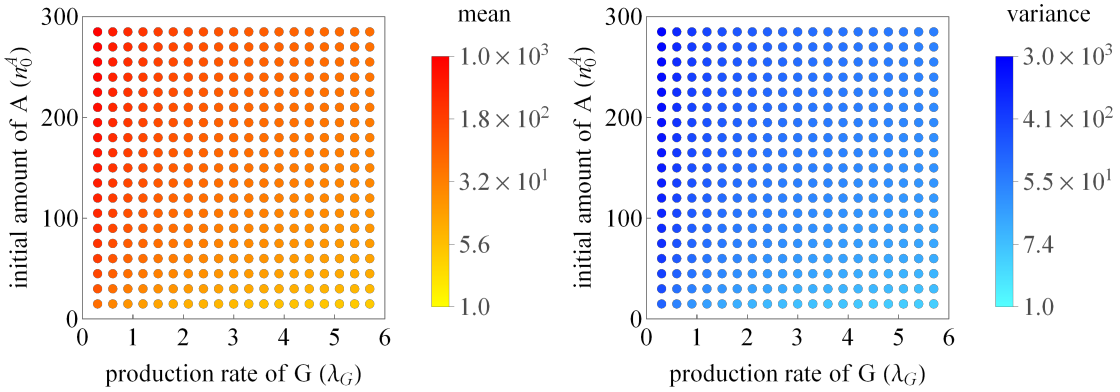


Figure 2.36: Fano factor corresponding to Figure 2.35.



(a) Mean value of building blocks produced.

(b) Variance of building blocks produced.

Figure 2.37: Mean value and variance of the final amount of building blocks produced as function of the production rate of G and the initial amount of proteins A for the system with **no oligomerization of A** and **no production of MG**. All the other parameters are fixed at 0.5. All the other species but A are initially absent. Due to the huge variation, we used a logarithmic color scale.

With regard to the initial conditions, there are no qualitative differences between Figures 2.37 and 2.38 and Figures 2.26 and 2.27. The same is true for comparing Figure 2.39, which shows the dependency on the initial amount of G for the simplified system, and Figure 2.29, which refers to the complete system.

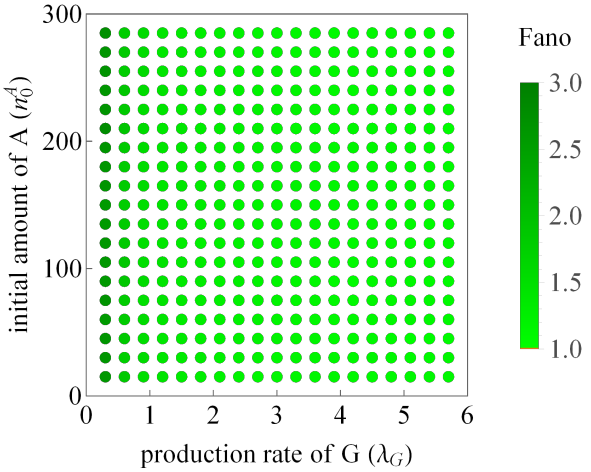
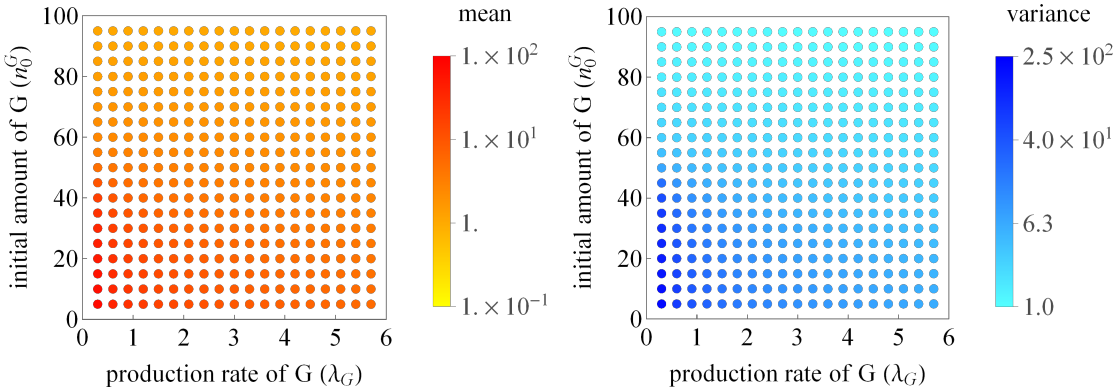


Figure 2.38: Fano factor corresponding to Figure 2.37.



