The Microsoft Research - University of Trento
Centre for Computational
and Systems Biology

Technical Report CoSBi 01/2008

Modeling and parameter estimation of the SOS response network in E.coli

Angela Baralla, Matteo Cavaliere

Microsoft Research - University of Trento, Centre for Computational and Systems Biology, CoSBi, Trento, Italy

angela.b3@virgilio.it, cavaliere@cosbi.eu

Alberto de la Fuente

CRS4 - Bioinformatics Laboratory Sardegna Ricerche

alf@csr4.it

UNIVERSITÀ DEGLI STUDI DI TRENTO

The Microsoft Research - University of Trento
Centre for Computational and Systems Biology

Second Level International Master in
Computational and Systems Biology

Master Thesis

Modeling and parameter estimation of the SOS response network in *Escherichia coli*

Advisor
Alberto de la Fuente

Student
Angela Baralla

Co-advisor
Matteo Cavaliere

Academic year 2006/2007

Contents

Introduction

The SOS response is an inducible DNA repair system that allows bacteria to survive in presence of an increased level of DNA damage. More than 40 genes are induced in response to DNA damage as part of the SOS regulon in Escherichia coli. Two main proteins play a key role in the regulation of this response: a repressor LexA that prevents the expression of these genes and an inducer RecA that induces the LexA cleavage reaction and the subsequent expression of the response genes. Most of these response genes are responsible for error-free DNA-damage repair and for the regulation of cell division. In this thesis we have investigated a network of nine genes including the principal mediators of the SOS response: [1] *lexA, recA, ssB, recF, dinI and umuDC* and three sigma factor genes: *rpoD, rpoH and rpoS.*

Starting from this well studied gene network we have employed four different mathematical models that describe the dynamics of this network. The four models are based on four distincts "principles" :

- mass action kinetics;
- Michaelis Menten-type kinetics;
- "Mendes" kinetics;
- S-systems.

One of the major problems in modeling gene networks is the estimation of the kinetic parameters. In this thesis we present several methods to estimate the parameter values of the constructed models. In particular we have investigated which methods provide the parameters that fits better the experimental data obtained from the M3D database [3] as time series data concerning samples from cultures under norfloxacin perturbation.

For this purpose we have used the software COPASI [2] and we have investigated all the different algorithms of parameter fitting available in the software. Specifically, we have studied global optimization algorithms such as Particle Swarm, Genetic Algorithms, and Evolutionary Programming and local optimization algorithms like Hookes and Jeeves. Of the four models the last two were the ones that fit better the time course experimental data. In particular the best results were obtained by running Particle Swarm followed by Hookes and Jeeves.

Once obtained the parameters of the model, we have studied how the model could predict the dynamics of the network under specific perturbations. Specifically, we have tested the models by using experimental data from four gene over-expressions under low antibiotic. In this way we have found out that the model using Mendes kinetics could predict the steady state values for some of the genes in the wild-type experiment, but not in the over-expression experiments, while the S-model was not able to predict such steady states.

The thesis is organized in the following way: in the first chapter we present the biological aspects of the SOS response gene network and the four models above mentioned explaining the kinetic functions used and present the mathematical equations derived for each of the genes involved in the network.

In the second chapter we describe the algorithms used to estimate the parameters focusing on the description of the two algorithms that are the ones that perform better with the used experimental data.

In the third chapter we present in more details the tools used, in particular the software COPASI and the database from which the experimental data have been obtained.

In the fifth chapter we investigate the influence of the parameter values on the dynamics of the models and on the way they fit the experimental data.

In the final chapter we summarize the obtained results, and we underline some open problems.

Chapter 1

Modelling the SOS gene network

In this chapter we will present the biological phenomen of the SOS response in *E. coli.* This phenomen involves a lot of genes, but in particular we will focus on the principal mediators of it. Once described the principal actors of this network we will present four ODEs models.

1.1 The SOS response in *Escherichia coli*

Different sources such as radiation, chemical mutagens and products of metabolism induce damage to the genomes of organisms from bacteria to man. DNA damage can be fatal for the organism since they halt the DNA replication machinery preventing the duplication of genetic information and thus cell division [8]. During evolution bacteria have evolved different mechanisms to repair or bypass this damage .

In *Escherichia coli* this repair mechanism is called 'SOS response' a term coined by Miroslav Radman in 1974 to describe cellular response after exposure on ultraviolet radiation. The network involves nearly 40 genes directly regulated by *lexA* and *recA*, and tens or possibly hundreds of indirectly regulated genes [1]. Under normal conditions the cell does not need to express these repair genes so that they are kept silent until a DNA damage occurs.

Two main proteins play key role in the regulation of this response: a repressor LexA that prevents the expression of these genes and an inducer RecA that induces the LexA cleavage reaction and the subsequent expression of the response genes. The functioning of the network is the following one:

The repressor protein LexA binds to operators that contain a 20-bp[1] consensus sequence with different affinities so that genes with low affinities for LexA are induced early. Ultraviolet radiation can create on the DNA dimerization of neighbouring pyrimidines. Upon encountering such a lesion the replication machinery halts leaving single stranded DNA free to be attacked; to avoid this the DNA is covered with the protein RecA which polymerizes with the DNA becoming active RecA*.

[1] bp: base pairs

Active RecA* nucleoprotein filaments are formed in the presence of ATP by cooperative binding of 38-kDa[2] RecA monomers to single-stranded DNA, assembling in a $5' \to 3'$ direction. Filament disassembly also proceeds $5' \to 3'$ but requires ATP hydrolyses. A single helical turn covers 18 bases consisting of six RecA monomers, with each monomer binding three nucleotides [9]. The activaction of RecA permits the cleavage of LexA and so the induction of the response genes.

The first genes to be induced are the *uvr* genes which encode for protein that carry out the NER (nucleotide excise repair) reaction that excises the damaged nucleotides from double stranded DNA. About 85% of lesions are repaired in this way but some more intensive lesions are not repaired by this mechanism, so a second one is required: recombination, which is carried out by RecA. Homologous recombination allows the repair of lesions that occur on ssDNA[3] regions at replication forks by rendering them double-stranded and a substrate for NER.

When both NER and recombination fail in repairing DNA lesions, replication is restarted by using a special DNA poliymerase DNApol V which is the product of processed and assembled products of the gene *umuDC.* DNA pol V is composed of the 46-kDa UmuC subunit, which contains the active site of the polymerase, and two 12-kDa UmuD' proteins. Before association of the heterotrimer, UmuD undergoes a LexA-like cleavage reaction mediated by RecA* [10] [11]. Only the N-terminal truncated UmuD' is mutagenically active and it stabilizes UmuC by forming $UmuD'_2C$.

Figure 1.1 describes in a very general way the interactions above described.

The price paid for using this DNApol is mutagenesis, infact this polymerase is a member of the Y-family polymerases, which lack a $3' \to 5'$ exonuclease proofreading[4] activity and $5' \to 3'$ nick translation activity, so DNApol V can make substitution errors with a frequency of $10^{-2} - 10^{-3}$ on undamaged DNA.

After succesfull repair of DNA damage, the LexA concentration must be restored in order to shut off the SOS response genes that are not useful anymore. There are different proteins that seems to play this role, in particular these proteins interfer with the activaction and inactivation of the SOS response by moduling the stability of RecA.

The assembly-disassembly regulation of RecA nucleoprotein filaments mantains a balance between RecA activaction for recombination, SOS induction and mutagenesis [9].

Recent single-cell experiments measured the temporal dependence of the activity of LexA-regulated promoters [12] for different UV doses: at low UV doses, the promoter activity at about 10 minutes after the UV radiation showed a single peak and this was also observed in measurements of promoter activity averaged over a large population of cells [4] and can be attributed to the initial rapid drop in LexA levels after UV damage because of the activation of RecA. After the radiation with higher doses, LexA-regulated promoter activity often had a second peak at about 30 - 40 min, sometimes even followed by a third peak at 60 - 90 min.

So, in each cell, the SOS response is not simply turned on to an extent

[2] kDa: kiloDalton

[3] ssDNA: single strand DNA

[4] proofreading: excision of a misincorporated nucleotide at a growing 3'-primer end by a 3'-exonuclease associated with the polymerase.

that depends on the level of DNA damage and then turned off but it acts in a modular manner. Thus there is a highly precise timing device as part of the SOS genetic network which is indipendent of the cell cycle. Walker and coworkers with their experiments on mutants suggested that the genes *umuC* and *umuD* might have a key role in the regulation of these modulations [9].

1.2 The Gene Network formalism

Functional interactions between genes can be represented through the use of gene networks. These networks are depicted as diagrams in which the nodes represent genes or their activities (mRNAs) and the edges represent the regulatory interactions between them.

Such interactions are not physical interactions but just causal. In fact, in reality gene expression is not directly influenced by a gene, there is not physical interaction between them, but usually are the gene products (proteins or metabolites) that mediate this reciprocal influence.

So, regulation not only involves the level of mRNAs but also the level of protein and the level of intermediary metabolism. This hierarchy can be visualized as in Figure 1.2 in which we can see that the three levels of interactions are represented as planes and the interactions between the genes are only indirect interactions (represented as dotted lines) [13].

The use of gene networks instead of complete biochemical models is an abstraction that can be useful when we want to study the functional dynamics of genes; abstracting at a gene level can make modeling and analysing more feasible.

In fact, not only gene networks are capable of describing a large number of interactions in a simple way, but they may also be very useful in representing the regulatory properties accompanying those interactions at a systems level.

Knowing the structure of gene networks and performing simulations of their behavior on computers will increase our fundamental understanding of living systems [13]. Knowledge about the dynamics of gene networks may help pharmaceutical research in discovering targets for complex diseases, tailoring drug therapy to the needs of patients [14].

In this thesis we investigate a subpart of the SOS gene network at such level of abstraction.

