



UNIVERSITÀ DEGLI STUDI DI PADOVA

SCUOLA DI INGEGNERIA

Corso di Laurea Magistrale in Bioingegneria

**SENSITIVITY ANALYSIS METHODS FOR
DYNAMICAL MODELS**

Laureanda

Elena Bergamo

Relatore

Prof.ssa Gianna M. Toffolo

Co-relatore

Dr. Alberto Giaretta

ANNO ACCADEMICO 2015/2016

*“If you only read the books that everyone else is reading,
you can only think what everyone else is thinking.”*

Haruki Murakami, Norwegian Wood

Contents

1	Background and motivations	1
1.1	Local sensitivities	2
1.2	Aim of the thesis	4
2	Sensitivity analysis methods	5
2.1	Indirect or brute force method	6
2.2	Direct method	6
2.3	Green's function method	8
2.4	Impulse Parametric Sensitivity Analysis (iPSA)	9
2.4.1	iPSA: Implementation	12
2.5	Sensitivity metrics	13
3	The model of HPV early promoter regulation	15
3.1	The HPV-16 genome and life cycle	16
3.2	The model	18
3.2.1	Model equations	20
3.2.2	Model predictions	23
4	Sensitivity analysis: MATLAB implementation	25
4.1	Indirect method	26
4.2	Direct method	26
4.3	iPSA	27
4.3.1	Numerical approach	27
4.3.2	Analytical approach	28
5	Results	31
5.1	Comparison between DQM and DM	31
5.2	PSA results with DM	37
5.2.1	Main viral DNA replication regulator E_1	37
5.2.2	Main transcriptional regulator E_2	41
5.2.3	The primary polycistronic transcript X	46

5.2.4	mRNA mE_1 encoding for protein E_1	51
5.2.5	mRNA mE_2 encoding for protein E_2	55
5.3	Impulse parametric sensitivity analysis	59
5.4	iPSA results with numerical approach	65
5.5	iPSA results with analytical approach	71
6	Conclusions	75
	Acronyms	77
	Appendices	79
A	Green's function method derivation	81
B	MATLAB code implementations	85
B.1	Difference quotient method	87
B.2	Direct method	87
B.3	iPSA	89
B.3.1	Numerical approach	89
B.3.2	Analytical approach	91
	Bibliography	97

Abstract

Sensitivity analysis methods are widely applied in many fields of engineering to study biological systems, including metabolic networks, signalling pathways and genetic circuits. Sensitivity analysis investigates the effect of parameter change on the solution of mathematical models in order to deduce which parameters constitute the controlling factor of the system and which are, on the contrary, insignificant for the model behavior. The applications of such analysis are many, including model reduction and parameter estimation. In this work, conventional Parametric Sensitivity Analysis (PSA) main methods (Direct, Indirect and Green's Function methods) were reviewed along with the novel analysis called Impulse Parametric Sensitivity Analysis (iPSA) presented by Perumal et al. The differences between the two analysis were investigated by their application on the HPV early promoter regulation model presented by Giaretta et al. Results were compared and each method was analyzed from the reliability of the results and computational complexity points of view. The comparison between Direct and Indirect method suggested to select the first method for the model under study, reaching the most correct results while maintaining a low computational complexity. Moreover conventional PSA produced results that were in accordance with biological knowledge, confirming the importance of the parameters belonging to the transactivation/repression feedback of protein E_2 in the transcription of the primary transcript. iPSA, besides showing again the importance of parameters linked to transcriptional regulation of E_2 , revealed important behaviors of the model, like its oscillatory nature under impulsive stimulus and the presence of a highly parametric sensitive region located in a specific time range.

Chapter 1

Background and motivations

The increasing number of applications of biological and chemical models in all areas of science and engineering has made Parametric Sensitivity Analysis (PSA) a powerful tool to provide valuable insights about the robustness and reliability of the system.

PSA is widely applied in many systems biology applications including metabolic networks, signalling pathways and genetic circuits to investigate the effect of parameter changes on the model variables of interest.

In fact the behavior of a system depends on the values of the parameters that characterize it and the importance of a parameter as controlling factor can be assessed by analyzing how the system responds to changes in that specific parameter. The more the systems outputs change after parameter perturbation, the more the system is sensitive to that particular parameter. In other words the parameter has a high sensitivity. Sensitivity analysis is, therefore, used to find the key factors that affect the model outputs of a system but has additional widely used applications such as uncertainty analysis, parametric scaling, parameter estimation, experimental design, stability analysis, repro-modeling, and investigation and reduction of complex reaction mechanisms. In many biochemical systems, if one or more parameters are varied slightly, while holding the remaining parameters fixed, the response also changes slightly but it can happen that, under another set of parameter combinations, the system reacts with a strong change after a slight variation. In this case, the system is said to behave in a parametrically sensitive manner. This kind of systems are very difficult to control, therefore it is important to identify the parametrically sensitive region of operation in order to define the boundary between runaway (explosive) and nonrunaway (nonexplosive) behavior [1].

There are many reasons why modelers may conduct sensitivity analysis such as the necessity to determine: (1) which parameters are insignificant and can

be eliminated from the final model (model reduction); (2) which parameters require additional research; (3) which inputs contribute most to output variability; (4) which parameters are most highly correlated with the output; and (5) while simulating, what is the consequence of a given input parameter change [2].

Sensitivity analysis can be performed in different ways and it is important to notice that the various analysis may produce different results [3]. In most applications, sensitivity coefficients or metrics are computed and generally reflect the ratios between the change in a biological model output and the perturbation on system parameters that cause the change.

Sensitivity analyses can be classified as local or global depending on the magnitude of the perturbations: the former refers to an infinitesimal perturbation, the latter to a finite perturbation. In both cases, the interpretation of the sensitivity metrics is intuitive; parameters with large sensitivity magnitude are considered to be important in the system functioning and to be the controlling factors in the system regulation.

1.1 Local sensitivities

As explained above, parametric sensitivity constitutes the analysis of how a system responds to changes in the parameters.

Sensitivity analysis is performed on the mathematical description of the system which is provided by a model, that describes the system through explicit or implicit relationships between the state or output values and the input parameters. Output values are in general indicated as time and/or space dependent variables and, in the case of biochemical model, often refer to model state concentrations. The input parameters include the biochemical parameters of the model (related to reaction kinetics, thermodynamic equilibria, and transport properties) but also operating conditions, and geometric parameters of the system. Parameter values are measured experimentally or estimated theoretically from the available literature and therefore they are always subject to uncertainties.

Let's consider the kinetics of a spatially homogeneous system modeled as

$$\frac{dx}{dt} = f(\mathbf{x}, \mathbf{p}), \quad \mathbf{x}(0) = \mathbf{x}^0, \quad (1.1)$$

where \mathbf{x} is the n-vector of concentrations and \mathbf{p} is the m-vector of system

parameters (initial concentrations are not considered in vector \mathbf{p}). The solutions of the system of equations (1.1) are concentration-time curves and in this case the sensitivity analysis gives concentration sensitivity as result. Local sensitivity are computed considering small changes of parameter values. Their effect on the solution can be expressed by a Taylor series expansion:

$$x_i(t, \mathbf{p} + \Delta\mathbf{p}) = x_i(t, \mathbf{p}) + \sum_{j=1}^m \frac{\partial x_i}{\partial p_j} \Delta p_j + \frac{1}{2} \sum_{l=1}^m \sum_{j=1}^m \frac{\partial^2 x_i}{\partial p_l \cdot \partial p_j} \Delta p_l \Delta p_j + \dots \quad (1.2)$$

where $\frac{\partial x_i}{\partial p_j}$ are called *first-order local concentration sensitivity coefficients*, while $\frac{\partial^2 x_i}{\partial p_l \cdot \partial p_j}$ are called *second-order local concentration sensitivity coefficients*. Normally only first order coefficients are studied and they constitute the *sensitivity matrix* \mathbf{S} .

Sensitivity matrices have a double time dependence $S_{i,j}(t, \tau) = \frac{\partial x_i(t)}{\partial p_j(\tau)}$ with $t > \tau$ where τ is the perturbation time and t the observation time. Usually $\tau = t(0) = 0$ and the term τ is commonly dropped out of the equation but, depending on the case, the particular application might need to compute sensitivity with perturbation time different from zero.

The implementation of *first-order local concentration sensitivity coefficients* can be obtained through different methods, among which:

- **The *brute force or indirect method***: the simplest way of calculating local concentration sensitivities through the finite different approximation (Difference Quotient).
- **The *direct method***: the sensitivity coefficient are calculated by directly differentiating the ODE model.
- **The *Green's function method***: a more sophisticated method to solve the differential equations of the *direct method*.
- Polynomial approximation method.
- etc...

All methods have different computation complexity and the method of choice is selected depending on the application and the model under consideration.

Conventional local sensitivity methods are all characterized by the introduction of permanent perturbation of parameter values but it was demonstrated by Perumal et. al [4] that this kind of analysis only partially reflects the dynamical behavior of the system and this may lead to incorrect results that

don't match with experimental evidences.

In order to overcome this issue, Perumal et al. introduced a novel type of sensitivity analysis based on impulsive perturbation rather than persistent perturbation, called Impulse Parametric Sensitivity Analysis (iPSA). Sensitivity coefficients calculated with this analysis are not directly comparable to the ones calculated with regular PSA but provide independent information about the dynamical importance of each parameter.

1.2 Aim of the thesis

In this work three of the most famous Local Sensitivity Analysis (LSA) methods are reviewed in Chapter 2: Indirect method with difference quotient approximation (Different Quotient Method (DQM)), Direct Method (DM) and Green's Function Method (GFM) which is presented in detail in Appendix A. They are also compared to the novel type of local PSA, the iPSA introduced by Perumal et. al. The different sensitivity methods are mathematically analyzed and their derivation is explained, in relation to a general ODE model. The previously mentioned methods are then applied to a novel model of HPV early promoter regulation developed by Giaretta et al. [5]. In Chapter 3, the biological model presented in the article "Modeling HPV early promoter regulation" is introduced, together with a review of Human Papillomavirus (HPV) gene expression and regulation. In Chapter 4 the PSA methods are considered from their computational implementation point of view in the software MATLAB (code implementation details in Appendix B), in Chapter 5 the results of PSA analysis are presented and compared in order to evidentiate the differences of the various methods. Finally in Chapter 6 the conclusions and the study future perspectives are discussed.

Chapter 2

Sensitivity analysis methods

Parametric Sensitivity Analysis (PSA) is one of the most powerful tool for the study of ODE models and it is well established in the field of engineering. In particular, sensitivity analysis investigates the effect of parameter change on the solution of mathematical models.

Sensitivity analysis is divided into two types according to the magnitude of the perturbation: local (LSA) and global analysis. The first one assesses an infinitesimal perturbation, the second one a finite perturbation. In both cases, the interpretation of the PSA results is simple: the parameters that have large sensitivity magnitude are considered to be driving and controlling factors in the functioning of the system.

As previously mentioned in Chapter 1, in this study four methods of LSA are reviewed: indirect, direct, Green's function method and the novel local analysis iPSA.

The generic nonlinear ODE model is described by the following equations

$$\begin{cases} \frac{d\mathbf{x}(t, \hat{\mathbf{p}})}{dt} = \mathbf{f}(\mathbf{x}, \hat{\mathbf{p}}) \\ \mathbf{x}(t_0, \hat{\mathbf{p}}) = \mathbf{x}^0 \end{cases} \quad (2.1)$$

where \mathbf{x} is the n-vector of concentrations ($x \in \mathbb{R}^n$) and \mathbf{p} is the m-vector of system parameters ($\hat{\mathbf{p}} \in \mathbb{R}^m$) with initial conditions $\mathbf{x}(t_0, \hat{\mathbf{p}}) = \mathbf{x}^0$. The effect of parameter perturbations is described by the first order partial derivatives of the state variables that represent the first-order coefficients:

$$S_{i,j}(t, \tau) = \frac{\partial x_i(t)}{\partial p_j(\tau)} \quad (2.2)$$

where τ is the time in which the perturbation is applied and t is the consequent observation time ($t > \tau$). In conventional PSA the perturbation can

be introduced at any time but it's usually set at the initial time $\tau = t_0 = 0$ (with initial conditions $\mathbf{x}(t_0) = \mathbf{x}_0 = 0$) and for that reason the term τ is often dropped out of the equations, but in the following it will be maintained. In order to compare the sensitivity coefficients they are usually normalized as follows:

$$\bar{S}_{i,j}(t, \tau) = \frac{\partial x_i(t)}{\partial p_j(\tau)} \frac{p_j(\tau)}{x_i(t)} = \frac{\partial \log x_i(t)}{\partial \log p_j(\tau)} \quad (2.3)$$

2.1 Indirect or brute force method

The most simple method to solve (2.2) is with the use of the finite difference approximation:

$$S_{i,j}(t, \tau) = \frac{\partial x_i(t)}{\partial p_j(\tau)} = \frac{\text{change in } i\text{-th state at time } t}{\text{perturbation on } j\text{-th parameter at time } \tau} \quad (2.4)$$

where $j = 1, \dots, m$ and the perturbation amplitude is the same for each τ . In this technique, called the brute force method or the indirect method, sensitivity coefficients are calculated changing the j -th parameter at time t by a variation depending on the parameter nominal value, while all other parameters are held fixed. This method requires the solution of n differential equations of the ODE model using the nominal value of parameters and $m \cdot n$ solutions of the equations using perturbed parameter sets, hence it needs to solve $(m + 1)n$ differential equations.

The indirect method is widely used because the code implementation is the simplest, however, this method is not recommended because there are methods that calculate sensitivity coefficients consuming much less computer time, like Direct method or Green's function method.