(a) Mean value of building blocks produced.

(b) Variance of building blocks produced.

Figure 2.39: Mean value and variance of the final amount of building blocks produced as function of the production rate of G and the initial amount of proteins G for the system with **no oligomerization of A** and **no production of MG**. All the other parameters are fixed at 0.5. All the other species but A (which is fixed at 25) are initially absent. Due to the huge variation, we used a logarithmic color scale.

Dimerization of G

What changes to the system do actually modify its qualitative behaviour? For instance, what happens if we remove the dimerization step for G? The new feedback would be

$$\begin{aligned} &\mathrm{A} + \mathrm{G} \xrightarrow{\lambda_f} \mathrm{GA} \\ &t_f = \lambda_f n^A n^G, \end{aligned}$$

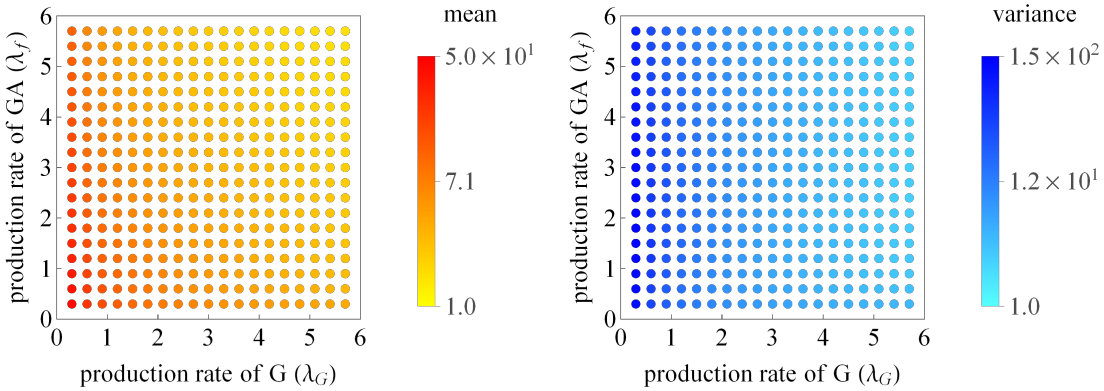
but, as you can see in Figure 2.40, the lack of the dimerization of G induces a non trivial dependency on λ_f which is not present in Figure 2.14. The reason is that, with this modification, the parameter λ_f is the only one to rule the feedback, while in the complete system that task was split between λ_f and λ_{GG}, which is now missing.

Degradation of G_2

One more reaction whose role we should check is the degradation of the dimers of G due to protein A

$$\mathrm{G}_2 + \mathrm{A} \xrightarrow{k_A} 2\mathrm{G} + \mathrm{A}.$$

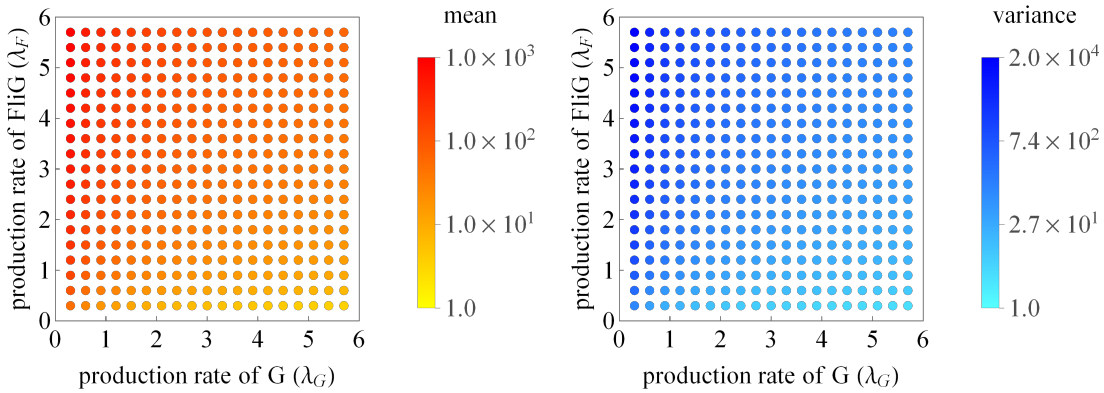
This means that there is no degradation of G_2 anymore, since we already removed protein M. Figures 2.41 to 2.45 show that nothing changes compared to the previous case.