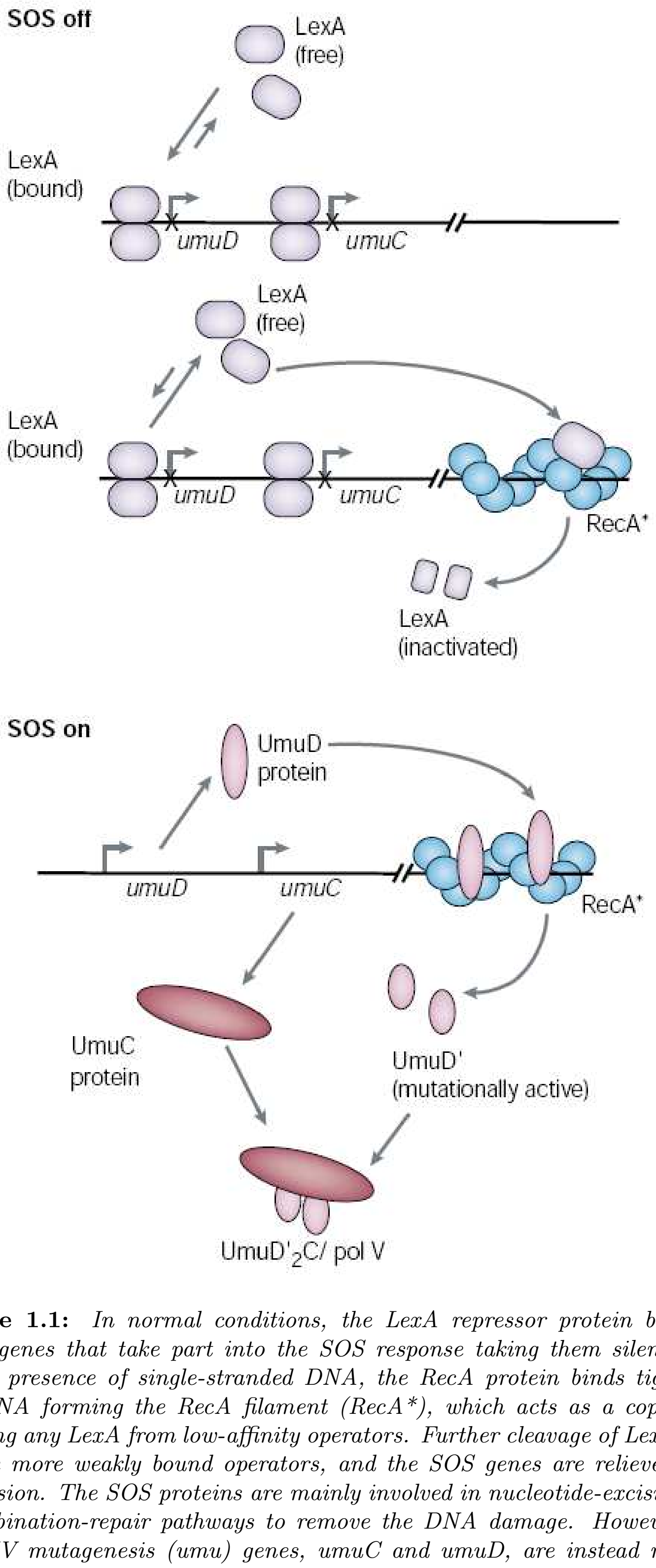

Figure 1.1: *In normal conditions, the LexA repressor protein binds to some genes that take part into the SOS response taking them silent. But in the presence of single-stranded DNA, the RecA protein binds tightly to the DNA forming the RecA filament (RecA*), which acts as a coprotease cleaving any LexA from low-affinity operators. Further cleavage of LexA frees up the more weakly bound operators, and the SOS genes are relieved from repression. The SOS proteins are mainly involved in nucleotide-excision and recombination-repair pathways to remove the DNA damage. However, the two UV mutagenesis (umu) genes, umuC and umuD, are instead required for replication past unrepaired lesions in the DNA template. They leave behind mutations targeted to sites of DNA damage. The figure shows the activaction of UmuD thanks to a cleavage reaction made by RecA and the subsequent formation of the multiprotein complex polV [17].*

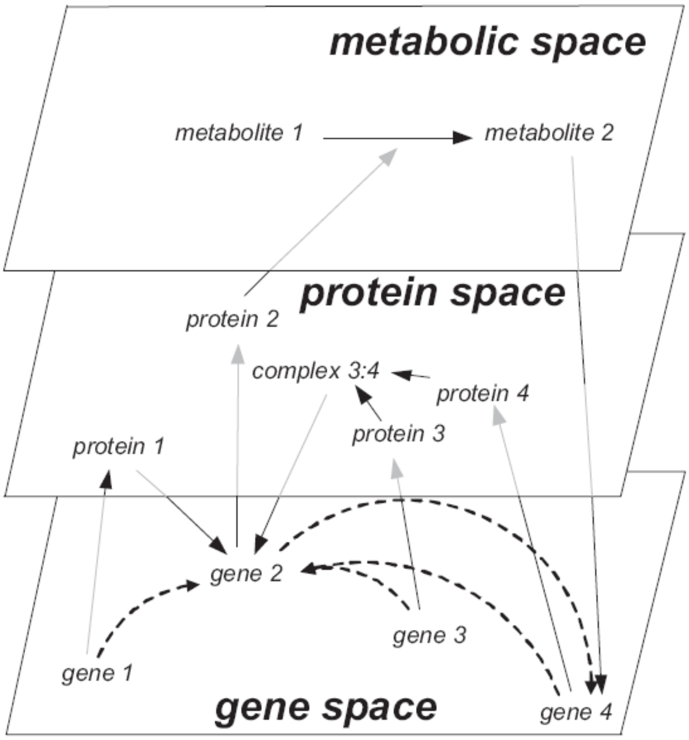

Figure 1.2: *Example of a biochemical network. Molecular constituents (nodes of the network) are organized in three levels (spaces): mRNAs, proteins, and metabolites. Lines indicate interactions. Projections of these interactions into the gene space, indicated by dashed arrows, constitute the corresponding gene network [13].*

1.3 The ODE framework

The purpose of mathematical modelling is to translate biological phenomena into a set of equations, determine the parameter values of that model to reflect a particular experimental set-up and then through simulations in such a way, confirming or disputing previous knowledge or hypotheses. The standard way to model (non-spatial) physical systems is using ODEs and they have been successfully applied to simulate metabolic systems [15] [16]. An example of an ordinary differential equation is the following:

$$\frac{d\mathbf{x}}{dt} = \sum f(\mathbf{x},\mathbf{p}) - \sum g(\mathbf{x},\mathbf{p})$$

where dx/dt represents the rate of change of the substrate concentration x, $f(x)$ corresponds to the steps leading to the accumulation of x, while $g(x)$ represents the steps that lead to a decrease in concentration of x.

The process of modeling identifies and defines the variables and the relationships among them and the simulation is an execution of the model (usually done on a computer). In case of differential equations models, a simulation refers to numerical integration, as the process of finding a solution to the set of equations.

The goal of using a set of ODEs is to describe a biological phenomen (like a gene network) to analyze the time dynamics of the system or to calculate the steady state values of the variables involved in the model.

1.4 A subnetwork of the SOS gene network

In this thesis we use four different ODEs models for a *subnetwork of the SOS global network*. We consider a well estabilished subnetwork that takes into account only those genes that seems to be the principle mediators of the SOS response. In Figure 1.3 the diagram shows the molecules involved in the response network, while Figure 1.4 shows the gene network derived from this diagram looking at the genes plane. The genes considered are the followings (we give for each one a short functional description) :

- *lexA*: repressor protein;
- *recA*: recombinase A;
- *ssB*: single-strand DNA-binding protein;
- *recF*: recombination protein F;
- *dinI*: DNA damage-inducible protein I;
- *umuDC*: DNA polymerase V subunits UmuC and UmuD;
- *rpoD*: RNA polymerase sigma factor;
- *rpoH*: RNA polymerase sigma factor;
- *rpoS*: RNA polymerase, sigma S (sigma 38) factor;

As described in Figure 1.3 a mutagenic agent, such as an antibiotic or UV rays can lead to the formation of single stranded DNA. As a consequence of this, the protein RecA binds to the single stranded DNA and becomes active RecA*. This active nucleoprotein filament promotes the cleavage of the repressor protein LexA which usually is bounded to the response gene promoters preventing their expression.

In Figure 1.3 the red lines ending with filled circles shows the genes directly affected by the LexA inhibition.

When this inhibition is removed the genes depicted as boxes in the picture, becomes active and can express their proteins whose functions are synthesized in circled boxes at the right side of the picture.

We are going to present several models for the considered subnetwork of the SOS response.

1.5 Mass Action and Michaelis Menten-type Models

Usually the interactions between genes are represented as inhibitions or activations of the rate of transcription of one mRNA by other mRNA species. In our attempt to find a good model for the SOS gene network, we propose three models using three different kind of kinetics.

We start from the “easiest” model, the one that uses mass action kinetics. In particular, we write a set of equations for the nine genes by considering for each gene the rate of variation in concentration expressed as variations of concentrations of the genes that acts as activators or inhibitors of the considered gene, with inhibitors modeled as activactors of degradation.

We write two kind of mass action kinetic equation: one that considers each gene variation dependent from the other genes as if they would act independently from each other and we call it *Mass Action Independent Model (MAI)*; in the other one, *Mass Action Dependent Model(MAD)*, the genes are instead modeled as they would interact in a cooperative way in influencing the gene expression. Below are defined the equations for the two models:

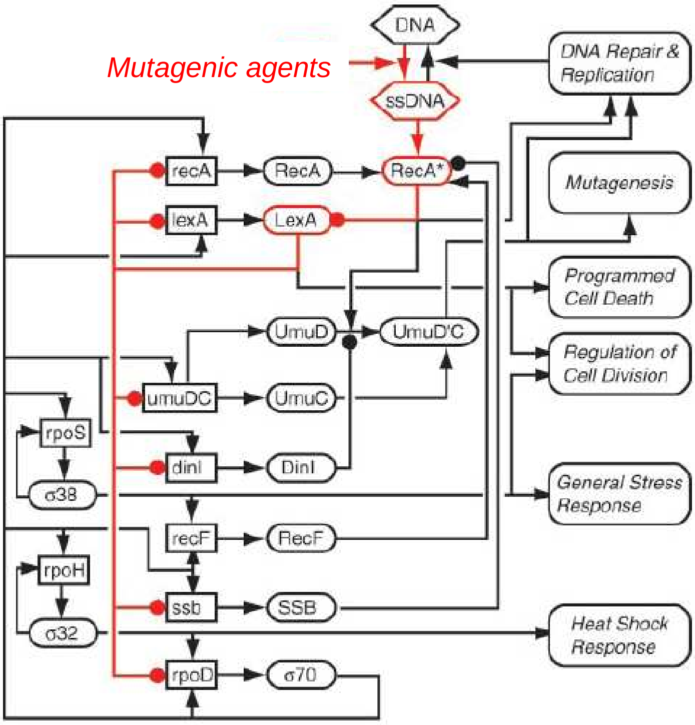

Figure 1.3: ***Diagram showing the principle mediators of the SOS response***. *DNA lesions caused by mutagenic agents are converted to single stranded DNA during chromosomal replication. The RecA protein after binding to the ssDNA is activated and induces LexA cleavage thereby diminishing the repression of the SOS response genes. Boxes denote genes, ellipses denote proteins, hexagon indicate metabolites, arrows denote positive regulation, filled circles denote negative regulation. Red emphasis denotes the primary pathway by which the network is activated after DNA damage [1].*

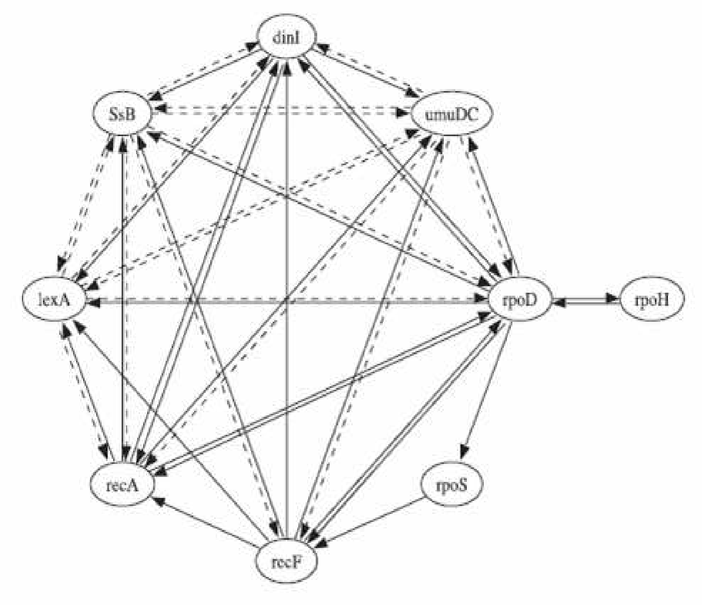

Figure 1.4: *Gene-gene interaction between the nine genes of SOS network in E.coli known in literature. Positive interactions are shown as line, and negative interactions are shown as dotted line [4].*

MAD model

$$
\begin{aligned}
\frac{dlexA}{dt} &= k_1\,[recA]\,[recF]\,[rpoD]\,[dinI] - k_2\,[SsB]\,[umuDC]\,[lexA] \\
\frac{drecA}{dt} &= k_3\,[recF]\,[rpoD]\,[dinI] - k_4\,[lexA]\,[SsB]\,[umuDC]\,[recA] \\
\frac{drecA}{dt} &= k_3\,[recF]\,[rpoD]\,[dinI] - k_4\,[lexA]\,[SsB]\,[umuDC]\,[recA] \\
\frac{drecF}{dt} &= k_5\,[rpoD]\,[rpoS] - k_6\,[SsB]\,[umuDC]\,[recF] \\
\frac{drpoS}{dt} &= k_7\,[rpoD] - k_8\,[rpoS] \\
\frac{drpoD}{dt} &= k_9\,[drecF]\,[recA]\,[dinI]\,[rpoH] - k_{10}\,[umuDC]\,[SsB]\,[lexA]\,[rpoD] \\
\frac{dumuDC}{dt} &= k_{11}\,[rpoD]\,[recF]\,[recA]\,[dinI] - k_{12}\,[SsB]\,[lexA]\,[umuDC] \\
\frac{ddinI}{dt} &= k_{13}\,[rpoD]\,[recF]\,[recA] - k_{14}\,[umuDC]\,[lexA]\,[SsB]\,[dinI] \\
\frac{dSsB}{dt} &= k_{15}\,[dinI]\,[rpoD]\,[recF]\,[recA] - k_{16}\,[umuDC]\,[lexA]\,[SsB] \\
\frac{drpoH}{dt} &= k_{17}\,[rpoD] - k_{18}\,[rpoH]
\end{aligned}
$$