2.2 Direct method

An alternative way to compute the sensitivity coefficients is by differentiating the ODE model equations in what is called the Direct method (**DM**). The sensitivities of the solutions with respect to the system parameters are calculated by solving the following ODE system:

$$\frac{d}{dt} \frac{d\mathbf{x}(t)}{d\mathbf{p}(\tau)} = \frac{\partial f(t)}{\partial \mathbf{x}(t)} \frac{d\mathbf{x}(t)}{d\mathbf{p}(\tau)} + \frac{\partial f(t)}{\partial \mathbf{p}(\tau)} \quad (2.5)$$

where $\frac{\partial f(t)}{\partial \mathbf{x}(t)}$ is the Jacobian. Equation (2.5) can be written in matricial form as follows:

$$\frac{d}{dt} \mathbf{S}(t, \tau) = \frac{\partial f(t)}{\partial \mathbf{x}(t)} \mathbf{S}(t, \tau) + \frac{\partial f(t)}{\partial \mathbf{p}(\tau)} \quad (2.6)$$

where $S(t, \tau)$ is the sensitivity $n \times m$ matrix

$$S(t, \tau) = \begin{bmatrix} \frac{\partial x_1(t)}{\partial p_1(\tau)} & \frac{\partial x_1(t)}{\partial p_2(\tau)} & \cdots & \frac{\partial x_1(t)}{\partial p_m(\tau)} \\ \frac{\partial x_2(t)}{\partial p_1(\tau)} & \frac{\partial x_2(t)}{\partial p_2(\tau)} & \cdots & \frac{\partial x_2(t)}{\partial p_m(\tau)} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial x_n(t)}{\partial p_1(\tau)} & \frac{\partial x_n(t)}{\partial p_2(\tau)} & \cdots & \frac{\partial x_n(t)}{\partial p_m(\tau)} \end{bmatrix}, \quad (2.7)$$

$\frac{\partial f(t)}{\partial x(t)}$ is the $n \times n$ Jacobian matrix

$$\frac{\partial f(t)}{\partial x(t)} = J(t) = \begin{bmatrix} \frac{\partial f_1(t)}{\partial x_1(t)} & \frac{\partial f_1(t)}{\partial x_2(t)} & \cdots & \frac{\partial f_1(t)}{\partial x_n(t)} \\ \frac{\partial f_2(t)}{\partial x_1(t)} & \frac{\partial f_2(t)}{\partial x_2(t)} & \cdots & \frac{\partial f_2(t)}{\partial x_n(t)} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial f_n(t)}{\partial x_1(t)} & \frac{\partial f_n(t)}{\partial x_2(t)} & \cdots & \frac{\partial f_n(t)}{\partial x_n(t)} \end{bmatrix} \quad (2.8)$$

and $\frac{\partial f(t)}{\partial \mathbf{p}(\tau)}$ is $n \times m$ non homogeneous term matrix

$$\frac{\partial f(t)}{\partial \mathbf{p}(\tau)} = \begin{bmatrix} \frac{\partial f_1(t)}{\partial p_1(\tau)} & \frac{\partial f_1(t)}{\partial p_2(\tau)} & \cdots & \frac{\partial f_1(t)}{\partial p_m(\tau)} \\ \frac{\partial f_2(t)}{\partial p_1(\tau)} & \frac{\partial f_2(t)}{\partial p_2(\tau)} & \cdots & \frac{\partial f_2(t)}{\partial p_m(\tau)} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial f_n(t)}{\partial p_1(\tau)} & \frac{\partial f_n(t)}{\partial p_2(\tau)} & \cdots & \frac{\partial f_n(t)}{\partial p_m(\tau)} \end{bmatrix}. \quad (2.9)$$

Like in the Indirect method, the sensitivity coefficients are then usually normalized according to Equation 2.3.

Direct method can be approached in many ways, depending on the number of equations that are solved simultaneously. Here it is a list of possible approaches:

(1) Equations (2.1) and eq. (2.5) are solved simultaneously for all $j = 1, \dots, m$ and this requires $(m + 1)n$ differential equations. This approach requires an algorithm in which the ode solver is able to perform the decomposition of the $(m + 1)n \cdot (m + 1)n$ Jacobian in each step, hence the direct solution of this large system of differential equations is inefficient. However it was

demonstrated that a fast algorithm can be built exploiting the fact that this large Jacobian has an almost block-diagonal structure [6] [7].

(2) It can result convenient to subdivide the problem into m sub-problems, one for each parameter. This approach consists in integrating the m systems of $2n$ equations containing the couple of eq. (2.1) and eq. (2.5) together for each specific parameter. The approach requires the solution of $2n \cdot m$ differential equations, this version is very simple but is the least economical and in some cases can cause numerical difficulties.

(3) Equations (2.1) are solved firstly and the concentration-time curves are stored, then eq. (2.5) is applied and in case a solution at a time that is not stored is needed, it will be obtained through interpolation.

In the DM the faster approach needs to solve a $(m + 1) \cdot n$ differential equations system and since the equations of interest are usually stiff and m ($> n$) is in general very large, the method can be very expensive to compute. In this situation the Green's Function Method can provide a better way to reach the solutions.

2.3 Green's function method

Looking at Equation 2.5 and Equation 2.6, one can see that the m linear sensitivity equations differ only in the nonhomogeneous terms, $\frac{\partial f(t)}{\partial p_j(\tau)}$. For this reason in the Green Function method (GFM) the sensitivity equations solutions are constructed by first computing the solution of the homogeneous part and then by determining the particular solution corresponding to each input parameter. The homogeneous part is obtained by differentiating Equation 2.1 with respect to the state vector $\mathbf{x}(\tau)$ and correspond to the following Green's function problem [7]:

$$\frac{d}{dt}\mathbf{S}^{\mathbf{x}}(t, \tau) = \frac{\partial f}{\partial \mathbf{x}}\mathbf{S}^{\mathbf{x}}(t, \tau) \quad , t > \tau \quad (2.10)$$

where $\mathbf{S}^{\mathbf{x}}(t, \tau)$ is a $n \times n$ matrix called Green's function matrix (GFm) defined as

$$S_{i,j}^{\mathbf{x}}(t, \tau) = \frac{\partial x_i(t)}{\partial x_j(\tau)}, \quad \mathbf{S}^{\mathbf{x}}(\tau, \tau) = \mathbf{I}. \quad (2.11)$$

Then, the local sensitivities, that are obtained fixing the j -th column of the Green's Function Matrix (GFm) $S_{[1..n],j}(t, \tau) = \frac{\partial \mathbf{x}(t)}{\partial p_j(\tau)}$, can be written in terms of a linear integral transform of the nonhomogeneous terms [1], using the Gfm

(details in Appendix A):

$$S_{[1..n],j}(t, \tau) = \frac{\partial \mathbf{x}(t)}{\partial p_j(\tau)} = \int_{\tau}^t \mathbf{S}^{\mathbf{x}}(t, s) \frac{\partial f(s)}{\partial p_j} ds \quad (2.12)$$

. When the perturbation is set at the initial time, that is $\tau = t_0$, Equation 2.12 becomes:

$$S_{[1..n],j}(t, t_0) = \frac{\partial \mathbf{x}(t)}{\partial p_j} = \mathbf{S}^{\mathbf{x}}(t, t_0) \cdot \frac{d\mathbf{x}(t_0)}{dp_j} + \int_{t_0}^t \mathbf{S}^{\mathbf{x}}(t, s) \frac{\partial f(s)}{\partial p_j} ds \quad (2.13)$$

where

$$\frac{d\mathbf{x}(t_0)}{dp_j} = \begin{cases} \delta_i & p_j \text{ is an initial condition for one of the components } x_i \\ 0 & \text{otherwise} \end{cases} \quad (2.14)$$

where δ_i is the vector $\delta_i = [0, 0, ..1, 0, ..0]$ with 1 at the i -th position [8]. In the integral of eqn.(2.12), the first argument of $\mathbf{S}^{\mathbf{x}}(t, s)$, t , is fixed, whereas the second one varies.

Let's now compare the numerical complexity of GFM and DM in the case of n dependent variables and m input parameters. The GFM requires solving $n \cdot n$ differential equations plus n , thus $(n + 1) \cdot n$, integrals, while the DM involves solving $(m + 1) \cdot n$ differential equations. Since for most applications, the number of parameters exceeds the number of dependent variables, the GFM requires less numerical effort relative to the DM. Obviously, if the sensitivities of the n dependent variables with respect to only a few of the m input parameters are needed, the DM may be preferred over the GFM. The derivation of the equations of the Green's Function Method is presented in Appendix A.

2.4 Impulse Parametric Sensitivity Analysis (iPSA)

In a dynamic system, parameters are expected to vary with time. Local parametric sensitivities reflect changes in the system states when the parameters are persistently perturbed. Hence, there could be some discrepancies between the interpretation of the influence of system parameters suggested by the PSA and the experimental findings. The reason can be understood taking a look at the formula of Equation 2.6 expressing parametric sensitivity coefficients of PSA with the direct method, here re-written

$$\frac{d}{dt} \mathbf{S}(t, \tau) = \frac{\partial f(t)}{\partial \mathbf{x}(t)} \mathbf{S}(t, \tau) + \frac{\partial f(t)}{\partial \mathbf{p}(\tau)}$$

Two terms contribute to the sensitivity coefficient: the first is related to an indirect effect, propagated through the changes in the states \mathbf{x} , while the second accounts for the direct effect of parametric perturbation on the function $f(t, \mathbf{x}(t, \mathbf{p}))$.

Consequently in this kind of analysis normally what happens is that, perturbing parameters at the initial time t_0 , the parameters with high sensitivity at early times are favored and usually have high sensitivity throughout the simulation period. This can lead to misleading revelations on the dynamical importance of system parameters.

To overcome these issues Perumal et al. presented a novel type of sensitivity analysis that should be able to reflect dynamical aspects of the system behavior. This analysis is based on impulse rather than persistent perturbation and is called Impulse Parametric Sensitivity Analysis (iPSA) [4]. The corresponding sensitivity coefficient $iS_{i,j}(t, \tau)$ reflects the change in the i -th state x_i at time t due to an impulse perturbation on the j -th parameter p_j at time τ , thus iPSA coefficient $iS_{i,j}(t, \tau)$ depends on both the time of perturbation τ and the observation time t ($t > \tau$). Since impulse perturbations cause an immediate state changes at τ , the dynamical importance of the system parameters can be assessed from iPSA coefficients simply varying the time of perturbation, maintaining t fixed.

iPSA coefficients are defined as follows:

$$iS_{i,j}(t, \tau) = S_{i,[1..n]}^x(t, \tau) \frac{\partial f}{\partial p_j}(\tau) \quad (2.15)$$

where $S^x(t, \tau)$ is the Green's Function matrix defined in Equation 2.11 and $S_{i,[1..n]}^x(t, \tau)$ accounts for the i -th row of this matrix while $\frac{\partial f}{\partial p_j}(\tau)$ accounts for the nonhomogeneous term of DM equation (Equation 2.6) calculated in τ .

The new sensitivity coefficient presents two properties:

- If the sensitivity is evaluated in $t = \tau$, then $S_{i,j}^x(\tau, \tau) = \mathbf{I}$ and Equation 2.15 becomes

$$t = \tau : iS_{i,j}(\tau, \tau) = \frac{\partial f}{\partial p_j}(\tau) \quad (2.16)$$

-

$$t > \tau : iS_{i,j}(t, \tau) = S_{[1..n],j}^x(t, \tau) iS_{i,j}(\tau, \tau) \quad (2.17)$$

From eqn (2.17) one can see that the consequence on the state trajectory is equivalent to perturbing the states themselves, like in the GFM analysis because the impulse parameter perturbation takes effect only at time τ . In

order to compare sensitivity results and for parameter ranking purpose the iPSA coefficient need to be normalized as follows:

$$i\bar{S}_{i,j}(t, \tau) = iS_{i,j}(t, \tau) \frac{p_j}{x_i(t)} \quad (2.18)$$

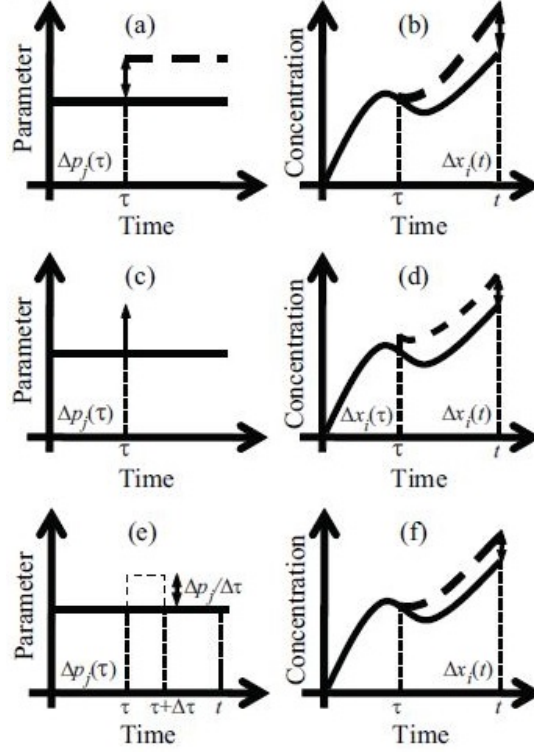


Figure 2.1: Parametric perturbation in sensitivity analysis and its effect on system dynamics in (a-b) PSA, (c-d) iPSA and (e-f) pulse approximation of iPSA. Solid lines represent the nominal and the dashed lines represent the perturbed trajectory. The figure is taken from [4].

In Figure 2.1 a graphical representation of the conventional local PSA (Figure 2.1(a-b)) is depicted in comparison with local iPSA (Figure 2.1(c-d)) and its pulse approximation (Figure 2.1(e-f)).

A persistent parametric perturbation leads to a different effect on the system behavior compared to an impulsive perturbations hence sensitivity analysis results can differ and produce different interpretations. It is important to notice that iPSA coefficients are not directly comparable to the once obtained from the conventional PSA methods. Their interpretation, in respect to the other methods, is explained in Chapter 5: Results.

2.4.1 iPSA: Implementation

Each iPSA coefficient is constructed by approximating an impulsive perturbation and quantifying the ratio between the change in i -th state x_i at time t and the causative pulse perturbation applied on the j -th parameter p_j at time τ of size $\frac{\Delta p_j}{\Delta \tau}$ for a duration of Δt in the limit Δp_j and $\Delta \tau$ tending to zero. The first step of the construction of iPSA coefficient is to define the change in the system states \mathbf{x} at the end of the parameter perturbation at time $\tau + \Delta \tau$ using the Taylor series expansion:

$$\Delta x(\tau + \Delta \tau) = S_{[1..n],j}(\tau + \Delta \tau, \tau) \frac{\Delta p_j}{\Delta \tau} + O(\Delta p_j^2) \quad (2.19)$$

where $S_{[1..n],j}$ represents the j -th column of the sensitivity matrix introduced in Equation 2.6. The change $\Delta x(\tau + \Delta \tau)$ is then translated in the change in the state x_i at time t using the GFm $\mathbf{S}^x(t, \tau + \Delta \tau)$ which contains the sensitivities of the state x_i to a perturbation in the state x_j at time τ as presented in Equation 2.11 :

$$S_{i,j}^x(t, \tau) = \frac{\partial x_i(t)}{\partial x_j(\tau)} \quad (2.20)$$

where $t \geq \tau$. The (i,j) -th element of the GFm quantify the relative change in the state x_i at time t due to an impulsive alteration in the state x_j at time τ given by

$$x_j + \delta(t - \tau) \Delta x_j \quad (2.21)$$

where the magnitude Δx_j tends to zero and $\delta(t)$ is the dirac delta function. Using the GFm the change $\Delta x_i(t)$ due to the pulse perturbation is given by

$$\Delta x_i(t) = S_{i,[1..n]}^x(t, \tau + \Delta \tau) \Delta x(\tau + \Delta \tau) \quad (2.22)$$

Substituing eqn. (2.19) in eqn. (2.22) it becomes

$$\Delta x_i(t) = S_{i,[1..n]}^x(t, \tau + \Delta \tau) S_{[1..n],j}(\tau + \Delta \tau) \frac{\Delta p_j}{\Delta \tau} + O(\Delta p_j^2) \quad (2.23)$$

The next step consists in taking the Taylor series expansion of the sensitivities at time τ and dividing both members of the equation for Δp_j :

$$\Delta x_i(t) = S_{i,[1..n]}^x(t, \tau + \Delta \tau) \left[S_{[1..n],j}(\tau, \tau) + \dot{S}_{[1..n],j}(\tau, \tau) + O(\Delta \tau^2) \right] \frac{\Delta p_j}{\Delta \tau} + O(\Delta p_j^2)$$

$$\frac{\Delta x_i(t)}{\Delta p_j} = S_{i,[1..n]}^x(t, \tau + \Delta \tau) \left[\dot{S}_{[1..n],j}(\tau, \tau) + O(\Delta \tau^2) \right] + O(\Delta p_j) \quad (2.24)$$

$$\frac{\Delta x_i(t)}{\Delta p_j} = S_{i,[1..n]}^x(t, \tau + \Delta\tau) \left[\frac{\partial f}{\partial p_j} + O(\Delta\tau^2) \right] + O(\Delta p_j)$$

iPSA coefficient are finally obtained by taking the limit as $\Delta p_j, \Delta\tau \rightarrow 0$ so the pulse perturbation becomes an impulse leading to

$$iS_{i,j}(t, \tau) = S_{i,[1..n]}^x(t, \tau) \frac{\partial f}{\partial p_j}(\tau) \quad (2.25)$$

2.5 Sensitivity metrics

Once first-order sensitivity coefficients are calculated, parameter ranking is commonly generated. In this work three sensitivity metrics were implemented for parameter ranking:

$$\begin{aligned} [S_{int}]_{i,j} &= \int_{t_0}^t \bar{S}_{i,j}(\hat{t}) d\hat{t}, \\ [S_{inf}]_{i,j} &= \max_k (|S_{i,j}(t_k)|) \\ [S_{t_k}]_{i,j} &= S_{i,j}(t = t_k) \end{aligned} \quad (2.26)$$

where the indices i and j denote the i -th state and j -th parameter. S_{int} , S_{inf} and S_{t_k} are the sensitivity metrics based on time integral [9], infinite norm [10] and sensitivity magnitude at a particular time, respectively.