(a) Mean value of building blocks produced.

(b) Variance of building blocks produced.

Figure 2.40: Mean value and variance of the final amount of building blocks produced as function of the production rate of G and GA for the system with **no oligomerization of A**, **no production of MG** and **no dimerization of G**. All the other parameters are fixed at 0.5. All the other species but A (which is fixed at 25) are initially absent. Due to the huge variation, we used a logarithmic color scale.



(a) Mean value of building blocks produced.

(b) Variance of building blocks produced.

Figure 2.41: Mean value and variance of the final amount of building blocks produced as function of the production rate of G and FliG for the system with **no oligomerization of A**, **no production of MG** and **no degradation of G_2**. All the other parameters are fixed at 0.5. All the other species but A (which is fixed at 25) are initially absent. Due to the huge variation, we used a logarithmic color scale.

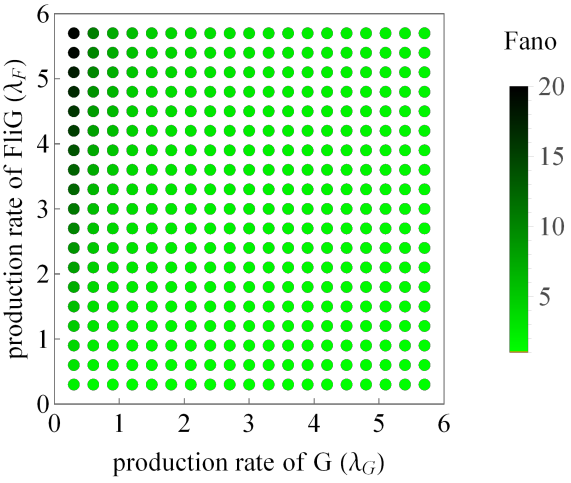
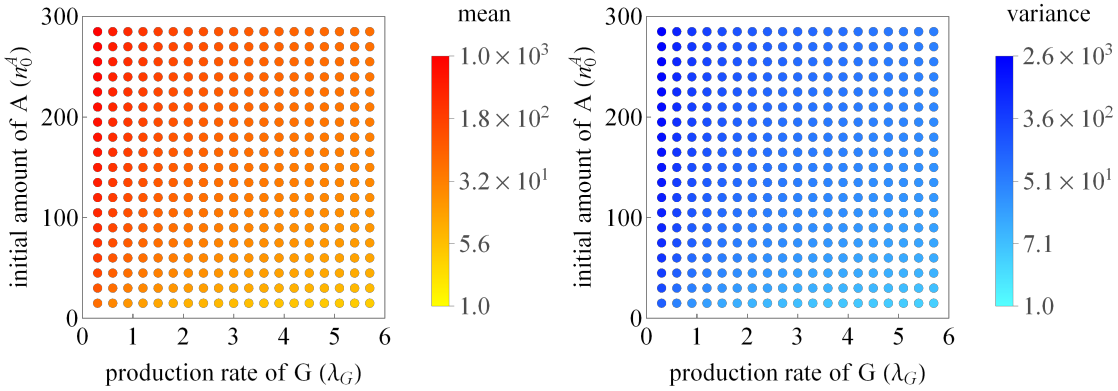


Figure 2.42: Fano factor corresponding to Figure 2.41.



(a) Mean value of building blocks produced.

(b) Variance of building blocks produced.