MAI model

$$
\begin{aligned}
\frac{dlexA}{dt} &= k_1\,[recA] + k_2\,[recF] + k_3\,[rpoD] + k_4\,[dinI] \\
&\quad - (k_5\,[SsB] + k_6\,[umuDC])\,[lexA] \\
\frac{drecA}{dt} &= k_7\,[recF] + k_8\,[rpoD] + k_9\,[dinI] \\
&\quad - (k_{10}\,[lexA] + k_{11}\,[SsB] + k_{12}\,[umuDC])\,[recA] \\
\frac{drecF}{dt} &= k_{13}\,[rpoD] + k_{14}\,[rpoS] - (k_{15}\,[SsB] + k_{16}\,[umuDC])\,[recF] \\
\frac{drpoS}{dt} &= k_{17}\,[rpoD] - k_{18}\,[rpoS] \\
\frac{drpoD}{dt} &= k_{19}\,[recF] + k_{20}\,[recA] + k_{21}\,[dinI] + k_{22}\,[rpoH] \\
&\quad - (k_{23}\,[umuDC] + k_{24}\,[SsB] + k_{25}\,[lexA])\,[rpoD] \\
\frac{dumuDC}{dt} &= k_{26}\,[rpoD] + k_{27}\,[recF] + k_{28}\,[recA] + k_{29}\,[dinI] \\
&\quad - (k_{30}\,[SsB] + k_{31}\,[lexA])\,[umuDC] \\
\frac{ddinI}{dt} &= k_{32}\,[rpoD] + k_{33}\,[recF] + k_{34}\,[recA] \\
&\quad - (k_{35}\,[umuDC] + k_{36}\,[lexA] + k_{37}\,[SsB])\,[dinI] \\
\frac{dSsB}{dt} &= k_{38}\,[dinI] + k_{39}\,[rpoD] + k_{40}\,[recF] + k_{41}\,[recA] \\
&\quad - (k_{42}\,[umuDC] + k_{43}\,[lexA])\,[SsB] \\
\frac{drpoH}{dt} &= k_{44}\,[rpoD] - k_{45}\,[rpoH]
\end{aligned}
$$

were $k_1, ..., k_{45}$ are constants taken in the set of real numbers.

The third model we develop is written by supposing a Michaelis-Menten type kinetic. This kind of kinetic comes from the study of the interaction between enzyme and substrate in a chemical reaction. In particular Michaelis and Menten suggested that during a chemical reaction an equilibrium exists between free and substrate-bound enzyme, governed by a dissociation constant K_d.

The key element of their derivation was the assumption that the rate of the reaction was dependent on the fraction of total enzyme that had the substrate bound. Precisely the kinetic law is the following (see, e.g., [5]):

$$v_0 = \frac{V_{max}A}{K_A + A}$$

where A is the substrate, k_m the Michaelis Menten constant and V_{max} represents a constant for a given enzyme and is the theoretical maximal rate of the reaction. We used a modified Michaelis Menten equation considering A as the gene that influence the substrate concentration. Using this function we can define the equations for the nine genes of the considered subnetwork and we call the obtained model MMM-type (Michaelis-Menten type model):

MMM-type model

$$
\begin{aligned}
\frac{dlexA}{dt} &= \frac{V_{max1}\,[recA]}{k_{m1}+[recA]} + \frac{V_{max2}\,[recF]}{k_{m2}+[recF]} + \frac{V_{max3}\,[rpoD]}{k_{m3}+[rpoD]} + \frac{V_{max4}\,[dinI]}{k_{m4}+[dinI]} \\
&\quad -\left(\frac{V_{max5}\,[SsB]}{k_{m5}+[SsB]} + \frac{V_{max6}\,[umuDC]}{k_{m6}+[umuDC]} + k_{1_b}\,[lexA]\right) \\
\frac{drecA}{dt} &= \frac{V_{max7}\,[recF]}{k_{m7}+[recF]} + \frac{V_{max8}\,[rpoD]}{k_{m8}+[rpoD]} + \frac{Vmax9\,[dinI]}{k_{m9}+[dinI]} \\
&\quad -\left(\frac{V_{max10}\,[lexA]}{k_{m10}+[lexA]} + \frac{V_{max2}\,[SsB]}{k_{m11}+[SsB]} + \frac{V_{max12}\,[umuDC]}{k_{m12}+[umuDC]} + k_{2b}\right)[recA] \\
\frac{drecF}{dt} &= \frac{V_{max13}\,[rpoD]}{k_{13}+[rpoD]} + \frac{V_{max14}\,[rpoS]}{k_{m14}+[rpoS]} \\
&\quad -\left(\frac{V_{max15}\,[SsB]}{k_{m15}+[SsB]} + \frac{V_{max16}\,[umuDC]}{k_{m16}+[umuDC]} + k_{3b}\,[recF]\right) \\
\frac{drpoS}{dt} &= \frac{Vmax17\,[rpoD]}{k_{m17}+[rpoD]} \\
&\quad -k_{18}\,[rpoS] \\
\frac{drpoD}{dt} &= \frac{Vmax19\,[recF]}{k_{m19}+[recF]} + \frac{V_{max20}\,[recA]}{k_{m20}+[recA]} + \frac{V_{max21}\,[dinI]}{k_{m21}+[dinI]} + \frac{V_{max22}\,[rpoH]}{k_{22}+[rpoH]} \\
&\quad -\frac{V_{max23}\,[umuDC]}{k_{m23}+[umuDC]} + \frac{V_{max24}\,[SsB]}{k_{m24}+[SsB]} + \frac{V_{max25}\,[lexA]}{k_{m25}+[lexA]} + k_{5b}\,[rpoD] \\
\frac{dumuDC}{dt} &= \frac{V_{max26}\,[rpoD]}{k_{m26}+[rpoD]} + \frac{V_{max27}\,[recF]}{k_{m27}+[recF]} + \frac{V_{max28}\,[recA]}{k_{m28}+[recA]} + \frac{V_{max29}\,[dinI]}{k_{m29}+[dinI]} \\
&\quad -\left(\frac{V_{maz30}\,[SsB]}{k_{m30}+[SsB]} + \frac{V_{max31}\,[lexA]}{k_{m31}+[lexA]} + k_{6b}\,[umuDC]\right) \\
\frac{ddinI}{dt} &= \frac{V_{max32}\,[rpoD]}{k_{m32}+[rpoD]} + \frac{V_{max33}\,[recF]}{k_{m23}+[recF]} + \frac{V_{max34}\,[recA]}{k_{m34}+[recA]} \\
&\quad -\left(\frac{V_{max35}\,[umuDC]}{k_{m35}+[umuDC]} + \frac{V_{max36}\,[lexA]}{k_{m36}+[lexA]} + \frac{V_{max37}\,[SsB]}{k_{m37}+[SsB]} + k_{7b}\,[dinI]\right) \\
\frac{dSsB}{dt} &= \frac{V_{max38}\,[dinI]}{k_{m38}+[dinI]} + \frac{V_{max39}\,[rpoD]}{k_{m39}+[rpoD]} + \frac{V_{max40}\,[recF]}{k_{m40}+[recF]} + \frac{V_{max41}\,[recA]}{k_{m41}\,[recA]} \\
&\quad -\left(\frac{V_{max42}\,[SsB]}{k_{m42}+[SsB]} + \frac{V_{max43}\,[lexA]}{k_{m43}+[lexA]} + k_{8b}\,[SsB]\right) \\
\frac{drpoH}{dt} &= \frac{V_{max44}\,[rpoD]}{k_{m44}+[rpoD]} - k_{9b}\,[rpoH]
\end{aligned}
$$

1.6 'Mendes' model

Another kind of approach we have investigated starts from the work of Mendes and collaborators [6]. In particular, in [6], the changes in the rate of transcription of an arbitrary gene depends on the changes in concentration of a few other gene products. Given that the number of gene copies for each gene is limited, and transcription is catalyzed by a limited number of transcription complexes, the rate law is then mainly influenced by the quantity of activators or inhibitors. This is formally expressed by the following general rate law for the transcription [6]:

$$\frac{dx_i}{dt} = V_i \cdot \prod_j \left(\frac{K_{i_j}^{n_j}}{I_j^{n_j} + K_{i_j}^{n_j}} \right) \cdot \prod_k \left(1 + \frac{A_k^{n_k}}{A_k^{n_k} + {Ka_k}^{n_k}} \right)$$

where dx_i/dt stands for gene rate of transcription, A_i and I_j are the quantities of activators and inhibitors which act indipendently from each other, V_i is the basal rate for transcription, and K_{i_j} and Ka_k are the concentrations at which the effect of the inhibitor or the activator, respectively, is half of its saturating value. The exponents n_i and n_k regulate the sigmoidicity of the curve like in the Hill kinetic [7]. For the mRNA degradation steps we use a simple mass action kinetics. Because this model is inspired from [6] we call it MM (Mendes Model).

For the considered subnetwork of genes there is the same number of activators and inhibitors. Below, are described the equations for the genes. The arrows with captions indicates the activaction and inactivaction of the gene in between and the caption represents the genes that influence these two reactions.

MM model

Rate laws for synthesis

4 Activators - 2 Inhibitors

$\overset{recA,recF,rpoD,dinI}{\longrightarrow} lexA \overset{SsB,umuDC}{\longrightarrow}$

$\overset{recA,recF,rpoD,dinI}{\longrightarrow} Ssb \overset{lexA,umuDC}{\longrightarrow}$

$\overset{dinI,recA,recF,rpoD}{\longrightarrow} umuDC \overset{SsB,lexA}{\longrightarrow}$

$$\begin{aligned} V_{synt_i} = V_1 \cdot & \frac{K_{1i}^{n1i}}{I1^{n1i} + K_{1i}^{n1i}} \cdot \frac{K_{2i}^{n2i}}{I2^{n2i} + K_{2i}^{n2i}} \cdot \left(1 + \frac{A_1^{n1}}{A_1^{n1} + K1^{n1}} \right) \\ & \cdot \left(1 + \frac{A_2^{n2}}{A_2^{n2} + K2^{n2}} \right) \cdot \left(1 + \frac{A_3^{n3}}{A_3^{n3} + K3^{n3}} \right) \\ & \left(1 + \frac{A_4^{n4}}{A_4^{n4} + K4^{n4}} \right) \end{aligned}$$

3 Activators - 3 Inhibitors

$\overset{recA,recF,rpoD}{\longrightarrow} dinI \overset{SsB,lexA,umuDC}{\longrightarrow}$

$\overset{dinI,rpoD,recF}{\longrightarrow} recA \overset{lexA,SsB,umuDC}{\longrightarrow}$

$$V_{synt_i} = V_1 \cdot \frac{K_{1i}^{n1i}}{I1^{n1i} + K_{1i}^{n1i}} \cdot \frac{K_{2i}^{n2i}}{I2^{n2i} + K_{2i}^{n2i}} \cdot \frac{K_{3i}^{n3i}}{I3^{n3i} + K_{3i}^{n3i}} \cdot \left(1 + \frac{A1^{n1}}{A1^{n1} + K1^{n1}}\right) \cdot \left(1 + \frac{A_2^{n2}}{A_2^{n2} + K2^{n2}}\right) \cdot \left(1 + \frac{A_3^{n3}}{A_3^{n3} + K3^{n3}}\right)$$