Chapter 3

The model of HPV early promoter regulation

Human papillomaviruses (HPVs) are a type of DNA virus from Papillomaviridae family. Most HPV infections will cause no physical symptoms; however, a persistent infection is considered to be the main causative agent of cervical cancer and other anogenital cancer [11]. Over 120 HPV types have been characterized by full genome sequencing to date [12] but only some of them are high risk types, like HPV 16 and 18 that are estimated to cause 70 percent of the cervical cancers. More specifically, HPVs are classified according to the malignancy of infections that they cause: low-risk virus (LR-HPV) that produce benign lesions (like HPV-6 and HPV-11 that lead to genital and laryngeal warts) and high risk viruses (HR-HPV) that cause precancerous lesions and cancer (HPV types 16, 18, 31). Despite their differences, they all show a high degree of cellular tropism, in fact all types of HPVs only infect epithelial tissue, although the nature of epithelia can differ from dried or cutaneous squamous epithelium (skin) and mucous membranes (oral and genital) [13] [14]. Each year, about 33,000 new cases of cancer are found in parts of the body where HPV is often found and it is estimated to be the cause of about 26,900 of these cancers [15]; it is clearly of paramount importance to improve our knowledge about the mechanisms that regulate the infection stages. All human papillomaviruses have a similar genome organization, life cycle and regulation, but since HPV-16 is the most widely studied because it is by far the most common cancer-associated HPV type [16], from now on the word HPV will be used to refer to HPV-16.

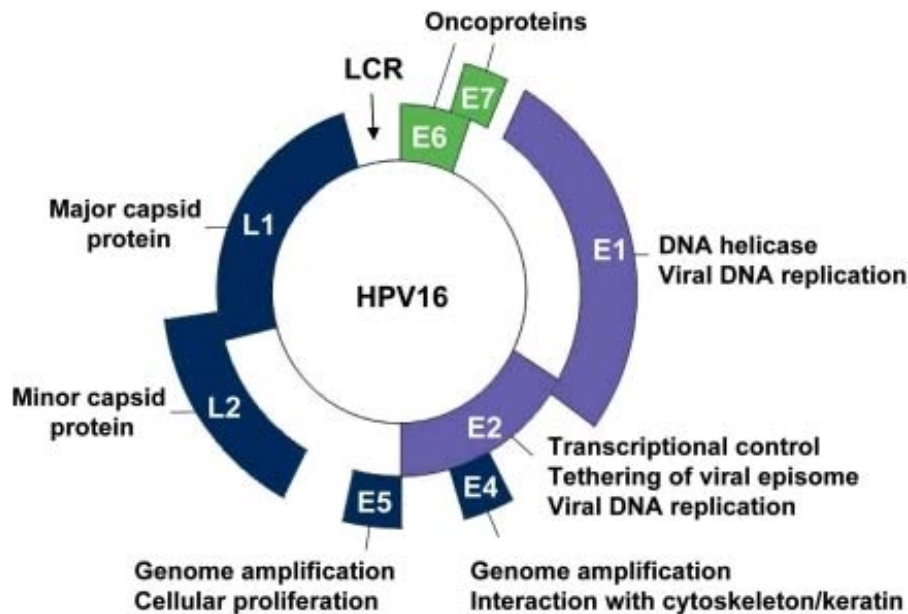


Figure 3.1: Schematic representation of the HPV-16 genome

3.1 The HPV-16 genome and life cycle

The DNA of HPV is circular, double stranded, and contains approximately 8 kilo base pairs (Kbp). Its genome can be divided into three main areas: the long control region (LCR), the early genes (E_1 , E_2 , E_4 , E_5 , E_6 , E_7), and the late genes (L_1 and L_2).

The LCR region corresponds to 15 percent of viral genome and does not contain genes but is the location of important regulatory elements such as the early promoter P97, which is very active during replication. The virus genome contains also a late differentiation-regulated promoter P670 located within the E_7 open reading frame [17].

Both promoters are polycistronic, hence produce polycistronic pre-mRNAs that are subjected to alternative splicing through the use of various early splice sites.

The early region represents 45 percent of the genome and contains the 7 early genes, which encode for non-structural homonym proteins, each with different functions. E_6 and E_7 are the main oncogenes and their expression is fundamental to induce cervical cancer. E_1 is the main responsible for DNA replication, in cooperation with E_2 , bringing together the episomal viral genome and the cellular DNA polymerase. E_2 is the main transcriptional regulator: it becomes active by binding with the early promoter and has both

the functions of transcription promoter and suppressor when present in low and high concentrations respectively [5].

E_4 produces proteins involved in the maturation of the viral particles, it is thought to facilitate virion release into the environment by disrupting intermediate filaments of the keratinocyte cytoskeleton. Moreover it is believed that E_4 facilitates genome amplification. E_5 is believed to have role in cellular proliferation and genome amplification.

The late region corresponds to 40 per cent of the genome and contains two genes that encode for structural proteins from the capsid: L1 and L2.

HPV life cycle (Figure 3.2) is strictly linked to the differentiation stage of the infected cells (keratinocytes) and requires a continuity of tissues in order to be able to penetrate and initiate an infectious process. For that reason infection starts in the basal layer of the epithelium.

In the early stages of the viral life cycle the promoter starts the replication with the help of E_1 and E_2 , as the infected cell divides and migrates to the upper layers E_6 and E_7 stimulate cell growth in order to continue replication and increase the viral early genes expression. After, thanks to high level of E_2 , the early promoter is inhibited and the cell downregulates the expression of E_6 and E_7 allowing the infected cell to differentiate. This is the beginning of the differentiation-dependent stage of HPV life cycle that is turned on by high levels of E_4 , which is believed to promote the release of progeny virions. At the end the capsid proteins L_1 and L_2 are produced and assemble the progeny virions at the upper layer of epithelium: the superficial zone constituted of squamous epithelium [5] [14] [16].

Different antiviral strategies against HPV high risk types have been proposed through the years targeting different stages of the virus life cycle. One therapy can target the actors of the late stages, namely the oncogenes or directly the late genes L_1 and L_2 .

Another strategy can target replication and transcription using compounds against E_1 and E_2 . Moreover it was discovered recently that E_1 has both the action of positive regulation of E_2 transcripts and of protein stabilization up to an half-life of 6 hours. This means that E_1 could be able to allow E_2 to reach the high levels that characterize its differentiation levels that turn down the early promoter. Since these new discoveries about E_1 and E_2 co-regulation result very interesting in the optic of creating new and efficient therapy, a novel model of HPV regulatory mechanisms from a qualitative/quantitative point of view was presented in the article "Modeling HPV early promoter regulation" by Giaretta et al. [5].

The next Section has the aim of introducing the model.

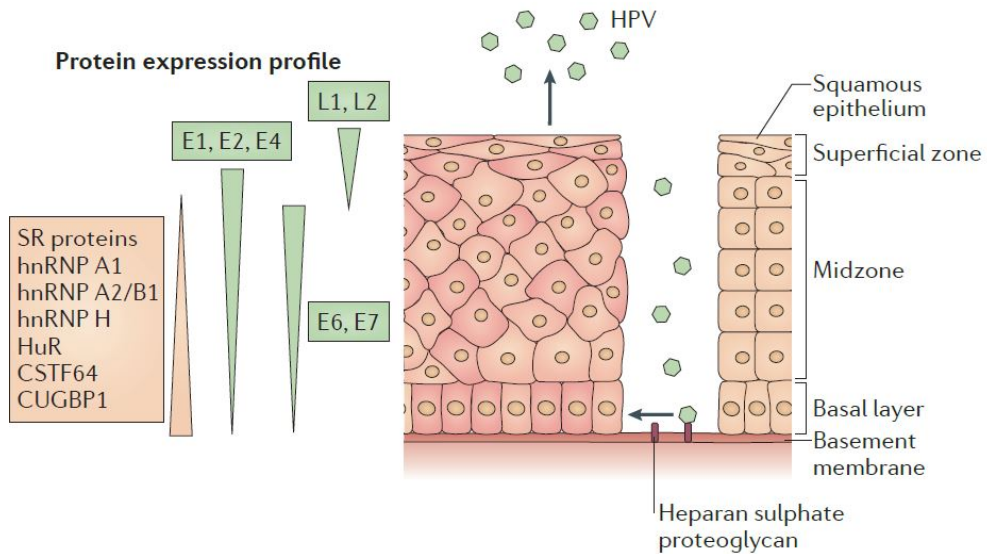


Figure 3.2: The Human Papillomavirus 16 life cycle. HPV-16 life cycle starts in the basal layer of the epithelium where it enters thanks to small wounds or abrasions and attaches via heparan sulphate proteoglycans. With the activation of the early promoter, replication is initiated with the mediation of E_1 and E_2 early proteins. Cellular splicing factors (serine-arginine-rich (SR) proteins), heterogeneous nuclear ribonucleoproteins (hnRNPs), cleavage stimulator factors (CSTF) and others are highly expressed in the early and middle phases but shut down at late stages. As the infected cell differentiates, cell growth is stimulated by E_6 and E_7 . Their expression is then shut down by high levels of E_2 . High levels of E_4 facilitates the release of progeny virions and the cell reaches the upper layer of the epithelium where late proteins L_1 and L_2 assemble the capsid constituting the ultimate progeny virions.

3.2 The model

The model presented in [5] is focused on E_1 and E_2 regulation on the HPV (in episomal form) early viral stage and collects and structures the available knowledge on the early promoter regulation .

The mechanisms of HPV early gene circuit is depicted in Figure 3.3.

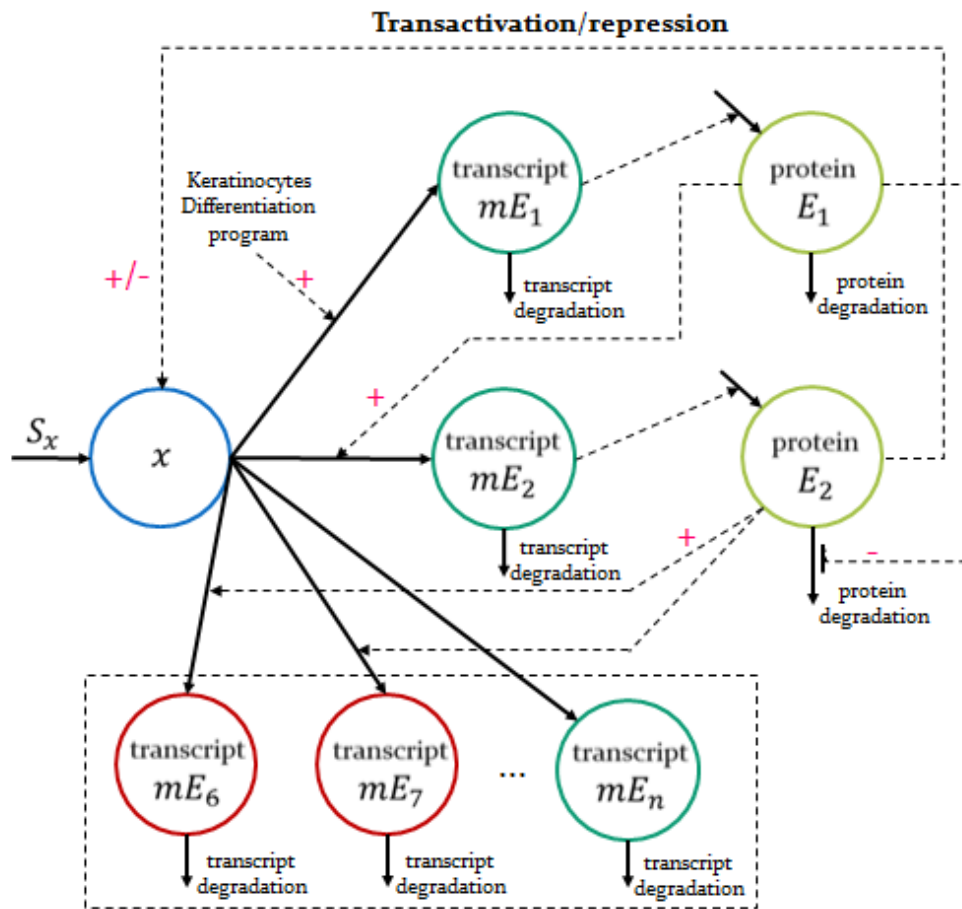


Figure 3.3: Diagram of the HPV-16 gene circuit. Solid arrows represent fluxes while dashed arrows the controls. In blue are represented the primary transcripts and in green the transcripts with their respective non-oncogenic proteins while in red the early oncogenes.

The primary polycistronic transcript x is controlled by the early promoter. When x is transcribed, it generates, by alternative splicing, mRNAs encoding all the early genes: in particular the model focuses on the spliced mRNA E_2 (mE_2) that encodes for E_2 protein and the spliced mRNA E_1 (mE_1) that encodes for E_1 protein. E_2 is the main regulator of the early promoter x act as an activator (positive feedback effect), when present in low concentrations, or a suppressor, when present in high concentration. E_1 act both with a positive feedback enhancing the mE_2 transcript and with a negative regulation on E_2 degradation, increasing E_2 stability.