Figure 2.43: Mean value and variance of the final amount of building blocks produced as function of the production rate of G and the initial amount of proteins A for the system with **no oligomerization of A**, **no production of MG** and **no degradation of G_2**. All the other parameters are fixed at 0.5. All the other species but A are initially absent. Due to the huge variation, we used a logarithmic color scale.

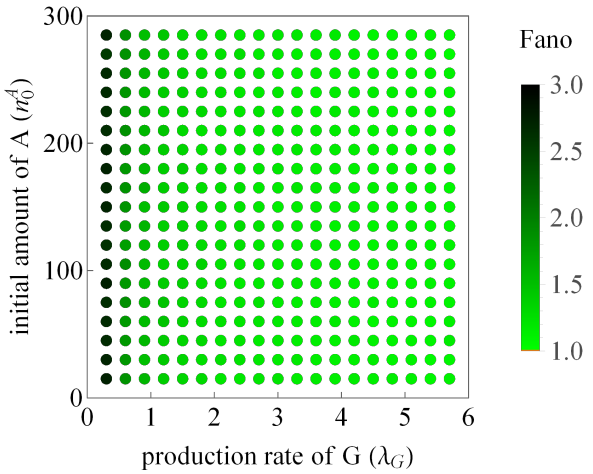
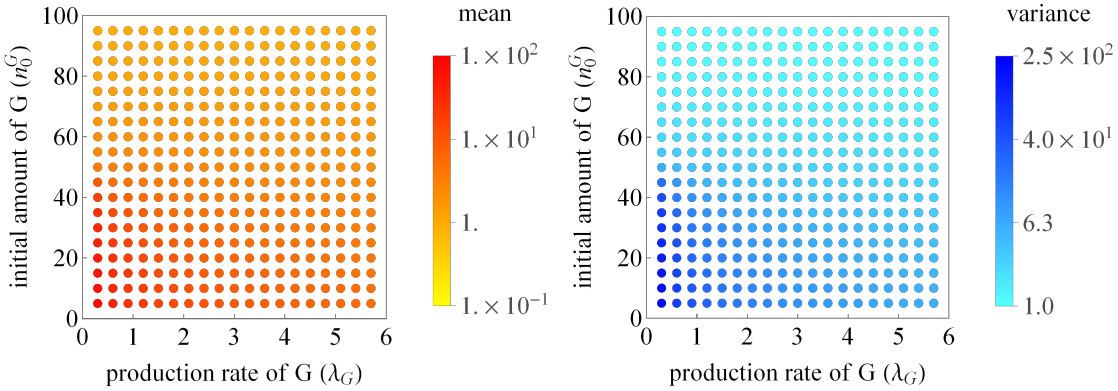


Figure 2.44: Fano factor corresponding to Figure 2.43.



(a) Mean value of building blocks produced.

(b) Variance of building blocks produced.

Figure 2.45: Mean value and variance of the final amount of building blocks produced as function of the production rate of G and the initial amount of proteins G for the system with **no oligomerization of A**, **no production of MG** and **no degradation of G_2**. All the other parameters are fixed at 0.5. All the other species but A (which is fixed at 25) are initially absent. Due to the huge variation, we used a logarithmic color scale.

2.4 Analytical study

In Section 2.2.2 we tried to exploit the theory developed in Section 1.4 with no success, because of the high complexity of the mathematical description of the system. The previous section gave us a simplified version of the original system, with the same qualitative features, but much less variables. So we try to use the machinery of Section 1.4 again and, maybe, to make use of the insight we gained through the simulations.

The simplified version of the system is described by the following set of reactions

$$\begin{aligned}
&\mathrm{A} \xrightarrow{\lambda_G} \mathrm{A} + \mathrm{G} && t_G = \lambda_G n^A \\
&\mathrm{A} \xrightarrow{\lambda_F} \mathrm{A} + \mathrm{FliG} && t_F = \lambda_F n^A \\
&2\mathrm{G} \xrightarrow{\lambda_{GG}} \mathrm{G}_2 && t_{GG} = \lambda_{GG} n^G (n^G - 1) \\
&\mathrm{A} + \mathrm{G}_2 \xrightarrow{\lambda_f} \mathrm{AG} && t_f = \lambda_f n^{G_2} n^A \\
&\mathrm{FliG} + \mathrm{G} \xrightarrow{\lambda_b} \text{M-FliG} + \mathrm{G} && t_b = \lambda_b n^G n^F,
\end{aligned}$$