2 Activators - 2 Inhibitors

$\overset{rpoS,rpoD}{\longrightarrow} recF \overset{umuDC,SsB}{\longrightarrow}$

$\overset{dinI,rpoD,recF}{\longrightarrow} recA \overset{lexA,SsB,umuDC}{\longrightarrow}$

$$V_{synt_i} = V_1 \cdot \frac{K_{1i}^{n1i}}{I1^{n1i} + K_{1i}^{n1i}} \cdot \frac{K_{2i}^{n2i}}{I2^{n2i} + K_{2i}^{n2i}} \cdot \frac{K_{3i}^{n3i}}{I3^{n3i} + K_{3i}^{n3i}} \cdot \left(1 + \frac{A_1^{n1}}{A_1^{n1} + K1^{n1}}\right) \cdot \left(1 + \frac{A_2^{n2}}{A_2^{n2} + K2^{n2}}\right) \cdot \left(1 + \frac{A_3^{n3}}{A_3^{n3} + K3^{n3}}\right)$$

1 Activator - No Inhibitors

$\overset{rpoD}{\longrightarrow} rpoH$

$\overset{rpoD}{\longrightarrow} rpoS$

$$V_{synt_i} = V_1 \cdot \left(1 + \frac{A_1^{n1}}{A_1^{n1} + K1^{n1}}\right)$$

Rate law for degradation

$$V_{deg_i} = K_b \cdot A_i$$

1.7 S-system model

Finally we present a model of the subnetwork based on equations in style of the S-systems framework [18]. The basic idea of this model is to represent interactions between biochemical species with power-law dynamics. In S-systems [19] [20], in which the set of genes activating another gene and the set of genes inhibiting it are aggregated in a power-product term, in the following way:

$$\frac{dX_i}{dt} = \alpha_i \cdot \prod_{j=1}^{n} X_j^{g_{ij}} - \beta_i \prod_{k=1}^{m} X_k^{h_{ij}}.$$

where X represents the variables (dynamic concentrations of internal metabolites), α_i and β_i are rate constants, and g_{ij}, h_{ij} are kinetic orders, which may be non-integer and non-positive; in our case X represent concentrations of genes.

These equations have many advantages: first, it is a concise representation, in the sense that the kinetic description of each process takes a nearly minimal number of parameters. Moreover the mathematical regularity can bring through effective numerical methods for the integration of the differential equations [18].

Since the details of the molecular mechanisms that govern interactions among system components are not substantially known or well understood, the description of these processes through the S-system can give a representation that is general enough to capture the essence of the experimentally observed response. We report below the equations of the considered subnetwork in the S-system model.

S-system model

$$
\begin{aligned}
\frac{dlexA}{dt} &= V_1 \cdot [SsB^{g_{21}} \cdot dinI^{g_{31}} \cdot umuDC^{g_{41}} \cdot rpoD^{g_{51}} \cdot recF^{g_{81}} \cdot recA^{g_{91}}] \\
&\quad -V_d \cdot lexA \\
\frac{dSsB}{dt} &= V_2 \cdot [lexA^{g_{12}} \cdot dinI^{g_{32}} \cdot umuDC^{g_{42}} \cdot rpoD^{g_{52}} \cdot recF^{g_{72}} \cdot recA^{g_{91}}] \\
&\quad -V_d \cdot SsB \\
\frac{ddinI}{dt} &= V_2 \cdot [lexA^{g_{13}} \cdot SsB^{g_{23}} \cdot umuDC^{g_{43}} \cdot rpoD^{g_{53}} \cdot recF^{g_{83}} \cdot recA^{g_{93}}] \\
&\quad -V_d \cdot dinI \\
\frac{dumuDC}{dt} &= V_4 \cdot [lexA^{g_{14}} \cdot SsB^{g_{24}} \cdot dinI^{g_{34}} \cdot rpoD^{g_{54}} \cdot recF^{g_{84}} \cdot recA^{g_{94}}] \\
&\quad -V_d \cdot umuDC \\
\frac{drpoD}{dt} &= V_5 \cdot \left[lexA^{g_{15}} \cdot SsB^{g^{25}} \cdot dinI^{g_{35}} \cdot umuDC^{g_{45}} \cdot rpoH^{g_{65}} \cdot recF^{g_{85}} \cdot recA^{g_{95}}\right] \\
&\quad -V_d \cdot rpoD \\
\frac{drpoH}{dt} &= V_6 \cdot [rpoD^{g_{56}}] - V_d \cdot rpoH \\
\frac{drpoS}{dt} &= V_7 \cdot [rpoD^{g_{57}}] - V_d \cdot rpoS \\
\frac{drecF}{dt} &= V_8 \cdot [SsB^{g_{28}} \cdot umuDC^{g_{48}} \cdot rpoD^{g_{58}} \cdot rpoS^{g_{78}}] \\
&\quad -V_d \cdot recF \\
\frac{drecA}{dt} &= V_9 \cdot [lexA^{g_{19}} \cdot SsB^{g_{29}} \cdot dinI^{g_{39}} \cdot umuDC^{g_{49}} \cdot rpoD^{g_{59}} \cdot recF^{g_{89}}] \\
&\quad -V_d \cdot recA
\end{aligned}
$$

Chapter 2

Parameter Estimation

Each of the models presented in Chapter 1 contains a different set of parameters that need to be estimated. To achieve this task we have used the optimization algorithms available in the software COPASI (Complex Pathway Simulator). In this chapter we will shortly describe the two algorithms that have been principally used in our analyses and we explain the reasons of their choice.

2.1 Overview of optimization algorithms

Once defined the models, to fit their dynamics to the experimental data we need to find the values of the parameters for the model that would produce the observed behaviour.

Here a technical clarification needs to be done: in modelling and simulation with the word *parameter* we mean those entities of a model that are constant in time, while with *variables* we mean entities whose values vary in time and are entirely determined by the parameters.

In optimization problems, although these words share the same meaning, the *variable* of the simulation are now part of the objective function and the *parameters* to optimize becomes variables (because they are varied in the course of optimization) [25]. *The problem in parameter estimation of non-linear dynamic systems is to minimize a function that measure the goodness of the fit of the model compared to a given experimental dataset.*

There are different kinds of optimization algorithms and the most used falls into three classes [26]:

- gradient search;
- deterministic direct search;
- stochastic direct search.

Gradient search methods look for the nearest minimum of the function by following a direction that is determined from derivatives of the function. The most popular method included in this class is the Levenberg-Marquardt method [21] [22]. A feature that characterizes these algorithms is that they search for the minimum that is closest to the initial guess but this could not be the global minimum.

Direct search methods are those that do not need to calculate derivatives to minimize an objective function. These algorithms are deterministic and use strategies based on keeping memory of previous solution and decide where to move step by step. Examples of algorithms that use this strategy are Nelder and Mead [23] and Hooke and Jeeves [24].

Gradient descent methods or deterministic direct search methods are local minimizers and this can be a problem because of the presence of so-called local minima. When the set of parameters to be determined is very large we need to use methods that are able to find global minima, these methods are called 'Global optimization methods'.

Global optimization methods can be classified as deterministic and stochastic strategies [27]. Of the two approaches the deterministic approach can lead to a more accurate solution, but with the increase of the parameters to estimate the time of computation becomes too big [27]. In contrast, stochastic algorithms can locate the vicinity of the solution in a reasonable computational time even if the values obtained are not so precise. Examples of global optimization algorithms are those grouped into the so called 'Evolutionary Computation' (EC) techniques such as Genetic Algorithms [28], Evolutionary Programming [29], Evolution Strategies [30]. This group of algorithms is a very popular class of methods based on the ideas of biological evolution driven by the mechanism of reproduction, mutation and the principle of survival of the fittest. The strategies used by this group exploit a set of potential solutions, named *population*, and detect the optimal problem solution through cooperation and competition among the individuals of the population [27]. Other examples of GO algorithms that take inspiration from nature are Particle Swarm [31] and 'Simulated Annealing' algorithms, the former takes inspiration from flock of birds searching for food and the second starts from the simulation of the cooling of metals where atoms adopt the most stable configuration as slow cooling of a metal takes place [32].

2.2 Optimization methods used in the thesis

In the thesis we focus the attention on two optimization algorithms (among the ones present in the software COPASI). These methods were the ones able to produce the best fit, i.e., they produce parameters for the models in such a way that the behaviour of the models was close to the experimental observed behaviour. The comparison among the behaviours (modelled and experimental) was done by using the root mean square value. Here we will give a brief description of these two methods.

2.2.1 Particle swarm

Particle swarm optimization method is inspired by a flock of birds or a school fish searching for food [31]. A swarm consists of N particles flying around in a D-dimensional search space; each particle has a position X_i and a velocity V_i in the parameter space and is able to successively adjust its position toward the global optimum according to two factors: the best positive visited so far (lbest) and the best position visited by the whole swarm (gbest). After finding the two best values the particle updates its velocity and positions with the following equations:

$$\begin{aligned}V_i\left(t+1\right) &= \omega V_i\left(t\right)+c_1\cdot rand\left(\right)\cdot\left(p_i-X_i\left(t\right)\right)+c_2\cdot rand\left(\right)\cdot\left(p_g-X_i\left(t\right)\right).\\ X_i\left(t+1\right) &= X_i\left(t\right)+V_i\left(t+1\right).\end{aligned}$$

where $X_i\left(t\right)$ is the current particle position, $V_i\left(t\right)$ is the particle velocity, p_i and p_g are lbest and gbest, $rand\left(\right)$ is a random number uniformly distributed between $(0,1)$, c_1, c_2 are constants and ω is called inertial weight and is used to limit the velocity.

Through these steps, the particle position is updated until a minimum error criteria is attained.

2.2.2 Hooke and Jeeves

The method of Hooke and Jeeves is a direct search algorithm that searches for the minimum of a nonlinear function without requiring derivatives of the function. It searches for trial solutions involving comparison of each trial solution with the "best" obtained up to that time together with a strategy for determining (as a function of earlier results) what the next trial solution will be [24].

Starting from a space of points P, which represent possible solutions, a point B_0 is arbitrarily selected to be the first "base point". A second point, P_1, is chosen and compared with B_0, if P_1 is a better solution than B_0, P_1 becomes the second base point B_1, if not, P_1 take the same value as B_0.

This process continues, and the strategy for selecting new trial points is determined by a set of finite "states" which provide the memory, so the search goes through a descent direction using the values of the function calculated in a number of previous iterations until the final state S is reached.

2.3 Choosing the best fit

The optimization problem is stated as the minimization of a weighted distance measure J between experimental and predicted values of the state variables represented by the vector y_p. Formally,

$$J=\sum_{i=1}^{n}\omega_i\left[y_{pred(i)}-y_{exp(i)}\right]^2.$$

This represents the objective value, where n is the number of data for each experiment, y_{exp} are the known experimental data, and y_{pred} represents the theoretical values obtained using the model with a given set of parameters . The terms ω_i correspond to the different weights taken to normalize the contribution of each term [27].

The software COPASI provides three methods to calculate these weights, but for our estimations we selected the weights calculated with the mean square method:

mean square $\omega_j = 1/\sqrt{< y^2_{exp(i)} >}$.

Once executed the algorithms the software returns the values of the root mean square for each gene calculated taking the square root of the objective value divided by the experimental points available (in the studied case, $n = 6$):

$$rms = \sqrt{\frac{\mathrm{J}}{n}}$$

Then the software provides a *global root mean square* value by considering the root mean square value of the ones calculated for each gene. This value is the one that we have used to compare the different optimization algorithms: The smaller is this value and the better is the fit of the curve (obtained from the model) to the experimental data. Formally such root mean square is defined as:

$$RMS = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(rms)^2}.$$

where in this case, m is the number of the genes considered .