3.2.1 Model equations

Model equations are the following:

$$\dot{x} = S_x(E_2) - k_s x \quad (3.1)$$

$$m\dot{E}_1 = k_{1s}(t)k_s x - \delta_{1m}mE_1 \quad (3.2)$$

$$m\dot{E}_2 = k_{2s}(E_1)k_s x - \delta_{2m}mE_2 \quad (3.3)$$

$$\dot{E}_1 = \beta_1 mE_1 - \delta_{1p}E_1 \quad (3.4)$$

$$\dot{E}_2 = \beta_2 mE_2 - \delta_{2p}(E_1)E_2 \quad (3.5)$$

$$m\dot{E}_i = k_{is}x - \delta_{im}mE_i \quad i = 4, 5, 6, 7 \quad (3.6)$$

where the state variables are the primary transcript x , mRNAs mE_i , $i = 1, 2, 4, 5, 6, 7$, proteins E_1 and E_2 . Protein translation is assumed to be linearly related to mRNA concentrations and its rate constants for E_1 and E_2 are β_1 and β_2 . δ_{im} and δ_{ip} are degradations of transcripts and proteins respectively and are first order processes with exception of δ_{2p} (see (3.12)). S_x is a function that represent the transcription of x and depends on E_2 concentration. Low concentrations of E_2 enhances the transcription while high concentrations have the effect of inhibiting transcription, according to the formula:

$$S_x(E_2) = \begin{cases} S_b + \frac{a_1 E_2^{q_1}}{\lambda_1^{q_1} + E_2^{q_1}}, & E_2 < E_2^{th} \\ \frac{a_2 \lambda_2^{q_2}}{\lambda_2^{q_2} + (E_2 - E_2^{th})^{q_2}}, & E_2 > E_2^{th} \end{cases} \quad (3.7)$$

For the continuity of the function, a_2 is parameter dependent, following the expression:

$$a_2 = S_b + \frac{a_1 (E_2^{th})^{q_1}}{\lambda_1^{q_1} + (E_2^{th})^{q_1}} \quad (3.8)$$

k_{1s} is modeled as a time variant forcing function and represents the splicing flux for the mE_1 transcript. As the life cycle of HPV is strictly linked to the host cell differentiation, also k_{1s} function is bounded to this differentiation and the variable t is the time evolution associated to it.

$$k_{1s}(t) = \begin{cases} k_{1s}^{min} + \frac{a_3 t^{q_3}}{\lambda_3^{q_3} + t^{q_3}}, & t < t_h \\ \frac{a_4 \lambda_4^{q_4}}{\lambda_4^{q_4} + (t - t_h)^{q_4}}, & t > t_h \end{cases} \quad (3.9)$$

Like a_2 , also a_4 is parameter dependent for the continuity of the function, following the expression:

$$a_4 = k_{1s}^{min} + \frac{a_3 t_h^{q_3}}{\lambda_3^{q_3} + t_h^{q_3}} \quad (3.10)$$

On the other hand k_{2s} accounts for the splicing flux for the mE_2 transcript and depends on E_1 concentration.

$$k_{2s}(E_1) = \frac{(f_1 - 1)k_{2s}^{min}}{1 + \exp\left(\frac{\lambda_5 - E_1}{\sigma_1}\right)} + k_{2s}^{min} \quad (3.11)$$

δ_{2p} refers to the degradation of E_2 , negatively regulated by E_1 according to

$$\delta_{2p}(E_1) = (f_2 - 1)\delta_2^{min} \left(1 - \frac{1}{1 + \exp\left(\frac{\lambda_6 - E_1}{\sigma_2}\right)} \right) + \delta_2^{min} \quad (3.12)$$

The meaning and nominal values of model parameters inferred from the available literature are summarized in Table 3.1, taken from [5].

22 CHAPTER 3. THE MODEL OF HPV EARLY PROMOTER REGULATION

Par	Description	Value	u.m.
S_b	Basal transcription synthesis of the primary transcript x	10	[nM/min]
k_s	Nucleo cytoplasmatic transport	0.0693	[-]
a_1	Transactivation constant of x	1.6	[nM/min]
E_2^{th}	E_2 concentration threshold of transactivation/repression	5	[nM]
λ_1	x concentration for half-maximal x transactivation	2.83	[nM]
q_1	Hill coefficient of E_2 positive feedback	2	[-]
a_2	Repression constant of x	11.21	[nM/min]
λ_2	x concentration for half-maximal x repression	2	[nM]
q_2	Hill coefficient of E_2 negative feedback	2	[-]
a_3	Saturating value of mE_1 transcription rate during early phase	12.34	[1/min]
λ_3	Half-maximal of mE_1 transcription rate during early phase	$3.76 \cdot 10^4$	[min]
q_3	Hill coefficient of mE_1 transcription rate during early phase	4	[-]
a_4	Initial value of mE_1 transcription rate during late phase	0.38	[1/min]
λ_4	Half-maximal of mE_1 transcription rate during late phase	$5 \cdot 10^3$	[min]
q_4	Hill coefficient of mE_1 transcription rate during late phase	2	[-]
t_h	Time threshold of mE_1 transcription rate	12	[days]
f_1	Fold increase of mE_2 transcription rate	10	[-]
k_{1s}^{min}	Basal mE_1 transcription rate	$9 \cdot 10^{-3}$	[1/min]
k_{2s}^{min}	Basal mE_2 transcription rate without E_1 positive regulation	$9 \cdot 10^{-3}$	[1/min]
λ_5	Half-maximal of mE_2 transcription rate	9	[nM]
σ_1	Slope parameter of mE_2 transcription rate	0.7	[nM]
λ_6	Half-maximal of E_2 degradation rate fixed to obtain mE_2 enhancement	13	[nM]
σ_2	Slope parameter of E_2 degradation rate	0.7	[nM]
δ_2^{min}	Basal E_2 degradation rate without E_1 negative regulation	0.0019	[1/min]
f_2	Fold decrease of E_2 degradation rate	4	[-]
δ_{1m}	mE_1 degradation rate	$2.9 \cdot 10^{-2}$	[1/min]
δ_{2m}	mE_2 degradation rate	$2.9 \cdot 10^{-2}$	[1/min]
δ_{1p}	E_1 degradation rate	$5.7 \cdot 10^{-3}$	[1/min]
β_1	E_1 translation rate	$1.5 \cdot 10^{-2}$	[1/min]
β_2	E_2 translation rate	$2 \cdot 10^{-2}$	[1/min]

Table 3.1: Table of the model parameters. First column contains the names of the parameters; second column shows their meanings; third column contains their nominal values; the last one presents the units of measure.

3.2.2 Model predictions

Model equations were implemented in MATLAB in order to assess the trends of the model state variables (Figure 3.4), the simulation time is set from 0 to 14 days (early stages).

The initial phase appears very fast and, more or less 6 hours after infection, every variables reach a steady state. This behavior agrees with experimental results that state the presence of the early transcripts between 4-10 hours after infection. The primary transcript x shows a slow switching off around 8-12 days post infection and this is consistent with literature, it is also important to note that the only effect of E_2 transactivation/repression is not sufficient to turn off the early promoter, but also E_1 effect is needed.

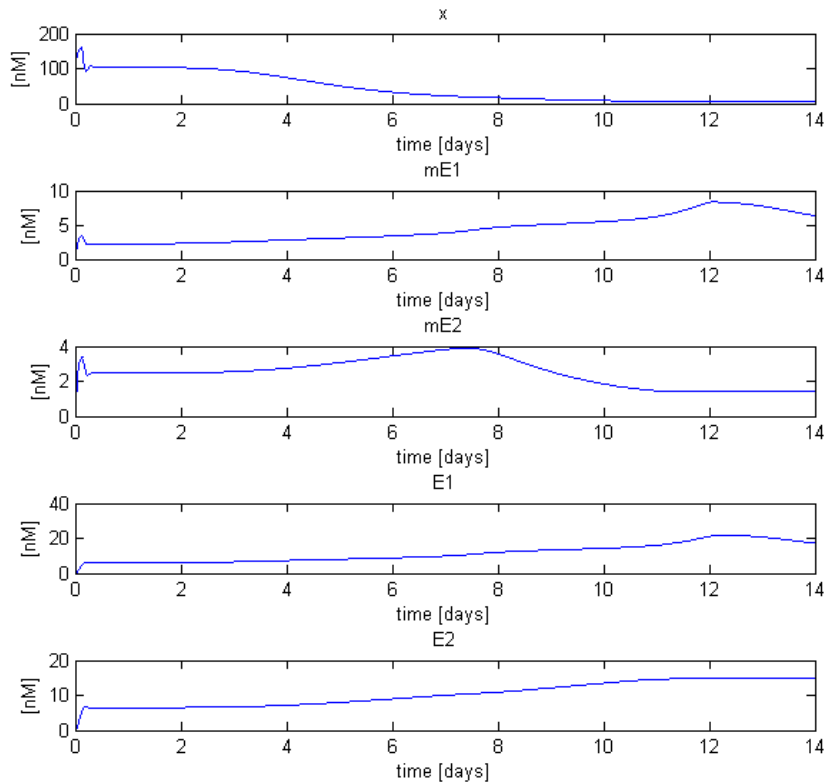


Figure 3.4: Curve trends of the model state variables.

Synthesis of the primary transcript, rate constant for mE_1 splicing, E_2 degradation mediated by E_1 , rate constant for mE_2 splicing mediated by E_1

trends are depicted in Figure 3.5.

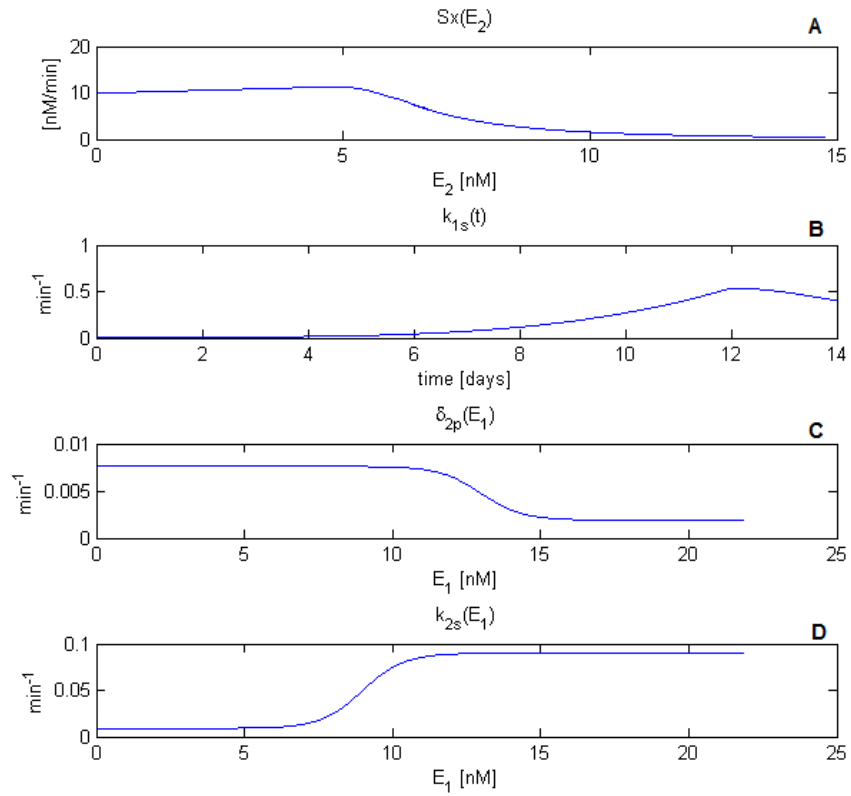


Figure 3.5: (A) Synthesis of the primary transcript depending on E_2 concentration. (B) Rate constant for mE_1 splicing. (C) E_2 degradation mediated by E_1 . (D) Rate constant for mE_2 splicing mediated by E_1 .

Chapter 4

Sensitivity analysis: MATLAB implementation

In the present work, local sensitivity analysis is performed on the model of Chapter 3 with the use of the software MATLAB. The methods that were implemented are: the Indirect method or Difference Quotient method (DQM), the Direct method (DM) and Impulse Parametric Sensitivity Analysis (iPSA). For the case of conventional PSA, sensitivity coefficients were calculated assuming the time of perturbation corresponding to $t_0 = 0$, hence the sensitivity coefficient have only one time dependence on the observation time t . On the other hand, for iPSA the definition presented in the previous chapter is maintained. Since the matlab code implementation requires some degrees of approximation and discretization that have to be considered for the reliability of the results, in this Chapter the approximations introduced in the code implementation of each method are described, along with the issues that were faced during implementation and running of the code.

PSA analysis is first restricted to the state variables E_1 and E_2 , the two proteins, in order to select the best approach between DM and DQM.

These two methods are compared based on computational complexity and results reliability. iPSA was first implemented at the beginning for protein E_2 with both numerical and analytical approach. The sensitivities of system parameters were calculated for all the state variables and the metrics of eq. (2.26) were performed for conventional PSA, in order to select the most conditioning parameters for every variable. On the other hand iPSA was applied only to state variables x , mE_2 and E_2 and for the parameters that were considered the most interesting in PSA analysis results and from the biological point of view. Results and comparison of conventional PSA and iPSA are presented in Chapter 5. Moreover code implementations are reported in Appendix B for the specific example of the parameter q_2 and

state variable E_2 .

4.1 Indirect method

Considering a generic prediction of the state variables depending on model parameters and initial conditions $y(t) = f(p_1, p_2, \dots, p_m, t)$, the first-order coefficients can be written for Matlab implementation as **difference quotient** :

$$\bar{S}_{i,j}(t) = \frac{f(p_1, \dots, p_j + \Delta p_j, \dots, p_m, t) - f(p_1, \dots, p_j, \dots, p_m, t)}{\Delta p_j} \frac{p_j}{y(t)} \quad (4.1)$$

For this reason the Indirect method is named from now on **Difference Quotient method (DQM)**.

DQM was implemented with the following magnitude of parameter perturbation $\Delta p_j = 1\%, 5\%, 10\%$ of p_j nominal values (for nominal values see Table 3.1).

This choice was made with the aim of discriminating what degree of approximation can be accepted: in order to complete this task the results need to be compared with the ones of the direct method that directly calculates the sensitivity coefficients with, in theory, the smallest approximations.

In the calculation of sensitivity coefficients with DQM is necessary to know the concentration-time curves of the state variables with the non perturbed parameters and to calculate concentration-time curves of the state variables with the perturbed value of the specific parameter under consideration. Both cases involve the use of a ordinary differential equations solver and ODE45 MATLAB function was used. The non-perturbed and perturbed trajectories are then subtracted and divided for the magnitude of perturbation (see Appendix section B.1).

4.2 Direct method

The Direct method was applied on the state variables using approach (2) described in section 2.2. This approach consists in the subdivision of the sensitivity problem in m sub-problems, one for each parameter obtaining m systems of $2n$ equations, n for eq. (2.1) and n for eq. (2.5). This approach requires the solution of $2n \cdot m$ differential equations, this version is very simple but is the least economical and in some cases can cause numerical difficulties. Even if approach (2) is the least efficient and economical, in the case of the

model of Chapter 3 this is not relevant because the model does not present a large number of equations and the software manages to run the code without any problems of computation time. Only an ODE function, containing $2n = 2 \cdot 5 = 10$ differential equations, is needed for each parameter both for the calculation of state variables concentration-time curves and for the calculation of equation (2.5).

Again the ODE solver that was used for the simulation is ODE45, the obtained values were then interpolated using the function "*interp1*".

It is important to notice that the Jacobian matrix, implemented within the ODE functions, remains the same for each parameter.

Matlab code is presented in Appendix B.2 for the example of state variable E_2 with respect to q_2 , even if each parameter requires its own function to implement the calculus of equation (2.5) since $\partial f / \partial p_j$ is specific for each parameter.

4.3 iPSA

As mentioned above, iPSA was implemented with both numerical and analytical approaches. In particular, numerical approach consists in applying the iPSA coefficient definition which states that the iPSA coefficients quantify the ratio between the change in the state x_i at time t and the perturbation of size $\frac{\Delta p_j}{\Delta \tau}$ for a duration of Δt applied to the parameter p_j at time τ with Δp_j and $\Delta \tau$ tending to zero. Analytical approach consists in applying the formula of equation (2.15) which can be done with different implementations, two of them are presented in the following. Both matlab implementations result to be computationally heavy giving the need to produce huge size matrices of data with different ode solver applications.