for which we also computed the transition probabilities according to the combinatorial kinetics. Now, we can explicitly write the master equation (1.10) for the probability density function $f(n^A, n^G, n^F, n^{G_2}, n^{GA}, n^b, t)$, but, since the list of arguments is quite long, we will use a more compact notation by writing explicitly only those variables that change value, for example $f(n^G - 1) \equiv f(n^A, n^G - 1, n^F, n^{G_2}, n^{GA}, n^b, t)$. With this notation, the master equation is

$$\begin{aligned}
\partial_t f =& \lambda_G n^A f(n^G - 1) + \lambda_F n^A f(n^F - 1) + \\
&+ \lambda_{GG}(n^G + 1)(n^G + 2) f(n^G + 2, n^{G_2} - 1) + \\
&+ \lambda_f (n^A + 1)(n^{G_2} + 1) f(n^A + 1, n^{G_2} + 1, n^{GA} - 1) + \\
&+ \lambda_b n^G (n^F + 1) f(n^F + 1, n^b - 1) + \\
&- \Big(\lambda_G n^A + \lambda_F n^A + \lambda_{GG} n^G (n^G - 1) + \lambda_f n^A n^{G_2} + \\
&\qquad + \lambda_b n^G n^F \Big) f(n^A, n^G, n^F, n^{G_2}, n^{GA}, n^b, t).
\end{aligned} \tag{2.1}$$

The moment-generating function for f is

$$\begin{aligned}
G(\alpha_A, \alpha_G, \alpha_F, \alpha_{G_2}, \alpha_{GA}, \alpha_b, t) = \sum_{n=0}^{+\infty} & \alpha_A^{n^A} \alpha_G^{n^G} \alpha_F^{n^F} \alpha_{G_2}^{n^{G_2}} \alpha_{GA}^{n^{GA}} \alpha_b^{n^b} \times \\
& \times f(n^A, n^G, n^F, n^{G_2}, n^{GA}, n^b, t),
\end{aligned}$$

where the sum over n must be intended as it is done over n^A, n^G, etc.

Using equation (2.1) and taking the derivative of G with respect to time, we get an equation for the moment-generating function, which can be rewritten in the form of a PDE. In particular, it includes terms like

$$\sum_{n=0}^{+\infty} n^A \alpha_A^{n^A} \alpha_G^{n^G} \alpha_F^{n^F} \alpha_{G_2}^{n^{G_2}} \alpha_{GA}^{n^{GA}} \alpha_b^{n^b} f(n^A, n^G, n^F, n^{G_2}, n^{GA}, n^b, t) = \alpha_A \partial_{\alpha_A} G,$$

or

$$\begin{aligned}
&\sum_{n=0}^{+\infty}\sum_{n^G=1}^{+\infty} n^A(\alpha_A)^{n^A}(\alpha_G)^{n^G}(\alpha_F)^{n^F}(\alpha_{G_2})^{n^{G_2}}(\alpha_{GA})^{n^{GA}}(\alpha_b)^{n^b}\times \\
&\qquad\qquad \times f(n^A, n^G-1, n^F, n^{G_2}, n^{GA}, n^b, t) = \\
&= \sum_{n=0}^{+\infty}\sum_{n^G=0}^{+\infty} n^A(\alpha_A)^{n^A}(\alpha_G)^{n^G+1}(\alpha_F)^{n^F}(\alpha_{G_2})^{n^{G_2}}(\alpha_{GA})^{n^{GA}}(\alpha_b)^{n^b}\times \\
&\qquad\qquad \times f(n^A, n^G, n^F, n^{G_2}, n^{GA}, n^b, t) = \\
&= \alpha_A\alpha_G\partial_{\alpha_A}G.
\end{aligned}$$