Chapter 3

Methods

3.1 ***CO***mplex ***PA***thway ***SI***mulator software

COPASI (Complex Pathway simulator) [2] is a software package that permits the simulation of biochemical pathway through a simple user interface. In COPASI we can choose a biochemical or a mathematical view. From the biochemical widget it is possible to insert reactions, selecting substrates, products and modifiers (which are the enzymes that influence a reaction). Once inserted all the reactions the program is able to translate all in differential equations according to the function that has been previously inserted to describe the dynamic of the reaction. In another widget called *Multiple task* it is possible to do different kind of analyses. In particular, in this work, we present the steady state analyses and parameter estimation analyses. In the parameter estimation analyses there are several optimization algorithms like: genetic algorithm, hooke and jeeves, Levenberg-Marquardt, Evolutionary Programming, random search, Nelder-Mead, Particle swarm, simulated annealing, evolution strategy and steepest descent. Usually is better to test which of the algorithms is best suited for the experimental data under consideration, since there is no general algorithm best for general data: each algorithm can be good on some data and bad on other data. Judged over all possible datasets all algorithms are equally good, so if an algorithm is better on one dataset it must be worst on other datasets (the so called no-free lunch theorem [34]). In COPASI it is even possible to do the analyses using different experimental dataset all togheter. We have used the COPASI parameter estimation to find the parameter values for our model. After the parameters have been calculated, we have analyzed the steady state values and we have compared the data obtained from the simulation with the experimental ones available in the M3D database.

time	ssB	lexA	recF	rpoH	rpoD
0	10.520933	10.3338	9.472773	9.713127	10.5659
12	11.271833	11.154867	11.276533	10.284833	12.194267
24	11.689033	11.313133	10.390933	9.683377	11.67
36	11.571967	11.519667	10.177587	9.54622	11.1672
48	11.425567	11.5337	10.1489	9.358997	11.077733
60	11.312734	11.465633	10.0525	9.47462	10.948433

Table 3.1: Time series data from a Norfloxacin perturbation for some of the SOS response genes.

time	rpoS	recA	umuD	dinI
0	10.564034	10.4524	7.34347	9.311147
12	10.420633	13.559767	8.087973	11.505433
24	10.3458	13.7375	8.965397	12.3316
36	10.031827	13.681466	9.106427	12.531933
48	9.770947	13.5981	9.236267	12.512233
60	9.80332	13.656733	9.329837	12.516367

Table 3.2: Time series data from a Norfloxacin perturbation for some of the SOS response genes.

gene	dinI U-N0025	lexA U-N0025	recA U-N0025	umuD U-N0025
lexA	11.468867	12.87	11.384833	11.3117
dinI	12.465967	10.042667	10.407267	10.476067
umuD	8.038043	7.73134	8.03572	13.200167
rpoD	10.545367	10.5738	10.620433	10.619833
rpoH	9.45017	9.468853	9.41849	9.424997
rpoS	10.8222	10.740933	10.577067	10.538466
recF	9.619356	9.634883	9.67105	9.814993
recA	11.714533	11.3685	13.254167	11.5945
Ssb	10.544267	10.5773	10.641367	10.655033

Table 3.3: Overexpression experiments for four of the SOS response genes, U-0025 stays for upregulation and 0025 corresponds to the norfloxacin antibiotic concentration that is $0.025\mu g/ml$.

3.2 Many Microbe Microarrays Database

The Many Microbe Microarrays Database M3D contains over 1000 microarrays for *E. coli* (507), *Saccaromices cerevisiae* (530) and *Shewanella onideinsis* (14) all of which collected and combined from individual investigators GEO, Array Express and ASAP. The expression data is uniformly normalized and simple analysis tools are provided for the interrogation of the dataset. It is possible to download raw data and normalized data. The raw probe-level microarray data are normalized as a group with RMA. To use this procedure all expression profiles for a particular array-design are collected, uniformly normalized, and deposited as a so-called 'build'. Periodically, the database is updated with new expression profiles and all the data are renormalized. Browsing the database, it is possible to select any experiment and then submit the list of genes on which one is interested. For our purpose we have downloaded the expression profiles of the nine genes of the SOS response in E. coli. The same database also provides heat plots, expression histograms, scatter plots and a genome browser for visualization in a genome context.

The build used (E. coli v31) contains 445 microarrays covering 189 experimental conditions. We have used two of the experiments available in this build:

- **Time course experiment**: Time series data from a Norfloxacin perturbation where samples from perturbed cultures were taken at 0, 12, 24, 36, 48 and 60 minutes (see Tables 3.1 and 3.2);

- **Steady state experiment**: Overexpression perturbations from four of the nine genes investigated: *dinI, lexA, recA* and *umuD* in LB[1] $0.025\mu g/ml$ norfloxacin antibiotic[2] (see Table 3.3). DNA damage responses were induced by growing transformed E. coli cells for 3 hours in $0.025\mu g/ml$ of norfloxacin antibiotic [33].

The treatment of *E.coli* with Norfloxacin is equivalent to making a perturbation to recA [4]. Norfloxacin is a member of fluoroquinolone[3] class of antimicrobial agents that target the prokaryotic type II topoisomerase (DNA gyrase[4]) and topoisomerase IV inducing the formation of single-stranded DNA and thus activating the SOS pathway via activation of the recA protein.

[1] LB: cultivated nutritive substrate.

[2] genes of interest were overexpressed in E. coli by transforming these bacterial cells with vector pBADX53. This vector generally increases gene expression level 2 to 10 fold above native expression level

[3] The quinolones are a family of broad-spectrum antibiotics. The majority of quinolones in clinical use belong to the subset of fluoroquinolones, which have a fluoro group attached the central ring system.

[4] DNA gyrase is an essential bacterial enzyme that catalyzes the ATP-dependent negative super-coiling of double-stranded closed-circular DNA.

Chapter 4

Results

4.1 Selecting the optimization algorithm

To estimate the parameter values of the models presented in Chapter 1 we have used the algorithms available in the software COPASI. In particular, we have applied the algorithms in the case of the MAD, MAI and MMM models. To compare the algorithms results with the experimental data, we used the global root mean square value calculated by COPASI as described in Chapter 2.

Table 4.1 shows these results using dataset v31 (described in Chapter 3) and setting the starting value for all the parameters to infer to 0.1 and the boundaries for the search between 0.1 and 10.

As we can see, particle swarm optimization method is the one that provides the best fit among the eleven algorithms, since gives the smaller root mean square value. Moreover the computational time was much smaller (15 minutes) for particle swarm algorithm than the one necessary for completing the simulated annealing algorithm (6 hours for MAD model, for the other two models the computational time was clearly excessive).

Considering the results presented in Table 4.1 we have decided to use Particle swarm and the MMM model for developing our analyses by considering the dataset v31.

4.1.1 Michaelis Menten-type model parameter estimation

We have started by estimating the parameters for the Michaelis Menten-type model, presented in Chapter 2, using boundaries between 0.1 and 10, considering 0.1 as starting value and running the algorithm several times (34 times) and for each run we use the previous obtained value, as new starting value. The value of the root mean square decreased in this way from 0.277723 to 0.208325. Then we changed the boundaries fixing them between 0 and 100 but after 10 runs of the algorithm, the root mean square stayed fixed to the value of 0.279178 (which, however, can be considered still not an adequate fit).

4.1.2 Mendes model and S-system model parameter estimation

Because the Michaelis Menten-type model was not enterely succesfull, we decided to test the Mendes Model and the S-system (described in Chapter 1).

In this case we have started by using wider boundaries for the parameters: $10E^{-12}$ and $10E^{06}$ and we have made some trials using different starting values for the parameters in the Mendes Model with the following results:

- starting value root mean square (RMS) 0.01 → 0.153064;
- starting value 0.1 → 0.171091;
- starting value 1 → 0.248452;
- starting value 10 → 0.279178;
- starting value 100 → 0.279178;

Considering the obtained values we have decided to use an initial value for the parameter of 0.01, for both MM and S-system models. After several runs of the particle swarm algorithm followed by the local optimizer Hooke and Jeeves we have obtained the root mean square values of: 0.0871909 for the S-system and 0.153064 for the Mendes model.

In Figures 4.1, 4.2, 4.3 and 4.4 we present the fit obtained by using particle swarm algorithm for each of the nine genes of the network presented in Chapter 1 and in the tables 4.3, 4.4, 4.5, 4.6, 4.7, 4.8, 4.9, 4.10, the values obtained for the parameters in the Mendes model and in the S-system model. The plots show a comparison between the experimental data fitted using Microsoft office excel and the fit obtained using the models equations (with the estimated parameter values) obtained through the particle swarm algorithm and Hooke and Jeeves one.

Method	MAD	MAI	MMM
Genetic algorithm	7.41043	4.07289	2.73041
Genetic algorithm SR	8.01853	4.04922	77.6712
Hooke and Jeeves	5.86722	3.5768	no result
Levenberg-Marquardt	4.66375	9.07598	no result
Evolutionary Programming	6.68054	4.56884	5.01391
Random Search	9.09124	9.07598	0.278172
Nelder Mead	7.91474	7.64839	no result
Particle swarm	6.19276	3.57677	0.277723
Simulated annealing	4.53036	...	...
Evolution Strategy	6.05042	9.07598	1.36394
Steepest Descent	7.64048	4.6486	no result

Table 4.1: Root Mean Square values of the fitting using the listed methods

4.1.3 Steady state predictions

In order to test the ability of the used models, to predict dynamics of the network we have used the software f and calculate the steady state values of the nine considered genes. The steady state is a state in which there are no changes in concentrations of the considered variables. Table 4.2 shows the results obtained for the two considered models, compared to the experimental data. As we can see even if the time course simulation gave a better fit for the S-system model, this model gives very different steady state values while the Mendes model gives a much better approximation of these values only for some of the considered genes.

Genes	Experimental data	Mendes Model	S-system model
lexA	11.47125	11.4599	0.112364
dinI	9.916875	12.1365	1.07362
umuD	7.819235	9.35587	14.9465
rpoD	9.46181	5.20705	0.0463255
rpoH	8.59825	1.12474	0.02446
rpoS	10.3827	3.46706	0.00736807
recF	8.974975	4.87275	0.203794
recA	11.79455	16.492	5.39728
ssB	10.21345	11.2432	0.17333

Table 4.2: Steady state values of experimental data compared to the ones obtained by the models simulations using the software COPASI.

4.1.4 Ability of prediction using gene overexpression experiments

The goal of mathematical modeling biochemical systems is to be able to make predictions of how the system responds to interventions. We here test how well our model predicts the system response to overexpressions in some of the state variables. We took the data from the M3D database in which some perturbation of the genes involved in the network were provided. In particular, as described in details in Chapter 3, we have used the overexpression experiments for four of the nine genes: *dinI, lexA, recA* and *umuD.*

To make comparisons between the experimental data and the values obtained from the models, we used the Co-response coefficient R.

The Co-response coefficient R represents how two system variables respond to a common perturbation, so that, using this coefficient we can quantify how perturbation propagate from one variable to another. In our case, the perturbation is on the gene overexpressed and so we have calculated the Co-response coefficient from both experimental data and from simulated data. Such coefficient is defined, similarly to [13] as:

$$R_j^i = \frac{\Delta x_j / x_j}{\Delta x_i / x_i}$$

where, for the coefficient R that represents the variation between experiments, Δx_j stands for the difference in concentration between each of the nine genes in the overexpression experiment and the experiment in normal conditions (WT). Δx_i stands for the difference in concentration between the gene directly overexpressed in the overexpression experiment and the same gene in the WT experiment.

To obtain the steady state concentrations of the genes in overexpression conditions using the two models ($x_{i_m}^{ss^*}$), we have calculated how much a gene is increased in the overexpression experiment and by that amount we have increased the model steady state and fixed it, using COPASI. In this way we have increased the level of the gene steady state concentration to the same percentage as increased in the experiment. We calculate this as follows:

$$x_{i_m}^{ss^*} = \frac{x_i^{ss^*}}{x_i^{ss}} \cdot x_{i_m}^{ss}$$

where $x_i^{ss^*}$ is the concentration at the steady state in the overexpression experiment, x_i^{ss} is the WT steady state concentration, and $x_{i_m}^{ss}$ is the steady state of the model in normal conditions. For instance, if we look at the values collected in Tables 4.11 and 4.13 in which *dinI* is upregulated, the obtained $x_{dinI_m}^{ss^*}$ for the Mendes model is:

$$x_{dinI_m}^{ss^*} = \frac{12.465967}{9.916875} \cdot 12.1365 = 15.25613749$$

Table 4.13 shows the data obtained.