4.3.1 Numerical approach

Numerical approach needs to produce an impulsive perturbation at each τ ($0 \leq \tau \leq t$) to the nominal value of each parameter in order to quantify the ratio between the perturbed and non-perturbed state trajectories. The perturbation was introduced through a gaussian approximation of an impulse using the matlab function *gaussmf*($x, [\sigma \ c]$). *gaussmf* creates a gaussian function in the form

$$f(x; \sigma, c) = e^{\frac{-(x-c)^2}{2*\sigma^2}}; \quad (4.2)$$

and returns a matrix which is the Gaussian membership function evaluated at x . $[\sigma \ c]$ is a 2-element vector, containing the mean (c) and the standard deviation (σ), that determines the position and the shape of the function. The

gaussian perturbation is added to the nominal value of the parameter and needs to be produced inside the ODE solver that calculates the perturbed trajectories; for this reason a different ODE function was created for each parameter in order to perturb one parameter at a time. The level of perturbation that was set for this analysis is $\Delta p_j = 0.01 * p_j$ (1 % of p_j nominal value). Once perturbed trajectories are calculated, the (normalized) iPSA coefficients are obtained as follows:

$$iS_{i,j}(t, \tau) = \frac{x_i^\Delta(t) - x_i(t)}{\Delta p_j} \cdot \frac{p_j}{x_i(t)} \quad (4.3)$$

where $x_i^\Delta(t)$ is the i-th perturbed state trajectory consequent to a perturbation of the j-th parameter p_j at time τ and Δp_j is the perturbation magnitude.

4.3.2 Analytical approach

The implementation of iPSA coefficients through the analytical approach requires the calculation of the elements of the Green's Function matrix (GFm). For Matlab implementation the (i,j)-th of the GFm element was calculated as:

$$S_{i,j}^x(t, \tau) = \frac{\partial x_i(t)}{\partial x_j(\tau)} \cong \frac{\Delta x_i(t)}{\Delta x_j(\tau)} = \frac{x_i^\Delta(t) - x_i(t)}{\Delta x_j(\tau)} \quad (4.4)$$

Where $x_i^\Delta(t)$ is the i-th perturbed state trajectory consequent to a perturbation of the j-th state x_j at time τ .

In this work the perturbation of the system states was set as $\Delta x_j = 0.01 \max(x_j)$ (1% of x_j maximum value) and the same perturbation magnitude was used at different perturbation times τ . The perturbation was introduced at every τ through the initial conditions; in other words the initial conditions were changed at every cycle of computation adding Δx_j to the nominal value of $x_j(\tau)$, for example the vector of initial conditions for introducing the perturbation in E_2 is $[x(\tau); mE_1(\tau); mE_2(\tau); E_1(\tau); E_2(\tau) + \Delta E_2]$. The perturbed trajectories were obtained attaching the non-perturbed part of the signal to the perturbed part, going from τ to the end of simulation time. An alternative way to introduce the perturbation can be the use of the previously mentioned function *gaussmf*, added to the state variable nominal values inside the model equation of the ODE solver. However this alternative makes use of an impulse approximation thus the previous method is to be considered the most correct, for this reason the alternative using gaussian perturbation will no longer be investigated. In order to generate the GFm it is necessary to generate the trajectories of every single state variable consequent to the

perturbation on only one state variable, one at a time. Once GFm is generated, (normalized) iPSA coefficients are obtained applying the formula of equation (2.15):

$$iS_{i,j}(t, \tau) = S_{i,[1..n]}^x(t, \tau) \frac{\partial f}{\partial p_j}(\tau) \cdot \frac{p_j}{x_i(t)}$$

It is important to notice that iPSA coefficients calculated with this approach cannot be directly compared to the ones calculated with numerical approach since the perturbation is applied on the state variables and on the parameters respectively. A normalization between values [1;-1] needs to be applied in order to compare the final results.

Chapter 5

Results

As previously mentioned, in the case of conventional PSA, each method was first implemented on E_1 and E_2 state variables only in order to select the most efficient PSA method to use for the other state variables. Similarly, iPSA was firstly implemented on E_2 in order to reach the best MATLAB implementation and in this Chapter the final results are presented, also for the other state variables.

5.1 Comparison between DQM and DM

DQM and DM represent two ways to calculate the sensitivity coefficients, but their results depend on approximations, for DM of the ODE matlab function integration step and options and for DQM for the Δp_j set in the calculation. Theoretically the DM results should be considered the most accurate but, on the other hand, the DM is computationally more heavy than the DQM since the DM (with approach (2)) involves $2n \cdot m$ differential equations while the DQM involves only $(m + 1)n$ differential equations (usually $m > n$).

The two methods were applied on E_1 and E_2 and in particular the DQM was implemented with three different magnitude of perturbations related to the nominal value of the parameter p_j : 1%, 5%, 10% .

The results were compared in order to decide if for this model the use of DM is necessary or if the computationally simpler DQM can be applied. The results of local PSA applied to E_1 and E_2 for all model parameters are shown with $\Delta =$: 1%, 5%, 10% in Table 5.1 and Table 5.2 while a graphical comparison between DM results and DQM results with $\Delta p_j = 0.01 p_j$ is depicted in Figure 5.1 and in Figure 5.2. Data corresponds to the integrated sensitivities $[S_{int}]_{i,j}$ (eq. (2.26)) calculated through the DM and through the DQM with $\Delta =$: 1%, 5%, 10%.

As evident from Table 5.1 and 5.2, the values calculated with DQM with

$\Delta = 5\%, 10\%$ are sometimes significantly different from the ones calculated with the DM. While from Figure 5.1 and 5.2 DQM results with $\Delta = 1\%$ appear to be similar to DM results for most but not all parameters. From the complexity of computation point of view DM involves an acceptable numerical efforts since the model has not a high complexity and the number of differential equations to solve is "only" $2n \cdot m = (2 \cdot 5) \cdot 28 = 280$.

These considerations suggest to choose the DM instead of DQM for computing sensitivity coefficients and their metrics. From now on the results are obtained with DM, except if differently specified.

	Direct	Difference Quotient					
		Delta 1%		Delta 5%		Delta 10%	
		Value	Var. %	Value	Var. %	Value	Var. %
Beta1	0.9474	0.9528	0.57	0.9681	2.19	0.9785	3.28
Delta1p	-0.9737	-0.9571	1.70	-0.8846	9.15	-0.7800	19.89
Ks	6.15E-05	6.09E-05	0.96	5.86E-05	4.64	5.61E-05	8.86
Sb	0.2164	0.2161	0.16	0.2146	0.82	0.2127	1.69
E2th	0.2518	0.2522	0.14	0.2535	0.66	0.2549	1.23
a1	0.0262	0.0262	0.02	0.0262	0.09	0.0262	0.19
q1	0.0072	0.0072	0.30	0.0071	1.50	0.0070	3.00
q2	-0.7369	-0.7236	1.80	-0.6699	9.10	-0.6013	18.41
Lambda1	-0.0127	-0.0127	0.02	-0.0127	0.06	-0.0127	0.00
Lambda2	0.4656	0.4662	0.14	0.4679	0.48	0.4683	0.59
delta1m	-0.9525	-0.9368	1.65	-0.8684	8.83	-0.7702	19.14
k1s_min	0.0159	0.0159	0.01	0.0159	0.05	0.0159	0.10
th	6.2864	6.5884	4.81	7.2348	15.09	7.6720	22.04
a3	0.9314	0.9367	0.56	0.9516	2.16	0.9618	3.26
q3	-2.7729	-2.6738	3.57	-2.0615	25.66	-1.4580	47.42
q4	0.2620	0.2615	0.19	0.2593	1.03	0.2561	2.24
Lambda3	-3.5667	-3.3815	5.19	-2.3372	34.47	-1.5908	55.40
Lambda4	0.4225	0.4189	0.85	0.4046	4.24	0.3870	8.41
Delta2m	0.7046	0.7069	0.32	0.7130	1.19	0.7163	1.66
k2s_min	-0.7047	-0.6952	1.35	-0.6558	6.94	-0.6028	14.46
f1	-0.7047	-0.6952	1.35	-0.6558	6.94	-0.6028	14.47
Lambda5	2.34E-05	2.50E-05	6.71	3.28E-05	40.29	4.76E-05	103.55
Sigma1	2.29E-05	2.42E-05	5.46	3.00E-05	30.58	3.91E-05	70.41
Beta2	-0.7047	-0.6952	1.35	-0.6558	6.94	-0.6028	14.46
Delta2_m	0.7055	0.7076	0.30	0.7133	1.11	0.7165	1.55
f2	0.0028	0.0028	0.03	0.0028	0.16	0.0028	0.32
Lambda6	0.0385	0.0421	9.33	0.0612	58.83	0.0991	156.96
sigma2	0.0140	0.0144	2.26	0.0157	11.53	0.0174	23.57

Table 5.1: Comparison between integrated sensitivity results of model parameters for E_1 . Values of the integrated sensitivities for direct method and for the difference quotient method with $\Delta = 1\%, 5\%, 10\%$, and percentage difference between the direct method and the others.

	Direct	Difference Quotient					
		Delta 1%		Delta 5%		Delta 10%	
		Value	Var. %	Value	Var. %	Value	Var. %
Beta1	0.0354	0.0317	10.56	0.0210	40.80	0.0137	61.45
Delta1p	-0.0364	-0.0404	11.16	-0.0610	67.68	-0.0930	155.70
Ks	3.70E-05	3.66E-05	0.96	3.53E-05	4.63	3.37E-05	8.86
Sb	0.2366	0.2357	0.37	0.2323	1.81	0.2283	3.49
E2th	0.2750	0.2748	0.11	0.2737	0.49	0.2726	0.88
a1	0.0287	0.0287	0.05	0.0286	0.23	0.0285	0.45
q1	0.0079	0.0079	0.31	0.0078	1.53	0.0077	3.07
q2	-0.8062	-0.7981	1.01	-0.7684	4.69	-0.7359	8.72
Lambda1	-0.0139	-0.0139	0.03	-0.0139	0.12	-0.0139	0.13
Lambda2	0.5090	0.5074	0.33	0.5013	1.51	0.4949	2.79
delta1m	-0.0356	-0.0395	10.92	-0.0592	66.17	-0.0902	153.42
k1s_min	5.95E-04	5.94E-04	0.19	5.89E-04	0.94	5.83E-04	1.88
th	0.2332	0.1202	48.45	0.0299	87.17	0.0150	93.58
a3	0.0348	0.0312	10.40	0.0208	40.32	0.0136	60.95
q3	-0.1037	-0.1428	37.74	-0.3821	268.61	-0.5366	417.74
q4	0.0098	0.0094	3.43	0.0083	15.62	0.0070	28.01
Lambda3	-0.1333	-0.1998	49.87	-0.5496	312.27	-0.7008	425.64
Lambda4	0.0151	0.0142	5.68	0.0114	24.15	0.0090	40.31
Delta2m	-0.2298	-0.2303	0.23	-0.2306	0.36	-0.2282	0.66
k2s_min	0.2298	0.2268	1.30	0.2133	7.21	0.1931	16.00
f1	0.2298	0.2268	1.30	0.2133	7.21	0.1931	16.00
Lambda5	-1.62E-05	-1.73E-05	6.71	-2.27E-05	40.29	-3.29E-05	103.54
Sigma1	-1.51E-05	-1.59E-05	5.13	-1.94E-05	28.47	-2.49E-05	64.76
Beta2	0.2298	0.2268	1.30	0.2133	7.21	0.1931	16.00
Delta2_m	-0.2304	-0.2308	0.19	-0.2308	0.20	-0.2283	0.88
f2	-0.0019	-0.0019	0.06	-0.0019	0.28	-0.0019	0.55
Lambda6	-0.0266	-0.0289	8.92	-0.0411	54.59	-0.0627	135.92
sigma2	-0.0088	-0.0090	1.83	-0.0096	9.19	-0.0104	18.34

Table 5.2: Comparison between integrated sensitivity results of model parameters for E_2 . Values of the integrated sensitivities for direct method and for the difference quotient method with $\Delta = 1\%, 5\%, 10\%$, and percentage difference between the direct method and the others.

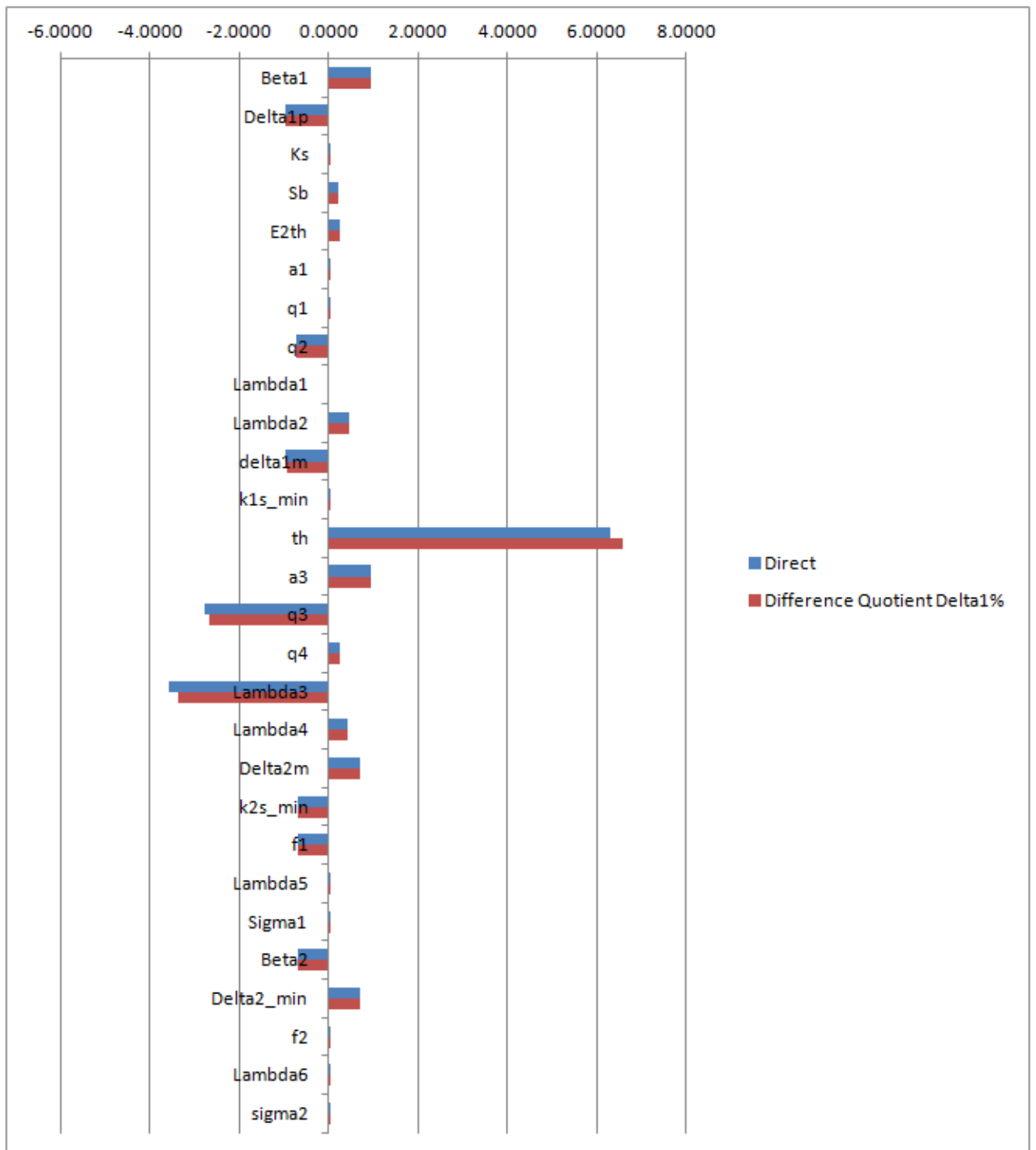


Figure 5.1: Comparison between integrated sensitivity results of model parameters for E_1 with Direct Method (blue) and with Difference Quotient Method with $\Delta p_j = 1\%$ (red).