Following similar steps, essentially redefining dumb indices, we transform each term into an appropriate derivative of G, and we arrive at the following second order PDE, of the form (1.11)

$$\begin{aligned}
\partial_t G =&\alpha_A(\lambda_G(\alpha_G-1)+\lambda_F(\alpha_F-1))\partial_{\alpha_A}G+ \\
&+\lambda_{GG}(\alpha_{G_2}-\alpha_G^2)\partial^2_{\alpha_G}G+ \\
&+(\lambda_f(\alpha_{GA}-\alpha_A\alpha_{G_2})+\lambda_b\alpha_G(\alpha_b-\alpha_F))\partial_{\alpha_G}\partial_{\alpha_F}G,
\end{aligned} \tag{2.2}$$

which is called Kolmogorov's equation. Unfortunately, we weren't able to solve it exactly. Instead, we try the method explained in Section 1.4.2: by taking the derivative of (2.2) with respect to each α and then setting all the others to 1, we get a set of ODE's for mean values

$$\begin{aligned}
&\partial_t\left\langle n^A\right\rangle = -\lambda_f\left\langle n^A n^{G_2}\right\rangle \\
&\partial_t\left\langle n^G\right\rangle = \lambda_G\left\langle n^A\right\rangle - 2\lambda_{GG}\left(\left\langle (n^G)^2\right\rangle - \left\langle n^G\right\rangle\right) \\
&\partial_t\left\langle n^{G_2}\right\rangle = \lambda_{GG}\left(\left\langle (n^G)^2\right\rangle - \left\langle n^G\right\rangle\right) - \lambda_f\left\langle n^A n^{G_2}\right\rangle \\
&\partial_t\left\langle n^{GA}\right\rangle = \lambda_f\left\langle n^A n^{G_2}\right\rangle \\
&\partial_t\left\langle n^F\right\rangle = \lambda_F\left\langle n^A\right\rangle - \lambda_b\left\langle n^G n^F\right\rangle \\
&\partial_t\left\langle n^b\right\rangle = \lambda_b\left\langle n^G n^F\right\rangle.
\end{aligned} \tag{2.3}$$

The above system is not closed, since there appear also higher-order terms such as $\left\langle (n^G)^2\right\rangle$, $\left\langle n^A n^{G_2}\right\rangle$ and $\left\langle n^G n^F\right\rangle$. As a result, to solve it, we would have to compute the equations of motion for these mean values. Doing this, we would always find a new higher-order term. In fact, consider for instance the ODE for $\left\langle n^A\right\rangle$, which we found by deriving (2.2) with respect to α_A and then setting all the α's to 1. In particular, focus just on the following term

$$\partial_t\partial_{\alpha_A}G = \cdots + \lambda_f\alpha_{G_2}\partial_{\alpha_A}\partial_{\alpha_{G_2}}G + \cdots.$$

If we want to find the ODE for $\left\langle n^A n^{G_2}\right\rangle$, we need to take the derivative of the previous equation again with respect to α_{G_2} which leads to the term (among all the others)

$$\partial_t\partial_{\alpha_A}\partial_{\alpha_{G_2}}G = \cdots + \lambda_f\alpha_{G_2}\partial_{\alpha_A}\partial^2_{\alpha_{G_2}}G + \cdots,$$

which contains $\left\langle n^A(n_2^G)^2\right\rangle$. So, every time we try to close the equation for $\left\langle n^A\right\rangle$ a new term appears. A similar thing happens for the other variables.

An aspect worth noticing is that this kind of analysis already fails even if we consider the feedback loop only. In fact, if we try to isolate the feedback, which means that we take λ_F and λ_b to be zero, we could in principle be able to extract an effective production rate of G, of the form $\lambda_G(\lambda_{GG}, \lambda_f, n_0^A, n_0^G)$, that could further simplify the analysis. However, the above argument for solving the equations of motion for mean values fails for the feedback too. As a result, this strategy doesn't work.