We then calculated the R values both for models (Tables 4.14 and 4.16 show these values) and for the experiment. We have plotted all the results to have a visual comparison of how the R values differ between experiment and models (Figures 4.12,4.14,4.16).

Mendes model parameter values.

Parameter	Value	Parameter	Value
(lexA activaction).V1	0.412155	(SsB activaction).V1	0.834594
(lexA activaction).k3	1.00E+07	(SsB activaction).k3	0.489622
(lexA activaction).k4	7.87E+06	(SsB activaction).k4	0.000776705
(lexA activaction).K1i	1.45E+06	(SsB activaction).K1i	9.97E+06
(lexA activaction).K2i	6.26966	(SsB activaction).K2i	0.0853612
(lexA activaction).k1	193.315	(SsB activaction).k1	1.14E-11
(lexA activaction).k2	5.23E-11	(SsB activaction).k2	1.00E+07
(lexA activaction).n1	9.99567	(SsB inactivaction).Kb	0.0684776
(lexA activaction).n2	9.67547	(SsB activaction).n1	9.14704
(lexA activaction).n3	9.99997	(SsB activaction).n2	9.9845
(lexA activaction).n4	0.189053	(SsB activaction).n3	0.1
(lexA activaction).n1i	10	(SsB activaction).n4	9.18061
(lexA activaction).n2i	9.99999	(SsB activaction).n1i	9.99757
(lexA inactivaction).Kb	0.00138536	(SsB activaction).n2i	0.372758

Table 4.3: Parameter values for lexA and ssB genes.

Parameter	Value	Parameter	Value
(dinI activaction).V1	17379.6	(umuDC inactivaction).Kb	0.0600737
(dinI activaction).K3i	0.0124648	(umuDC activaction).V1	0.094125
(dinI activaction).K1i	0.156487	(umuDC activaction).k3	1.00E-11
(dinI activaction).K2i	25.5756	(umuDC activaction).k4	9.61E+06
(dinI activaction).K3	39.801	(umuDC activaction).K1i	4.04E+06
(dinI activaction).K1	0.0105687	(umuDC activaction).K2i	46906
(dinI activaction).K2	0.0120594	(umuDC activaction).k1	14.0415
(dinI inactivaction).Kb	0.0302833	(umuDC activaction).k2	0.00249712
(dinI activaction).n1	0.224947	(umuDC activaction).n1	0.100002
(dinI activaction).n2	6.85644	(umuDC activaction).n2	9.99955
(dinI activaction).n2i	2.09636	(umuDC activaction).n3	0.19867
(dinI activaction).n3i	1.71202	(umuDC activaction).n4	9.99999
(dinI activaction).n3	0.1	(umuDC activaction).n1i	1.67879
(dinI activaction).n1i	0.1	(umuDC activaction).n2i	4.72143

Table 4.4: Parameter values for dinI and umuDC genes

Parameter	Value	Parameter	Value
(rpoD inactivaction).Kb	0.00638541	(rpoH activaction).V1	0.000519047
(rpoD activaction).V1	0.482133	(rpoH activaction).k1	6.70E+06
(rpoD activaction).K1i	6.22827	(rpoH inactivaction).Kb	0.000461482
(rpoD activaction).K2i	8.29965	(rpoH activaction).n1	9.99987
(rpoD activaction).K3i	0.20023	(rpoS activaction).V1	0.00528418
(rpoD activaction).K3	1.00E-11	(rpoS activaction).k1	2016.23
(rpoD activaction).K1	4.05E-05	(rpoS inactivaction).Kb	0.0020656
(rpoD activaction).K2	0.0280862	(rpoS activaction).n1	0.1
(rpoD activaction).K4	0.000200763		
(rpoD activaction).n1i	9.99999		
(rpoD activaction).n2i	0.100051		
(rpoD activaction).n1	9.99859		
(rpoD activaction).n2	0.176835		
(rpoD activaction).n3	0.100052		
(rpoD activaction).n3i	0.101233		
(rpoD activaction).n4	3.8883		

Table 4.5: Parameter values for gene rpoD, rpoS and rpoH genes.

Parameter	Value	Parameter	Value
(recF degradation).Kb	0.00610749	(recA inactivaction).Kb	0.0226016
(recF activaction).V1	6.383	(recA activaction).V1	7727.73
(recF activaction).K1i	5.10438	(recA activaction).K3i	0.0302787
(recF activaction).K2i	1.44E+06	(recA activaction).K1i	0.266951
(recF activaction).k1	0.000129844	(recA activaction).K2i	6.65718
(recF activaction).k2	4.05E+06	(recA activaction).K3	4.32E+06
(recF activaction).n1i	9.99997	(recA activaction).K1	9.70E+06
(recF activaction).n2i	10	(recA activaction).K2	6.18E-07
(recF activaction).n1	10	(recA activaction).n1	9.96024
(recF activaction).n2	9.99998	(recA activaction).n2	9.99995
(recA activaction).n2i	9.99995		
(recA activaction).n3i	0.780803		
(recA activaction).n3	9.80912		
(recA activaction).n1i	0.100012		

Table 4.6: Parameter values for recF and recA genes.

S-system model parameter values.

Parameter	value	Parameter	Value
(lexA activation).V	0.120798	(SsB activaction).V	0.0713686
(lexA activation).g1	-1.80E-05	(SsB activaction).g1	-8.04E-06
(lexA activation).g2	1.62441	(SsB activaction).g2	0.069987
(lexA activation).g3	-1.13139	(SsB activaction).g3	-0.102948
(lexA activation).g4	0.0128164	(SsB activaction).g4	0.467413
(lexA activation).g5	2.73E-06	(SsB activaction).g5	0.293946
(lexA activation).g6	0.0232403	(SsB activaction).g6	0.266941
(lexA inact).k1	0.0565731	(SsB inact).k1	0.0732245

Table 4.7: Parameter values for lexA and ssB genes.

Parameter	value	Parameter	value
(dinI activation).V	0.233409	(umuDC activation).V	0.0973935
(dinI activation).g1	-0.0233436	(umuDC activation).g1	-0.0247963
(dinI activation).g2	-0.0990657	(umuDC activation).g2	-1.25224
(dinI activation).g3	-0.00295359	(umuDC activation).g3	1.259
(dinI activation).g4	0.242035	(umuDC activation).g4	0.092878
(dinI activation).g5	0.422833	(umuDC activation).g5	0.0159113
(dinI activation).g6	0.00149227	(umuDC activation).g6	1.31317
(dinI inact).k1	0.065681	(umuDC inact).k1	0.452944

Table 4.8: Parameter values for dinI and umuDC genes.

Parameter	value	Parameter	value
(rpoD activation).V	0.0612521	(rpoH activation).V	1.00E-05
(rpoD activation).g1	-0.0264438	(rpoH activation).g1	1.95E-07
(rpoD activation).g2	-0.466528	(rpoH inact).k1	0.00040883
(rpoD activation).g3	0.486808	(rpoS activation).V	1.00E-05
(rpoD activation).g4	-0.235035	(rpoS activation).g1	0
(rpoD activation).g5	0.26667	(rpoS inact).k1	0.00135737
(rpoD activation).g6	1.13764		
(rpoD activation).g7	0.218669		
(rpoD inact).k1	0.153078		

Table 4.9: Parameter values for rpoD, rpoH and rpoS genes.

Parameter	value	Parameter	value
(recF activation).V	3.52731	(recA activation).g2	-0.271949
(recF activation).g1	-1.22353	(recA activation).g3	0.208153
(recF activation).g2	-0.014753	(recA activation).g4	-0.0229387
(recF activation).g3	4.05E-06	(recA activation).g5	0.0242887
(recF activation).g4	1.25263	(recA activation).g6	0.384584
(recF inact).k1	0.302526	(recA inact).k1	0.159817
(recA activation).V	1.06809		
(recA activation).g1	-0.019806		

Table 4.10: Parameter values for gene recF

Gene	*dinI**	*lexA**	*recA**	*umuD**	*WT*
lexA	11.468867	12.87	11.384833	11.3117	11.47125
dinI	12.465967	10.042667	10.407267	10.476067	9.916875
umuD	8.038043	7.73134	8.03572	13.200167	7.819235
rpoD	10.545367	10.5738	10.620433	10.619833	9.46181
rpoH	9.45017	9.468853	9.41849	9.424997	8.59825
rpoS	10.8222	10.740933	10.577067	10.538466	10.3827
recF	9.619356	9.634883	9.67105	9.814993	8.974975
recA	11.714533	11.3685	13.254167	11.5945	11.79455
Ssb	10.544267	10.5773	10.641367	10.655033	10.21345

Table 4.11: Steady state values from upregulation experiments and wild type experiment.

Gene	*dinI**	*lexA**	*recA**	*umuD**
lexA	-0.00080817	1	-0.060873869	-0.020211234
dinI	1	0.10402766	0.399586678	0.081939418
umuD	0.108864996	-0.092187184	0.223720616	1
rpoD	0.445519655	0.96382304	0.989487667	0.177848302
rpoH	0.385458897	0.830387372	0.770856145	0.139723457
rpoS	0.164678879	0.282960575	0.151270655	0.021800634
recF	0.279317926	0.60300483	0.626708019	0.136007265
recA	-0.026393089	-0.296244188	1	-0.024646993
Ssb	0.126009904	0.292159866	0.338555237	0.062827046

Table 4.12: R values, comparison from experiments)

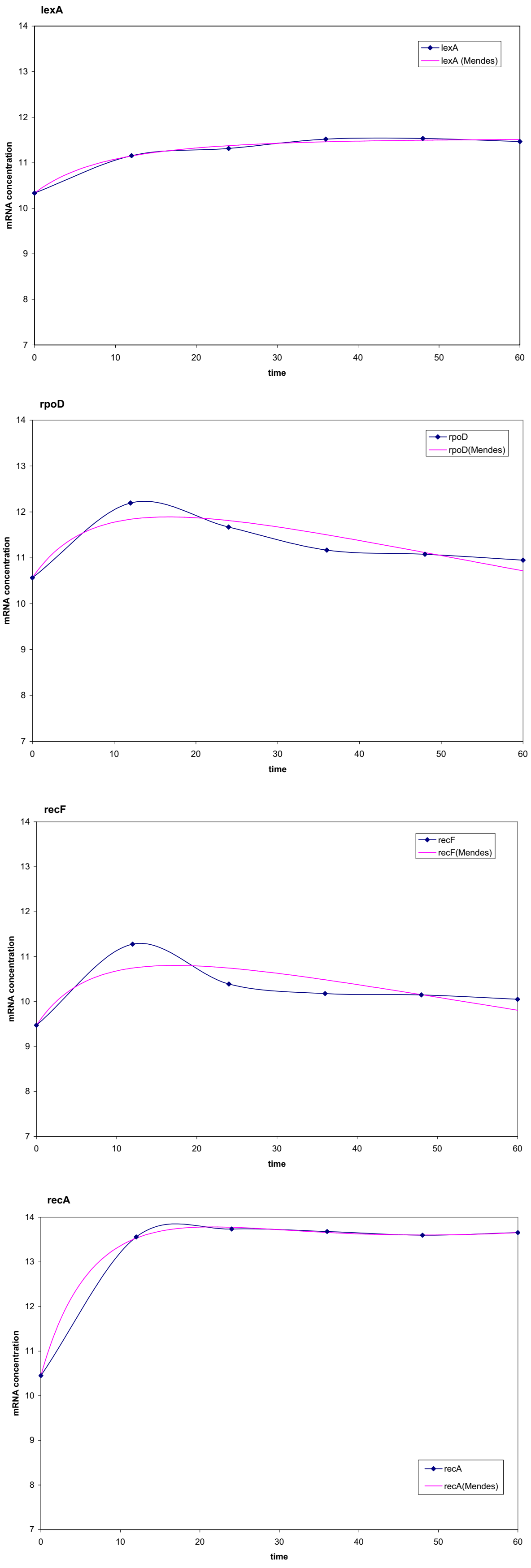

Figure 4.1: Curve fitting for each of *lexA*, *rpoD*, *recF*, *recA* genes using the Mendes model.