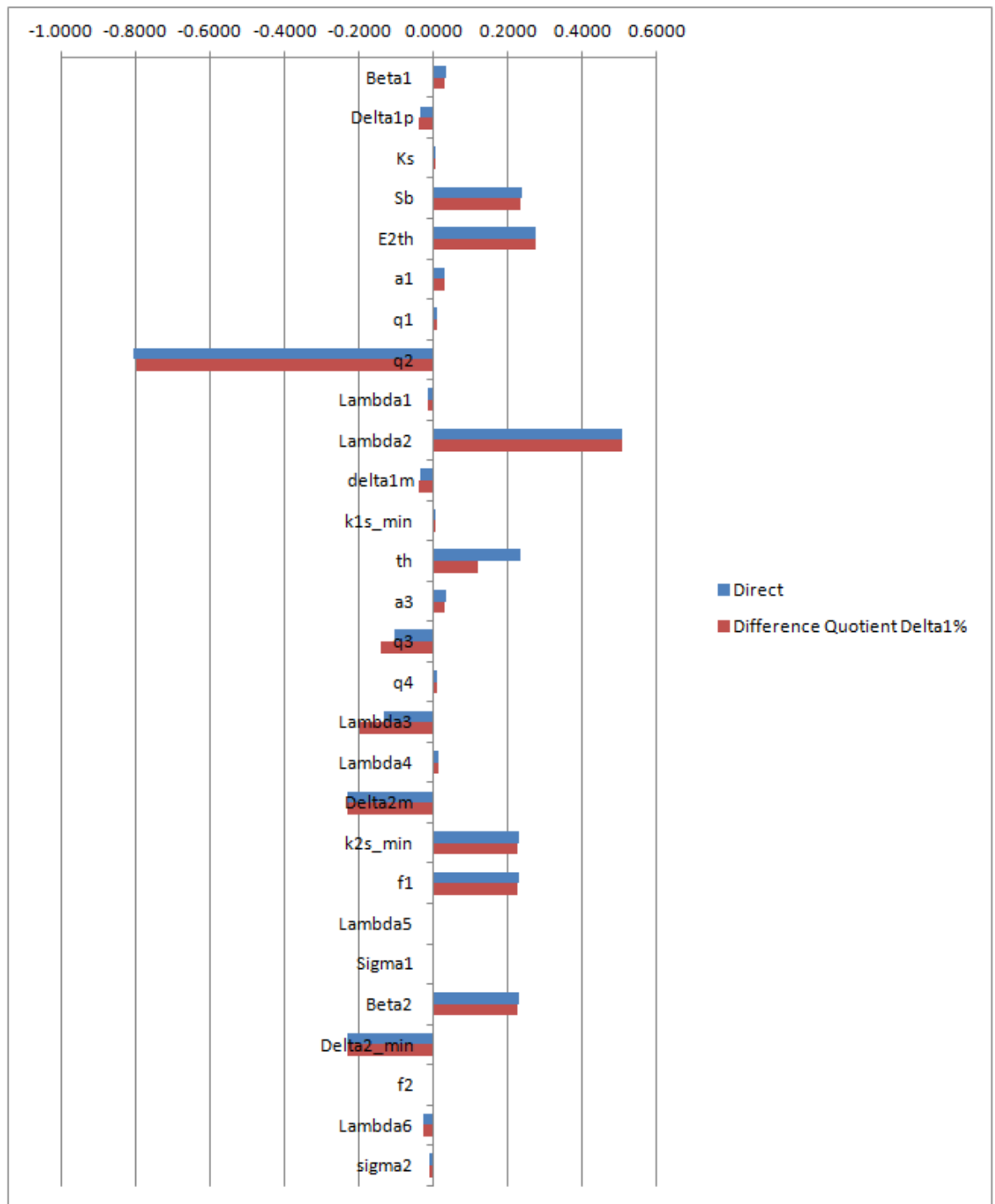


Figure 5.2: Comparison between integrated sensitivity results of model parameters for E_2 with Direct Method (blue) and with Difference Quotient Method with $\Delta p_j = 1\%$ (red).

5.2 PSA results with DM

5.2.1 Main viral DNA replication regulator E_1

It is evident from Figure 5.1 that, for the protein E_1 , the parameters with higher integrated sensitivity magnitude are λ_3 , q_3 and t_h . These parameters belong to the time variant forcing function of equation (3.9) that represents the splicing flux for the mE_1 transcript and their meanings is summarized in Table 5.3.

Parameter	Description
λ_3	Half-maximal of mE_1 transcription rate during early phase
q_3	Hill coefficient of mE_1 transcription rate during early phase
t_h	Time threshold of mE_1 transcription rate

Table 5.3: Model parameters with higher sensitivity for E_1 . First column contains the names of the parameters; second column shows their meanings.

In the next figures the curve trends of sensitivity coefficients are shown for model parameters λ_3 , q_3 and t_h .

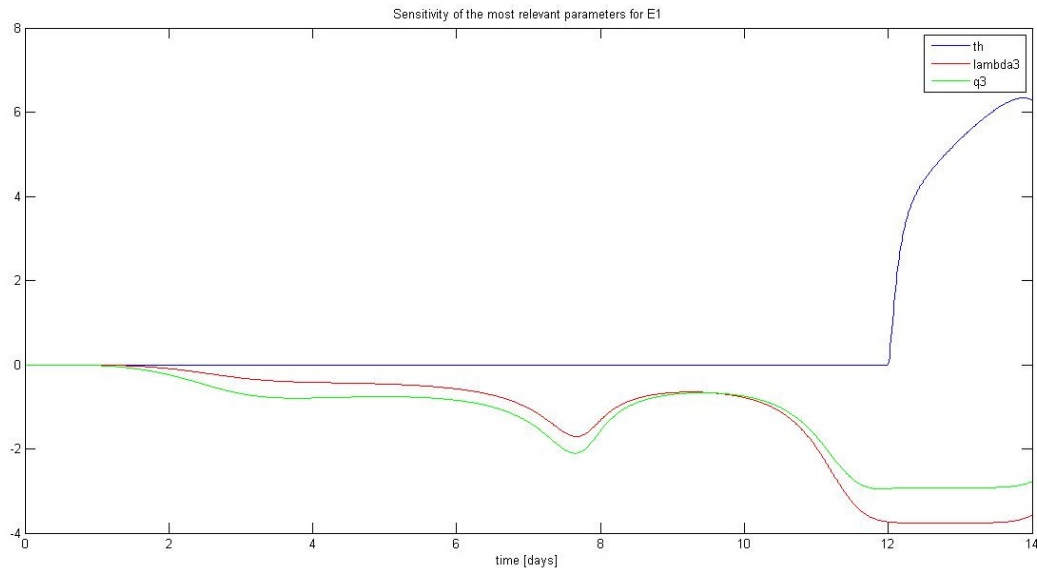


Figure 5.3: Time course of E_1 sensitivity to λ_3 , q_3 and t_h calculated with DM.

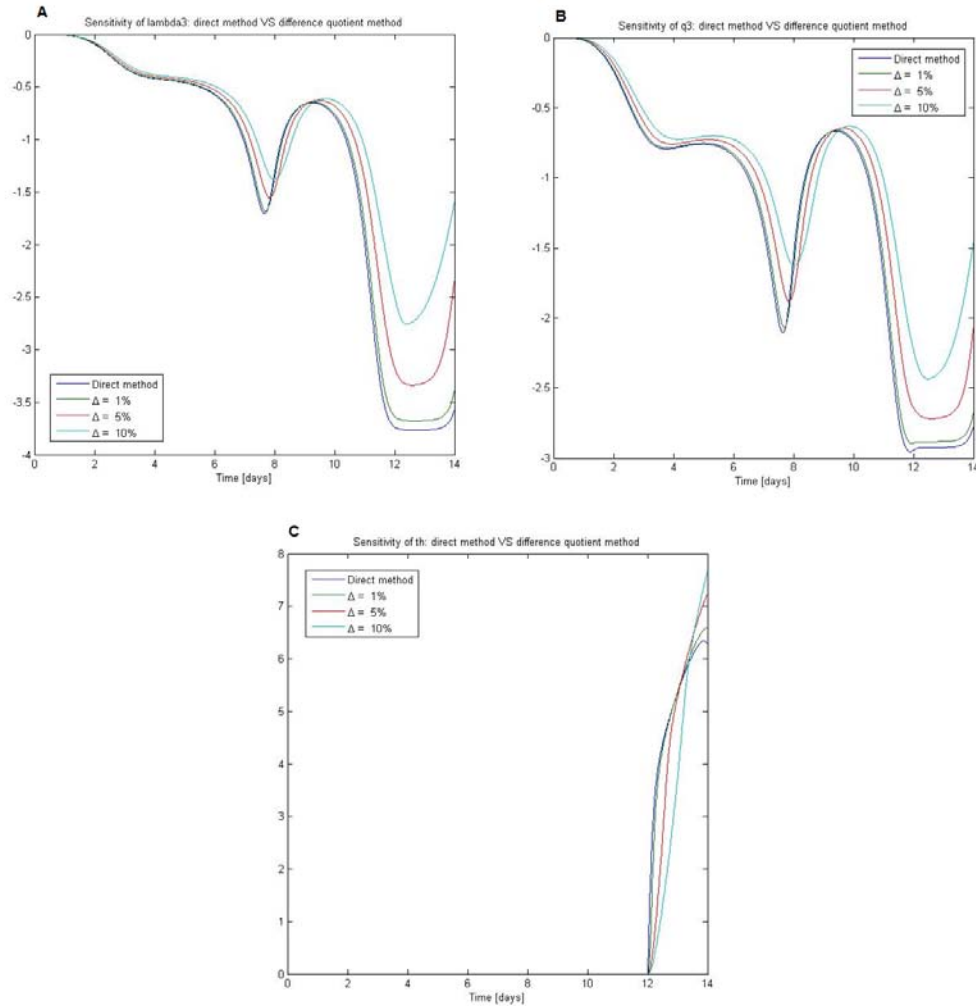


Figure 5.4: Time course of E_1 sensitivity to (A) λ_3 , (B) q_3 and (C) t_h calculated with DM and DQM.

As evidenced by Figure 5.3 and Figure 5.4, the parameter t_h seems to have a higher influence in the last part of the time scale, from $t = 12$ day to $t = 14$ day, on the other hand λ_3 and q_3 have a higher influence in the previous times especially around $t = 7 - 8$ days.

The sensitivity time curve of t_h is in agreement with its meaning: t_h is a threshold and it plays a role after 12 days. More specifically, t_h represents the time (nominal value equal to 12 days) when E_1 transcripts shows the stronger peak in the expression [18]. Hence, it is reasonable that parameter t_h presents a high sensitivity for E_1 since it represents the time in which mE_1 expresses the highest level of Viral DNA (vDNA) amplification.

Conversely the importance of λ_3 and q_3 and their sensitivity trends are in accordance with the literature. It was evidenced in the study of Ozbun et al. [18] that a tissue culture system infected by HPV faithfully mimics the differentiation and maturation processes of keratinocytes that form epithelium. The tissue culture undergo a program of stratification and differentiation by day 12, exactly like the healthy tissue. At day 12, the infected tissue is "*fully able to support the complete HPV life cycle as assayed by the ability to detect amplification, late gene transcripts, capsid proteins, and viral particles*" [18]. More specifically, the study showed an increase in the levels around 8-10 days, followed by a peak around day 12 and then a drop.

These aspects, mirrored by the systems and the sensitivity study, confirms the importance of the time dependent splicing function. Parameters λ_3 and q_3 represents negative sensitivity values since increasing values of these parameters leads to a decreasing in the values of $k_{1s}(t)$; moreover the sensitivity curve trends show for both λ_3 and q_3 a small peak around day 8 and a higher peak at day 12. These are the two times at in which the study in [18] showed the rising in E_1 RNA levels. From a biological point of view, these two parameters are connected to the way the viral cycle system controls DNA replication speed. The control is represented by a sigmoid where q_3 regulates the promptness of the system and the sigmoid slope while λ_3 regulates the velocity to reach the maximum. A small λ_3 causes higher E_1 concentrations and DNA replication is faster but normally the system tries to keep high λ_3 to counteract cellular differentiation.

Another observation concerns the importance of δ_2^{min} , that even if appears with a smaller sensitivity than the other three parameters, deserves a consideration. It is an important parameter involved in the co-regulation of E_1 and E_2 and it is expected to appear as a driving factor as confirmed by sensitivity results.

Besides the integrated sensitivities, the other two metrics of equation (2.26) were implemented : sensitivity based on infinite norm and sensitivity at a particular time both from the results of the direct method . In this analysis was chosen $t = 12.3$ day in order to highlight possible unexpected behaviors of the parameters after the threshold t_h .

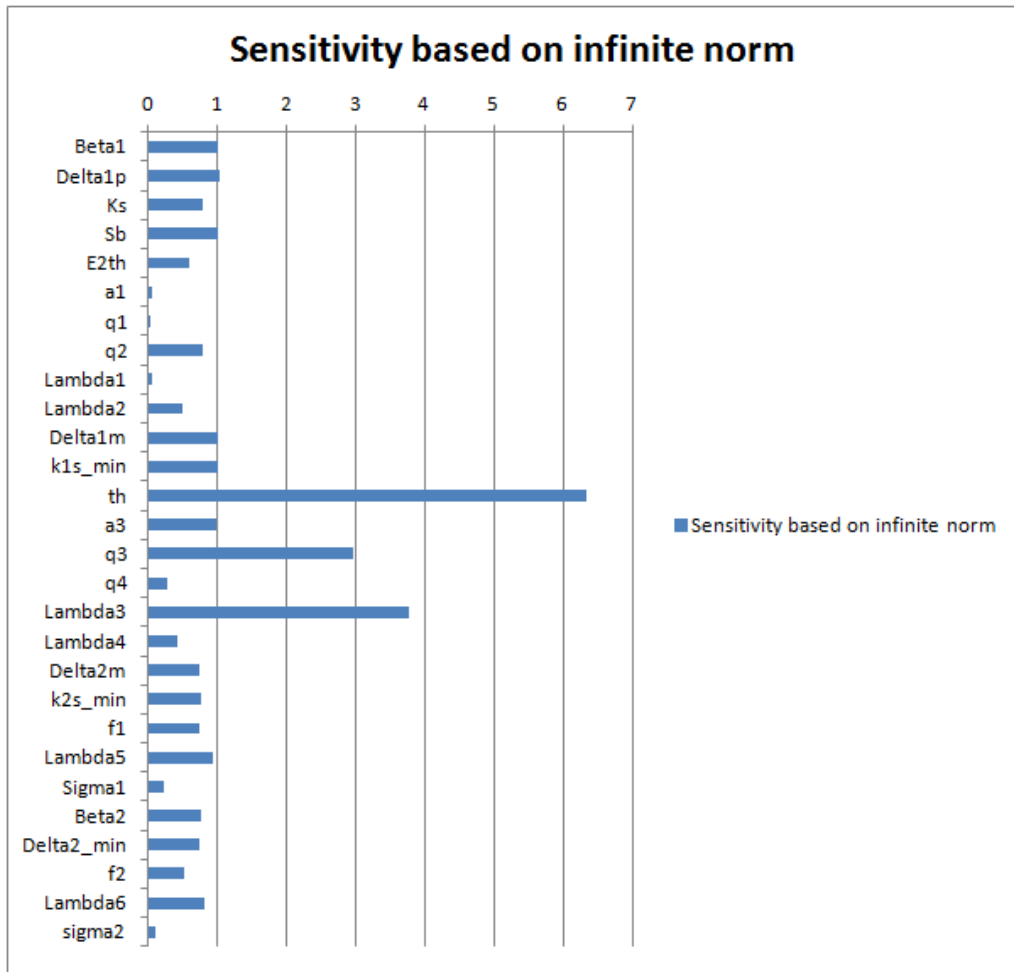


Figure 5.5: Sensitivity based on infinite norm of model parameters for E_1 .