The cheapest way to make the system solvable is by using the mean-field approximation, writing $\left\langle n^A n^{G_2} \right\rangle \simeq \left\langle n^A \right\rangle \left\langle n^{G_2} \right\rangle$ and $\left\langle n^G n^F \right\rangle \simeq \left\langle n^G \right\rangle \left\langle n^F \right\rangle$. Suppose we apply this approximation for different variables only, that is, we neglect the covariances, but not the variances, e.g. $\left\langle (n^b)^2 \right\rangle \neq (\left\langle n^b \right\rangle)^2$. In this way, using a similar procedure, we get an ODE for the variance of the building blocks

$$\partial_t \partial^2_{\alpha_b} G|_{\alpha=1} = \partial_t \left\langle (n^b)^2 \right\rangle - \partial_t \left\langle n^b \right\rangle = 2\lambda_b \left\langle n^b n^G n^F \right\rangle \simeq 2\lambda_b \left\langle n^b \right\rangle \left\langle n^G \right\rangle \left\langle n^F \right\rangle$$

$$\partial_t \mathrm{Var}[n^b] = \partial_t \left(\left\langle (n^b)^2 \right\rangle - \left\langle n^b \right\rangle^2 \right) = \partial_t \partial^2_{\alpha_b} G|_{\alpha=1} + \partial_t \left\langle n^b \right\rangle - \partial_t \left(\left\langle n^b \right\rangle^2 \right).$$

We can expand the last term and make use of the equation of motion for n^b in the mean-field approximation to get

$$\partial_t \left(\left\langle n^b \right\rangle^2 \right) = 2 \left\langle n^b \right\rangle \partial_t \left\langle n^b \right\rangle = 2\lambda_b \left\langle n^b \right\rangle \left\langle n^G \right\rangle \left\langle n^F \right\rangle,$$

hence

$$\partial_t \mathrm{Var}[n^b] = \partial_t \left\langle n^b \right\rangle.$$

The approximation results in the fact that the average and the variance trajectories of the number of building blocks differ just for a constant (fixed by initial conditions), so all the stochastic effects we saw in our simulations on the robustness of the system are missing. For this reason, an analysis of the deterministic trajectories of the system would be useless for our purposes.

A proof that the mean-field approximation cannot be used is given by computing numerically the covariance between different variables at each point in time. If it is negligible with respect to the mean value trajectories in the region of parameters space which produces robustness, then the approximation could be used. In order to build an average trajectory, a variance trajectory or a covariance trajectory for the amount of protein species we use the following procedure:

- first use the Gillespie's algorithm to produce N different instances of trajectories of the system,
- then divide the time domain T, common to all the trajectories, in small intervals, defined by dividing T by the maximum number of iterations of the algorithm (each run of the algorithm will reach the end of the process at a different time, so we consider T to be the maximum value),
- finally compute the average, variance or covariance among the values that fall in the same time interval.

For example, in the set of equations for mean values (2.3) the covariance between n^A and n^{G_2} appears often. We, thus, compute its trajectory for different values of the production rate of G, λ_G. Figure 2.46 shows that the covariance becomes stronger when the system is more robust, so the closer we are to the robustness region in parameters space, the less accurate mean-field approximation is.

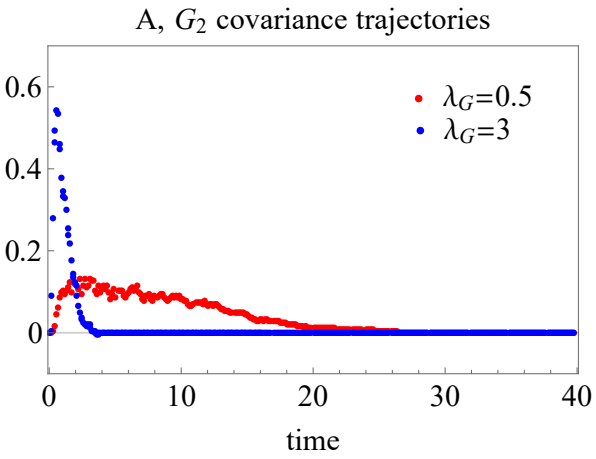


Figure 2.46: Covariances trajectories for the simplified version of the system with all the rates, but λ_G, set to 0.5, and all proteins initially absent except for A, which is set to 25 as usual.