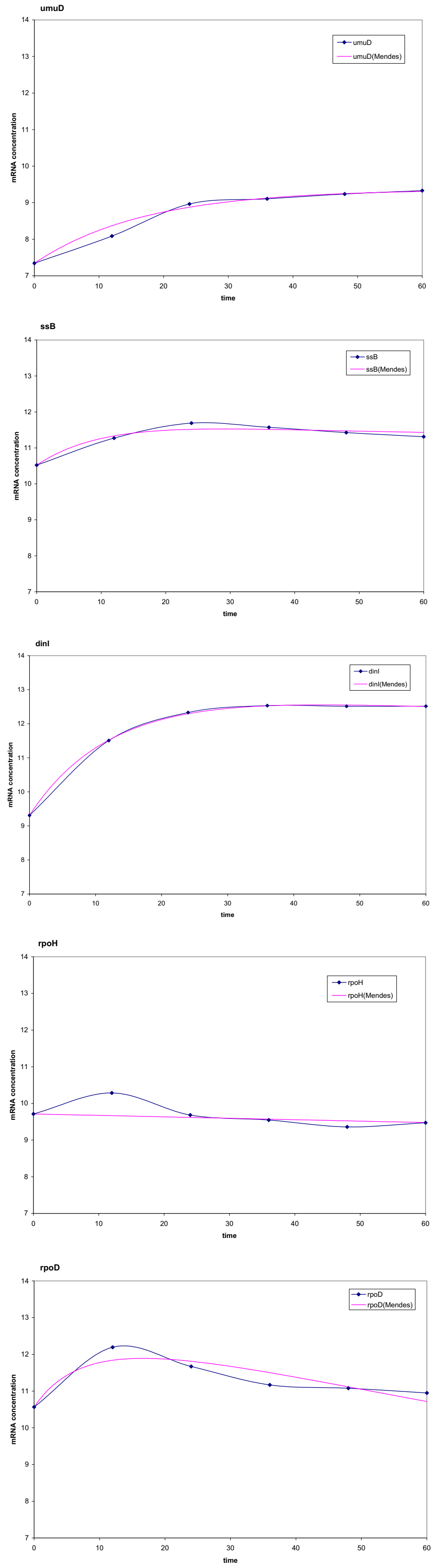

Figure 4.2: Curve fitting for *umuD, ssB, dinI, rpoH* genes using the Mendes model.

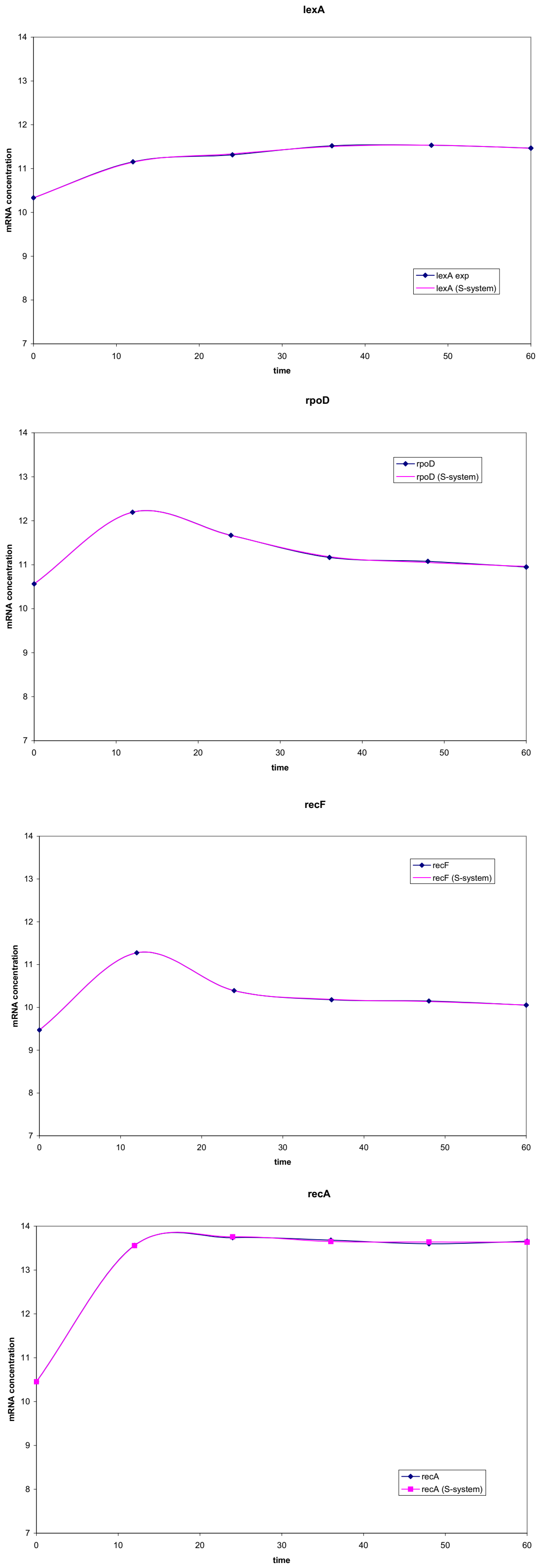

Figure 4.3: Curve fitting for each of *lexA*, *rpoD*, *recF*, *recA* genes using the S-system model.

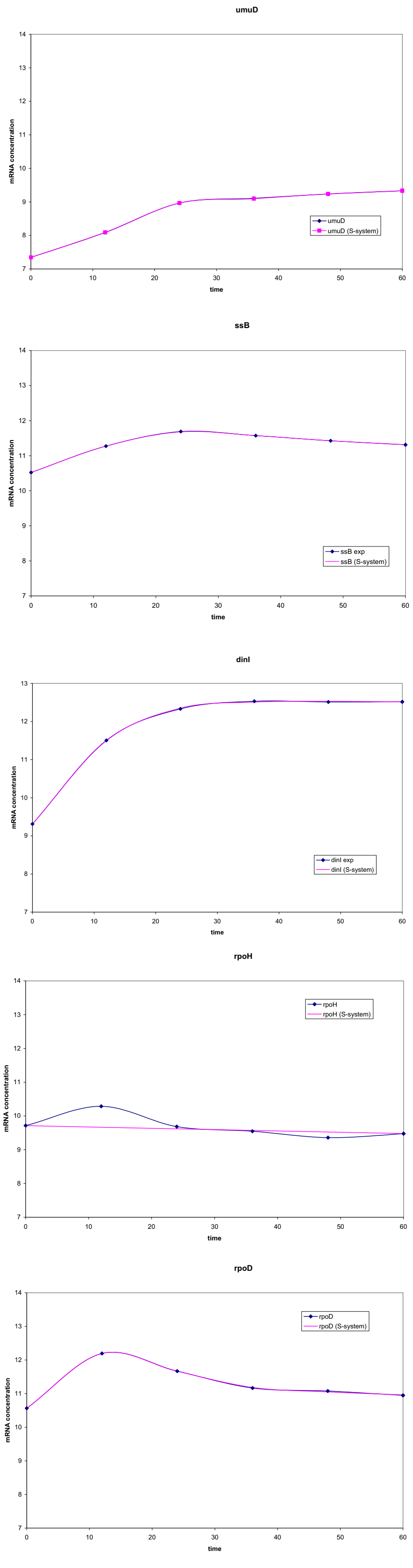

Figure 4.4: Curve fitting for *umuD, ssB, dinI, rpoH* genes using the S-system model.

Gene	$dinI^{*}fixed$	$lexA^{*}fixed$	$recA^{*}fixed$	$umuD^{*}fixed$	Mendes model
lexA	11.0713	12.85726603	11.4564	0.0718649	11.4599
dinI	15.25613749	11.6524	12.1608	5.3363	12.1365
umuD	9.39146	9.34954	9.35618	15.79426203	9.35587
rpoD	5.03237	5.18464	5.20549	0.0400115	5.20705
rpoH	1.12474	1.12474	1.12474	1.12474	1.12474
rpoS	3.46506	3.46681	3.46704	3.20536	3.46706
recF	4.69161	4.90577	4.87113	0.0259799	4.87275
recA	16.766	16.3642	18.53294294	164.806	16.492
Ssb	11.2236	11.2449	-2.22E-16	8.76072	11.2432

Table 4.13: Steady state values simulation of gene upregulation with Mendes model

Gene	$dinI$	$lexA$	$recA$	$umuD$
lexA	-1.32E-01	1	-0.002467912	-1.444025063
dinI	1	-0.327123894	0.016179134	-0.814207292
umuD	0.014799026	-0.005548686	0.000267744	1
rpoD	-0.130509091	-0.035295615	-0.002420891	-1.441971615
rpoH	0	0	0	0
rpoS	-0.002244181	-0.000591356	-4.66134E-05	-0.109685477
recF	-0.144620401	0.055574247	-0.002686478	-1.44539002
recA	6.46E-02	-0.063551846	1	13.06819432
Ssb	-0.006781964	0.001240023	-7.18708E-05	-0.320850399

Table 4.14: R values, comparison from Mendes model

Gene	$dinI^{*}fixed$	$lexA^{*}fixed$	$recA^{*}fixed$	$umuD^{*}fixed$	S-system model
lexA	0,111589	0.126065135	0.0950698	0.0650685	0.112364
dinI	1.349589613	1.07069	1.06848	1.10423	1.07362
umuD	20.9084	14.8154	17.2448	25.23217374	14.9465
rpoD	0.0477309	0.0462451	0.045975	0.0453676	0.0463255
rpoH	0.02446	0.02446	0.02446	0.02446	0.02446
rpoS	0.00736807	0.00736807	0.00736807	0.00736807	0.00736807
recF	0.202096	0.203942	0.202081	0.212593	0.203794
recA	5.59963	5.38502	6.065212362	5.57012	5.39728
Ssb	0.173815	0.173246	1.74E-01	0.166391	0.17333

Table 4.15: Steady state values simulation of gene upregulation with S-system model

Gene	$dinI^{*}fixed$	$lexA^{*}fixed$	$recA^{*}fixed$	$umuD^{*}fixed$
lexA	-2.68E-02	1	-0.277665798	-0.611644947
dinI	1	-0.022381422	-0.128541054	0.041430435
umuD	1.551795578	-0.071933937	0.030014927	1
rpoD	0.118023683	-0.014233332	-0.131319541	-0.030047395
rpoH	0	0	0	0
rpoS	0	0	0	0
recF	-0.032414223	0.005955812	-0.067921682	0.062740603
recA	1.46E-01	-0.018628855	1	0.046534609
Ssb	0.010885724	-0.003974442	0.041957658	-0.058174132