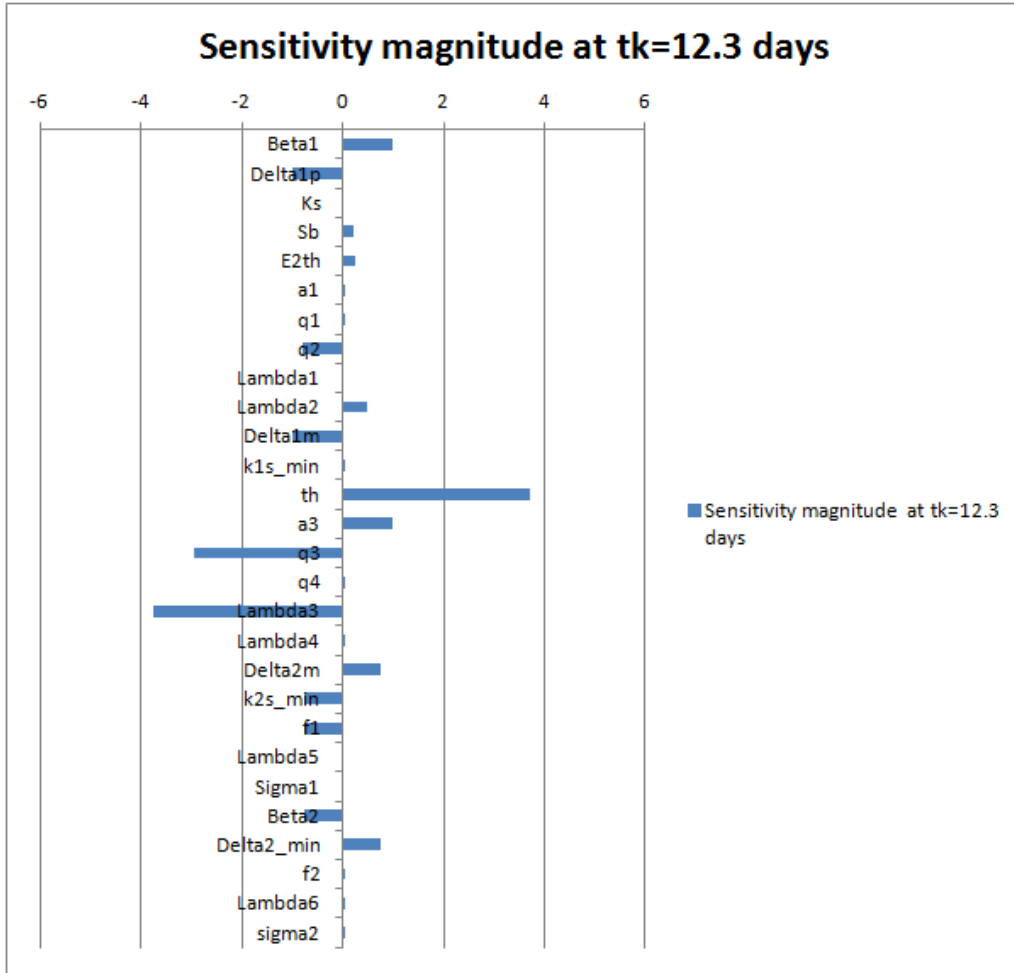


Figure 5.6: Sensitivity at $t = 12.3$ days for model parameters for E_1 .

As evident from Figure 5.5 and Figure 5.6 the results obtained with the integral metrics are confirmed and no other unexpected results are revealed.

5.2.2 Main transcriptional regulator E_2

Concerning protein E_2 , Figure 5.2 suggests that the parameters with higher integrated sensitivity magnitude are λ_2 , q_2 and E_2^{th} followed by S_b and δ_2^{min} , even if the magnitudes are much smaller than in E_1 case. The lower magnitude can be explained through the fact that E_1 RNA levels are much higher than E_2 levels during more or less all the differentiation period [18]. These parameters, with the exception of δ_2^{min} , belong to the equation (3.7) of S_x that refers to the transcription of x enhanced by low values of E_2 concentration and repressed by elevated E_2 concentration, confirming the role of E_2

as main transcriptional regulator.

The meaning of the previously mentioned parameters is summarized in the following Table.

Parameter	Description
λ_2	x concentration for half-maximal x repression
q_2	Hill coefficient of E_2 negative feedback
E_2^{th}	E_2 concentration threshold of transactivation/repression
S_b	Basal transcription synthesis of the primary transcript x
δ_2^{min}	Basal E_2 degradation rate without E_1 negative regulation

Table 5.4: Table of the model parameters with higher sensitivity for E_2 . First column contains the names of the parameters; second column shows their meanings.

In the next figures the curve trends of sensitivity coefficients are shown for the most sensitive parameters for E_2 .

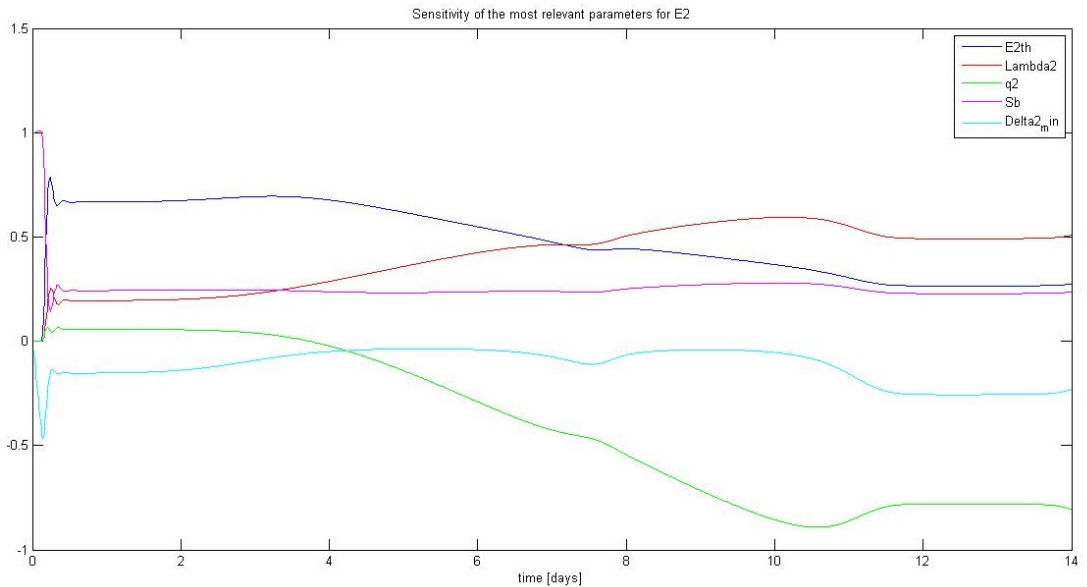


Figure 5.7: Time course of E_2 sensitivity to λ_2 , q_2 and E_2^{th} , S_b , δ_2^{min} calculated with DM.

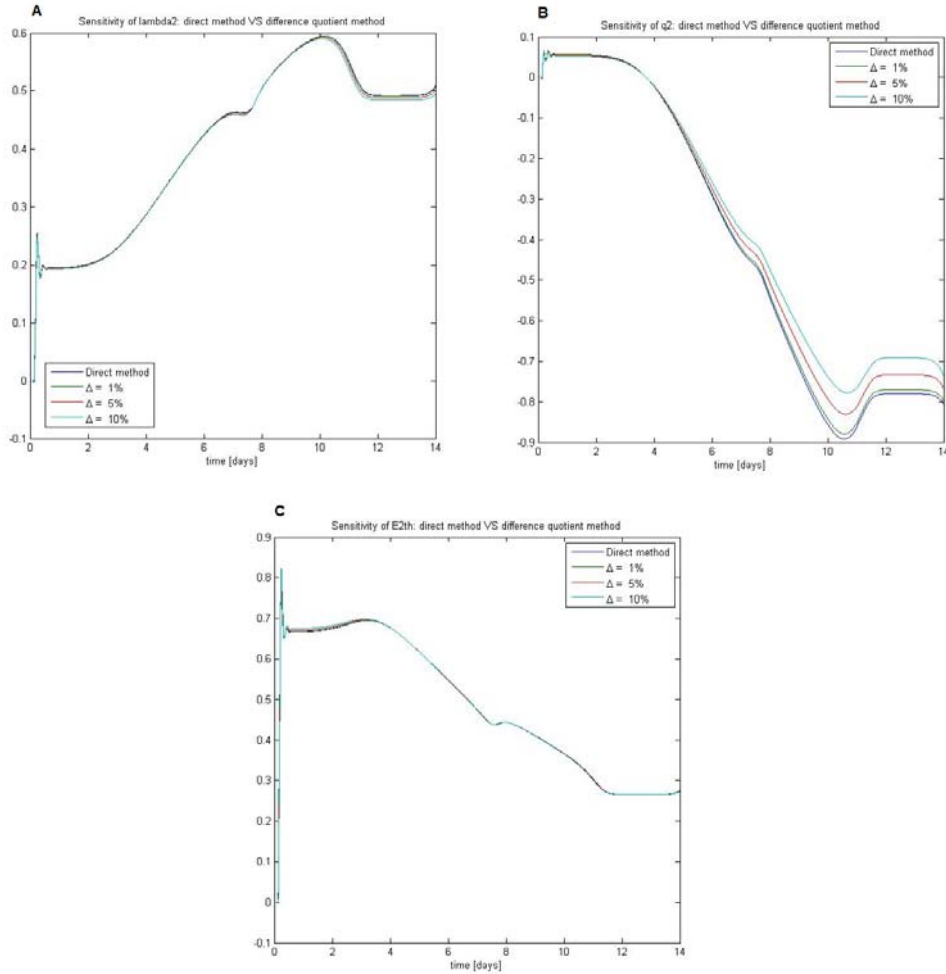


Figure 5.8: Time course of E_2 sensitivity to (A) λ_2 , (B) q_2 and (C) E_2^{th} calculated with DM and DQM.

As highlighted by Figure 5.7 and Figure 5.8, the parameter E_2^{th} seems to have a bigger influence in the first part of the time scale and has the same sensitivity of λ_2 around $t = 7 - 8$ day, on the other hand q_2 has a bigger influence in the second half part of the time scale. Moreover all sensitivity curves show a peak at the beginning of the time sequence that can be re-conducted to the trend of the system state variables that present a very fast initial phase before reaching the steady state. E_2^{th} dominance until 7-8 days can be explained pointing out that, enhancing E_2^{th} threshold means to prolong the phase of x transcription hence increasing E_2 concentration. On the other hand, λ_2 and q_2 importance highlights the predominance of negative feedback of E_2 over the positive one (statement that is confirmed by the low

sensitivity of parameter q_1 belonging to the positive feedback). Parameter q_2 has a negative sensitivity since high values of this parameter enhance the negative feedback of E_2 suppressing x transcription, hence E_2 production. Concerning S_b and δ_2^{min} , the first has not a very informative curve trend, reaching a steady state after the first peak, the second has a curve trends that after the first peak is more or less symmetrical to the one of E_2^{th} .

Figure 5.10 shows the sensitivity magnitude after time threshold t_h , the results confirm the ones of integrated sensitivities. On the other hand Figure 5.9 shows the results for sensitivity based on infinite norm from where it is highlighted the importance of other two parameters. In particular this analysis underlines the importance of E_1 feedback, both negatively, represented by the high sensitivity of parameter q_3 , and positively, represented by parameter k_{2s}^{min} . Also β_2 presents an high sensitivity for E_2 , representing its translation rate. Looking at integrated sensitivity, one can see that q_3 and k_{2s}^{min} and β_2 are also present as highly sensitive parameters, hence showing that infinite norm constitutes a confirmation of the previous results and not a contradiction.

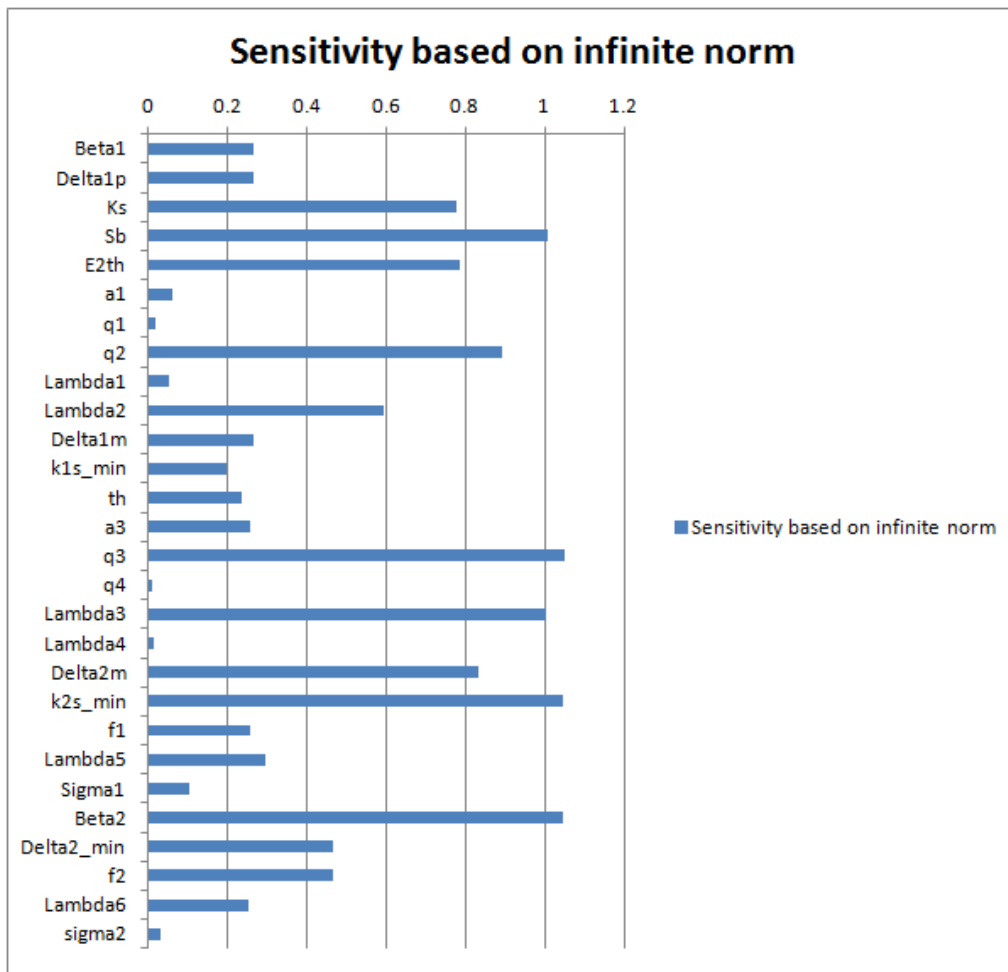


Figure 5.9: Sensitivity based on infinite norm of model parameters for E_2 .