Chapter 3

Conclusions

Our work explored the possibility that the bacterium *Shewanella putrefaciens* deploys a counting mechanism to control the number of flagellar motor assemblies. As explained earlier, in order to be called counting mechanism, the network of chemical reactions must robustly produce the right amount of building blocks. This is enough because its shape and, in particular, the number of subunits in the target structure are fixed. In our analysis we haven't included a study on the time scales involved in the process, so we ignored the assembling itself (and related aspects, such as the efficiency of the construction) and all the degradations that in biology always exist. The theory developed in Chapter 1 allowed us to describe the relevant quantities with stochastic variables and portray the system as a stochastic process. Gillespie's algorithm was used to perform numerical simulations in order to gain better insight into the role of each reaction.

This strategy highlighted the function of the feedback mechanism and, in particular the reaction that produces the protein G. If it is favoured with respect to opposing reactions (that is the production of FliG and M), then the consumption of the master regulator A is preferred and the growth of the building block population is constrained. As a result, the production of G reduces both the stochastic fluctuations and the dependency on the initial amount of A of the final quantity of blocks produced. Unfortunately, the system doesn't perform similarly well against changes in the amount of proteins G at disposal. In fact, it seems that G is also produced by other means, and there could be other controls that interfere.

Our target was to make use of the understanding obtained through all the simulations in order to generalise the crucial role of the feedback mechanism, both trying an analytical approach and comparing our particular system with others. In Chapter 2 we listed some of the attempts we made to analytically find the distribution of the final amount of building blocks. Essentially, we first tried to simplify the system, reducing the number of reactions, while keeping unchanged its qualitative features. Unfortunately, this level of simplification was not enough to get a treatable partial differential equation out of the master equation. In fact, it turns out that it's the feedback itself, the key feature of the system, that introduces non-linearities to the problem.

When we tried to break those non-linearities, by using a simple mean field approximation, it was soon clear that this strategy was too harsh: the dynamics becomes that of the deterministic system, with all its problems. In fact, a check of the covariances showed that it is impossible to neglect some of them. One last try was to search for an explicit parametrisation of the covariances, in order to weaken a bit the mean field approximation. This possibility, though, failed due to the large number of parameters.

Something that could still be done is studying the chemical pathway of other self-assembling

structures in order to classify all the possible ways to achieve robustness.

Furthermore, an analytical approach would be useful since it uncovers what different mechanisms have in common. In particular, if in some way it will be possible to find a model that explicitly describes the probability distribution of the final amount of building blocks, with the right dependencies on all the parameters of the system, then one could in principle use the framework provided by the sloppy model theory (reviewed in [9]) in order to systematically perform the kind of analysis we did numerically, that is finding what are the relevant reactions in a robustness perspective.

It would also be interesting to further develop our analysis by weakening some of the assumptions we made. In particular, time scales play an important role, as mentioned before, for the degradation of proteins and the assembling process, but also for the behaviour of robustness during the splitting of the cell.

Bibliography

[1] Florian Altegoer et al. “From molecular evolution to biobricks and synthetic modules: a lesson by the bacterial flagellum”. In: *Biotechnology and Genetic Engineering Reviews* 30.1 (2014).

[2] Gert Bange, Jan S. Schuhmacher, and Kai M. Thormann. “How bacteria maintain location and number of flagella?” In: *FEMS Microbiology Reviews* (2015).

[3] Gardiner Crispin. *Handbook of stochastic methods for physics, chemistry, and the natural sciences*. Springer-Verlag, 2004.

[4] Isabella R. Graf et al. “Stochastic Yield Catastrophes and Robustness in Self-Assembly”. In: *arXiv e-prints* (2019).

[5] Tijms Henk C. *A first course in stochastic models*. Wiley, 2003.

[6] A.S. Hennis and C.W. Birky. “Stochastic partitioning of chloroplasts at cell division in the alga *Olisthodiscus*, and compensating control of chloroplast replication”. In: *Journal of Cell Science* 70.1 (1984).

[7] Wallace F. Marshall. “Cell Geometry: How Cells Count and Measure Size”. In: *Annual Review of Biophysics* 45.1 (2016).

[8] Ross Sheldon M. *Stochastic processes*. Wiley, 1995.

[9] Mark K. Transtrum et al. “Perspective: Sloppiness and emergent theories in physics, biology, and beyond”. In: *The Journal of Chemical Physics* 143 (2015).