Table 4.16: R values, comparison from S-system model

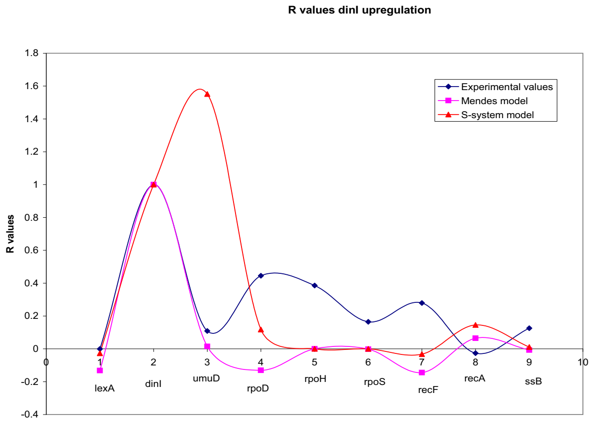

Figure 4.5: *R-values for dinI overexpression experiment.*

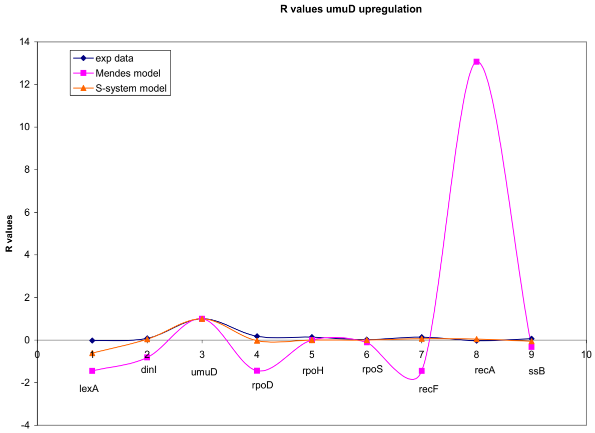

Figure 4.6: *R-values for umuD overexpression experiment.*

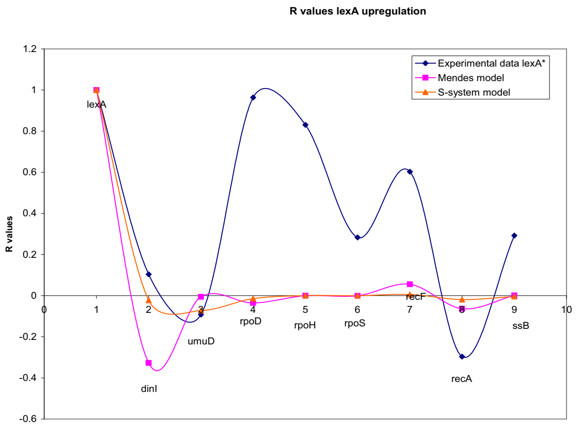

Figure 4.7: *R-values for lexA overexpression experiment.*

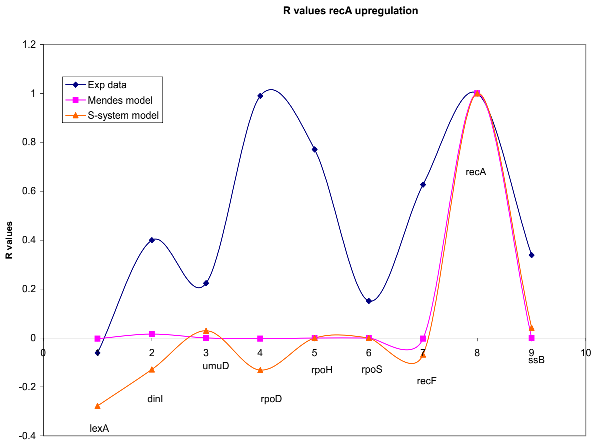

Figure 4.8: *R-values for recA overexpression experiment.*

Chapter 5

Conclusions and open problems

The Thesis presents four different models for a subnetwork comprising the principal mediators of the SOS response gene in *E. coli.* To construct the models we have used the gene network presented in [4] representing the interactions between nine genes of the SOS network. The goal of the thesis was to find a model able to predict the dynamics of the system. To achieve this we have used a time course dataset available in the M3D database that describes the behaviour of the nine genes in response to a perturbation with the antibiotic Norfloxacin. The perturbation with this antibiotic leads to the formation of single-stranded DNA and thus to the activaction of the SOS pathway via the recA protein. The main part of this work focuses on the estimation of the parameters, starting from the experimental data with the help of the optimization algorithms available in the software COPASI, in particular the global optimization algorithm particle swarm and Hooke and Jeeves.

Considering the general root mean square value, we have found out that Mendes model and S-system model where the two that produced the best fit.

Once obtained the parameters, we have used the models to predict the steady state concentrations in normal conditions and in overexpression experiments.

We have discovered that, although the S-system model gave the better fit between the two models, only the Mendes model was able to approximately retrieve the concentrations at the steady state at least for some of the genes considered in the network.

In particular the genes for which we have obtained predicted steady state concentrations distant from the experimental ones where *rpoD, rpoH, rpoS* and *recF.* Such similar result has been obtained also considering overexpression experiments.

The graphs presented in Figures 4.5, 4.6, 4.7 and 4.8, show the variance of the Co-response coefficient that indicates how two system variables respond to a common perturbation which in this case represents the overexpression of one of the genes in the network. Looking at these plots, we can see that the major differences between the R values calculated using the experiments and the ones calculated using the Mendes model present the major differences for the genes *rpoD, rpoH, rpoS* and *recF*, while the *R* values obtained using the S-system model simulations gave very different results from the experiments.

Focusing our attention on the Mendes model, we conjecture that we could not retrieve the right concentrations at the steady state for all the genes because of several problems: *(i)* just six data points are not enough to estimate the parameters of the models; *(ii)* the model is not appropriate to describe this network, probably not all the genes should have been modelled with the same equation. The wrong modeling of rpoH gene can probably have lead to a wrong estimation of the parameters for the genes that "influence" its regulation, as *rpoD, rpoS* and *recF*. If we look at the gene network we can see that rpoH gene is influenced only by rpoD, so maybe we should have added other genes in the regulation of its synthesis. In fact if we look at the parameter values estimated for this gene, we can see that they represent very low rate of synthesis. Due to this, even the R value for this gene it is always lower than the experimental one, in all the overexpression simulations and the same for the genes under its influence.

Acknowledgments

I would like to thank Alberto de la Fuente for his help and because he trusted me during the stage done in Cagliari at the CRS4 bioinformatics laboratory, giving me interesting tasks to achieve. I thank Matteo Cavaliere for his interesting lectures during the master and his useful suggestions during the writing of the thesis. And finally I thank Roberto and Valentina for making my stay in Trentino more pleasant.

Bibliography

[1] T. S. Gardner, D. di Bernardo, D. Lorenz, J. J. Collins. Inferring Genetic Networks and Identifying Compound Mode of Action via Expression Profiling. *Science*, 301, pages 102–105 (2003).

[2] S. Hoops, S. Sahle, R. Gauges, C. Lee, J. Pahle, N. Simus, M. Singhal, L. Xu, P. Mendes and U. Kummer. COPASIa COmplex PAthway SImulator. *Bioinformatics*, Vol 22, n 24, pages 3067-3074 (2006).

[3] J. J. Faith, M. E. Driscoll, V. A. Fusaro, E. J. Cosgrove, B. Hayete, F. S. Juhn, S. J. Schneider and T. S. Gardner. Many Microbe Microarrays Database: uniformly normalized Affymetrix compendia with structured experimental metadata. *Nucleic Acids Research*, pages 1–5, doi:10.1093/nar/gkm815, (2007).

[4] M. Bansal, G. Della Gatta and D. di Bernardo. Inference of gene regulatory networks and compound mode of action from time course gene expression profiles. *Bioinformatics*, Vol. 22 n 7, pages 815–822 (2006).

[5] L. Michaelis and M. L. Menten. Die Kinetik der Invertinwirkung. *Biochem. Z.*,49, pages 333–369 (1913).

[6] P. Mendes, W. Sha and K. Ye. Artificial gene networks for objective comparison of analysis algorithms. *Bioinformatics*, 19 Suppl. 2, pages ii122-ii129 (2003).

[7] A. V. Hill. The possible effect of the aggregation of the molecules of haemoglobin. *J. Physiol*, 40, iv–vii (1910).

[8] J. Stavans. The SOS response of bacteria to DNA damage. *Dynamics of Complex Interconnected Systems: Networks and Bioprocesses*, pages 39-47, Springer (2006).

[9] K. Schlacher and M. F. Goodman. Lessons from 50 years of SOS DNA-damage-induced mutagenesis. *Molecular Cell Biology*, pages 587-594, (2007).

[10] T. Nohmi, J. R. Battista, L. A. Dodson, and G. C. Walker. RecA-mediated cleavage activates umuD for mutagenesis: mechanistic relationship between transcriptional derepression and posttranslational activaction. *Proc. Natl Acad. Sci. USA*, 85, pages 1816-1820, (1988).

[11] H. Shinagawa, H. Iwasaki, T. Kato and A. Nakata. RecA protein-dependent cleavage of UmuD protein and SOS mutagenesis. *Proc. Natl Acad. Sci. USA*, 85, pages 1806–1810, (1988).

[12] S. Krishna, S. Maslov, K. Sneppen. UV-Induced Mutagenesis in *Escherichia coli* SOS Response: A Quantitative Model. *PLoS Comput Biol 3(3): e41. doi:10.1371/ journal.pcbi.0030041*, (2007).

[13] A. de la Fuente. Deciphering living networks Perturbation strategies for functional genomics. Phd thesis, Vrije Universiteit, (2006).

[14] R. Somogyi and L. D. Greller. The dynamics of molecular networks: applications to therapeutic discovery. *Drug Discov Today*, 6(24), pages 1267-1277 (2001).

[15] B. M. Bakker, P. A. M. Michelis, F. R. Opperdoes and H. V. Westerhoff. Glycolysis in bloodstream from Trypanasoma brucei can be understood in terms of the kinetics of the glycotic enzymes. *J. Biol. Chem.*, 272, 3207–3215, (1997).

[16] R. Eisenthal, and A. Cornish-Bowden. Prospects for antiparisitic drugs: the case of Trypanasoma brucei, the causative agent of African sleeping sickness. *J. Biol. Chem.*, 273, 5500–5505, (1998).

[17] M. F. Goodman, and B. Tippin. The expanding polymerase universe. *Nature Reviews, Molecular Cell Biology*, vol. 1, pages 101–109, (2000).

[18] D. H. Irvine and M. A. Savageau. Efficient solution of nonlinear ordinary differential-equations expressed in S-system canonical form. *Siam Journal on Numerical Analysis*, 27, pages 704–735, (1990).

[19] M. A. Savageau. Biochemical systems analysis. II. The steady-state solutions for an n-pool system using a power-law approximation. *J Theor Biol*, 25, pages 370–379, (1969).

[20] M. A. Savageau. Biochemical systems analysis. 3. Dynamic solutions using a power-law approximation. *J Theor Biol*, 26, pages 215–226, (1970).

[21] K. Levenberg. A method for the solution of certain nonlinear problems in least squares. *Quart. Appl. Math.*, 2, pages 164–168, (1944).

[22] D. W. Marquardt. An algorithm for least squares estimation of non-linear parameters. *SIAM Journal*, 11, pages 431–441, (1963).

[23] J. A. Nelder and R. Mead. A simplex method for function minimization. *Computer Journal*, 7, pages 308–313, (1965).

[24] R. Hooke and T. A. Jeeves. Direct search solution of numerical and statistical problems. *Journal of the Association of the Computing Machinery*, 8, pages 212–229, (1961).

[25] P. Mendes and B. K. Douglas. Non-linear optimization of biochemical pathways: applications to metabolic engineering and parameter estimation. *Bioinformatics*, Vol. 14, pages 869–883, (1998).

[26] P. Mendes. Modeling Large Biological Systems From Functional Genomic Data: Parameter Estimation. *Foundations of Systems Biology, H. Kitano (Ed) MIT Press, Cambridge, MA.* pages 163–186, (2001).

[27] C. Moles, P. Mendes and J. R. Banga. Parameter Estimation in Biochemical Pathways: A Comparison of Global Optimization Methods. *Genome Research.*, 13, pages 2467–2474, (2003).

[28] D. E. Goldberg. Genetic algorithms in search, optimization and machine learning. *Addison Wesley Longman,London*, (1989).

[29] D. B. Fogel. Evolutionary computation: Toward a new philosophy of machine intelligence. *IEEE Press, New York*, (1966).

[30] H. P. Schwefel. Evolution and optimum seeking. *Wiley, New York*, (1995).

[31] J. Kennedy and R. C. Eberhart. Swarm Intelligence. *Morgan Kaufmann Publishers*, (2001).

[32] S. Kirkpatrick, C. D. Gellatt and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220, pages 671–680, (1983).

[33] J. J. Faith, B. Hayete, J. T. Thaden, I. Mogno, J. Wierzbowski, G. Cottarel, S. Kasif, J. J. Collins and T. S. Gardner. Large-Scale Mapping and Validation of Escheichia coli Transcriptional Regulation from a Compendium of Expression Profiles. *PLoS Biology*, vol. 5, issue 1, e8, pages 54–66, (2007).

[34] D. H. Wolpert and W. G. Macready. No Free Lunch Theorems for Optimization. *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION*, vol. 1, n 1, pages 67–82, (1997).