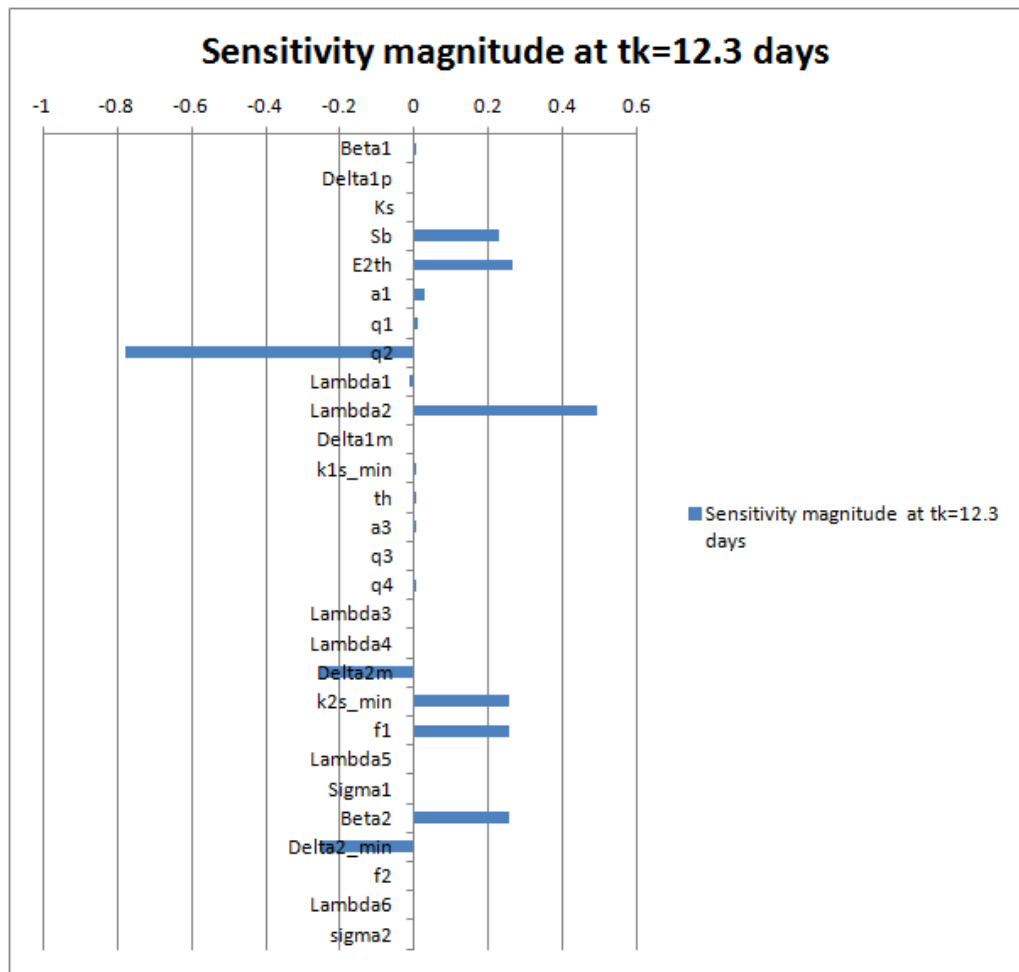


Figure 5.10: Sensitivity at $t = 12.3$ days for model parameters for E_2 .

5.2.3 The primary polycistronic transcript X

Sensitivity metrics of Equation 2.26 were implemented for the state variable x , representing the primary transcript.

In the order Figure 5.11 shows the time integrated sensitivity coefficients, Figure 5.12 shows sensitivity at the particular time $t = 12.3$ day and Figure 5.14 shows sensitivity based on infinite norm.

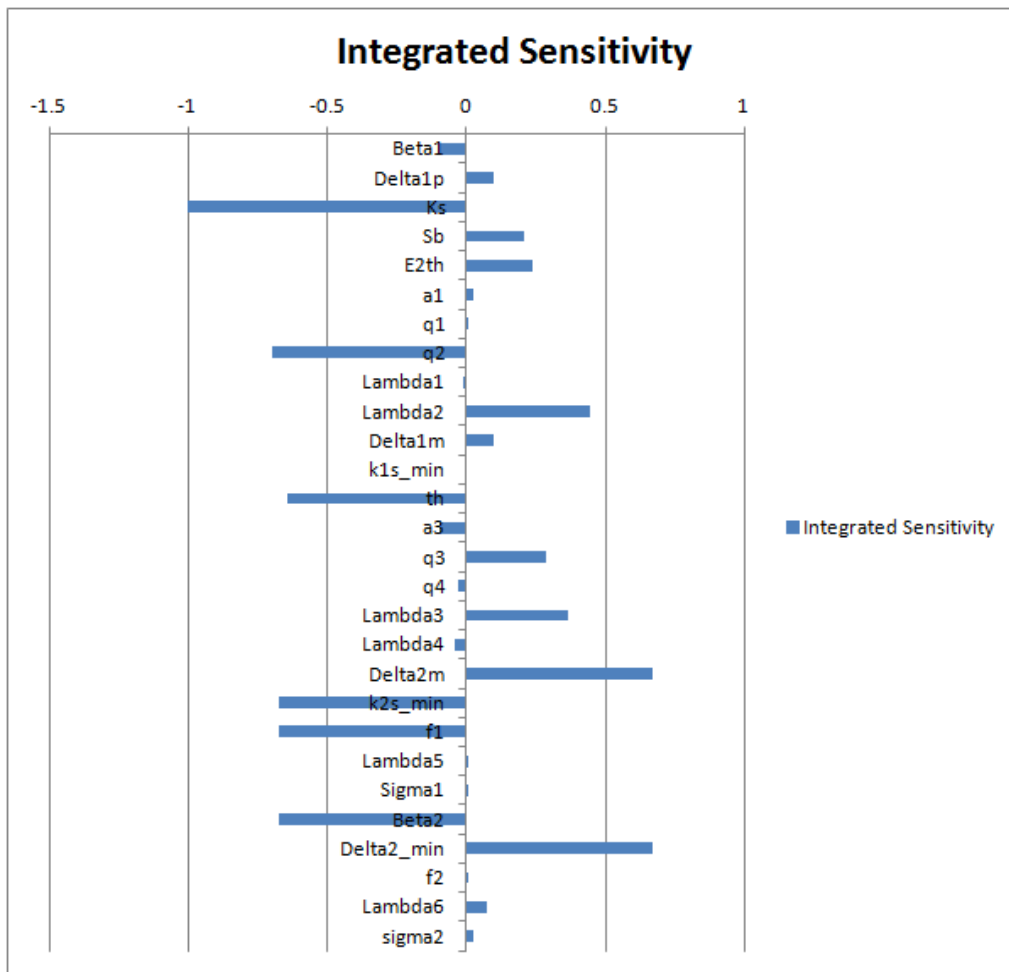


Figure 5.11: Time integrated sensitivity for x .

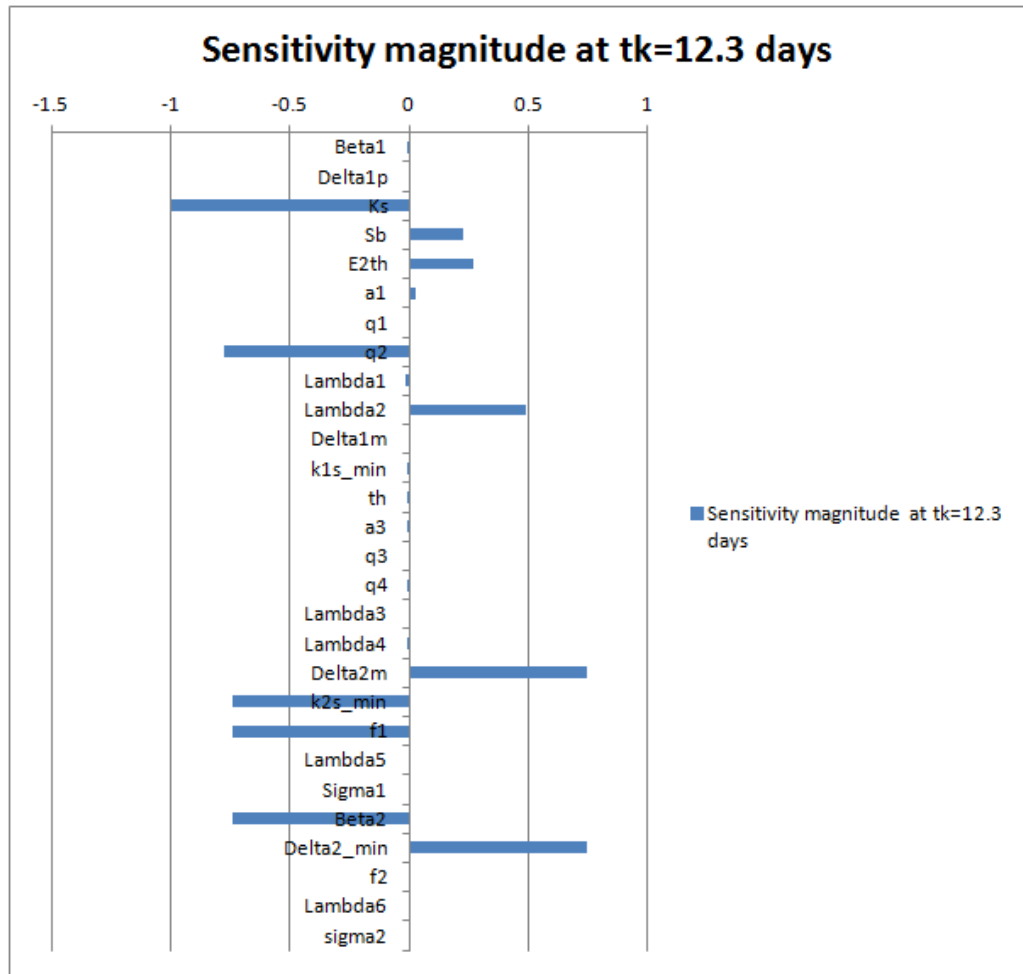


Figure 5.12: Sensitivity at $t = 12.3$ days for model parameters for x .

Time integrated sensitivity coefficients show that the three parameters with higher sensitivity for x are k_s , δ_2^{min} and q_2 and the same result is confirmed by sensitivity at time $t = 12.3$ day (Figure 5.12). Other noticeable parameters are k_{2s}^{min} , β_2 and f_1 . Their meanings are summarized in Table 5.5.

Parameter	Description
k_s	nucleo cytoplasmatic transport
δ_2^{min}	Basal E_2 degradation rate without E_1 negative regulation
q_2	Hill coefficient of E_2 negative feedback
f_1	Fold increase of mE_2 transcription rate
k_{2s}^{min}	Basal mE_2 transcription rate without E_1 positive regulation
β_2	E_2 translation rate

Table 5.5: Model parameters with higher sensitivity for x . First column contains the names of the parameters; second column shows their meanings.

As evident from the Table, k_s is the parameter that refers to x degradation through nucleo cytoplasmatic transport however δ_2^{min} refers to the degradation rate of E_2 (Equation 3.12) and q_2 is involved in the transcription of x regulated by E_2 concentration.

Parameters k_{2s}^{min} and f_1 belong to Equation 3.11 that accounts for the mE_2 transcript splicing flux, while β_2 accounts for E_2 translational rate. Concerning these results the importance of E_2 activity for the regulation of primary transcript x is evident. Sensitivity time curves of these parameters were plotted together for comparison in Figure 5.13.

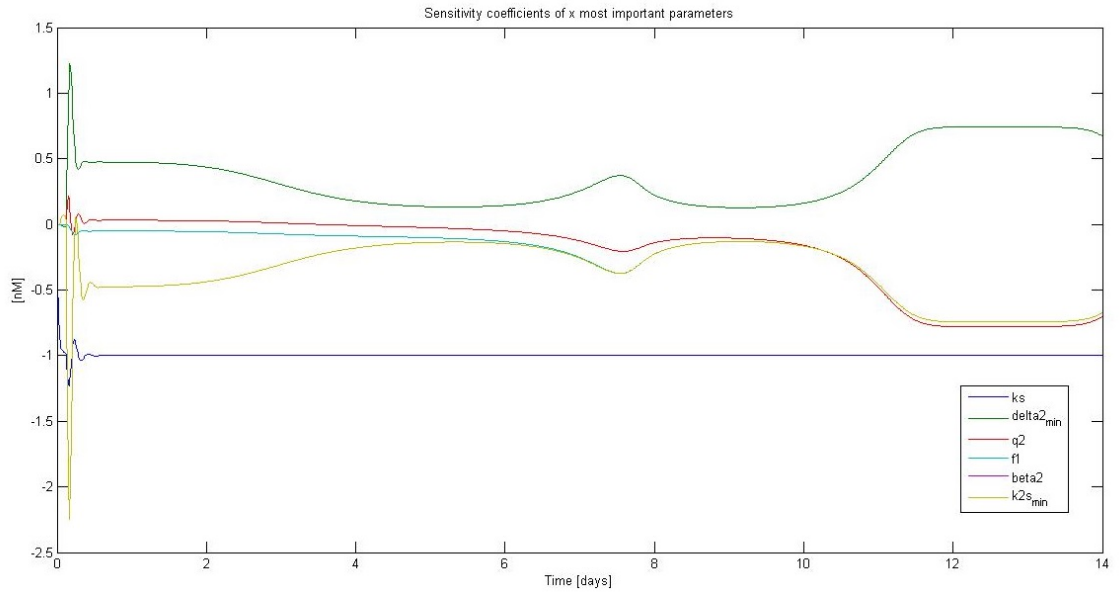


Figure 5.13: Time course of x sensitivity to k_s , δ_2^{min} and q_2 , β_2 , f_1 and k_{2s}^{min} calculated with DM.

All parameters show a fast peak at the beginning of the time scale, mirroring the trend of the system state variables. Besides the first peak of δ_2^{min} , the parameter with higher sensitivity is k_s that presents a steady sensitivity curve at -1 after the first peak. Even if this curve trend is not informative, it is mandatory to underline the importance of parameter k_s since it is bonded to the velocity of the splicing and conversion into transcripts, which is very high in eukaryotes; this is important considering that HPV is a DNA virus. The sensitivity curves of q_2 and δ_2^{min} are more or less symmetrical after day 6 and present a small peak around day 8 and highest sensitivity after day 12.

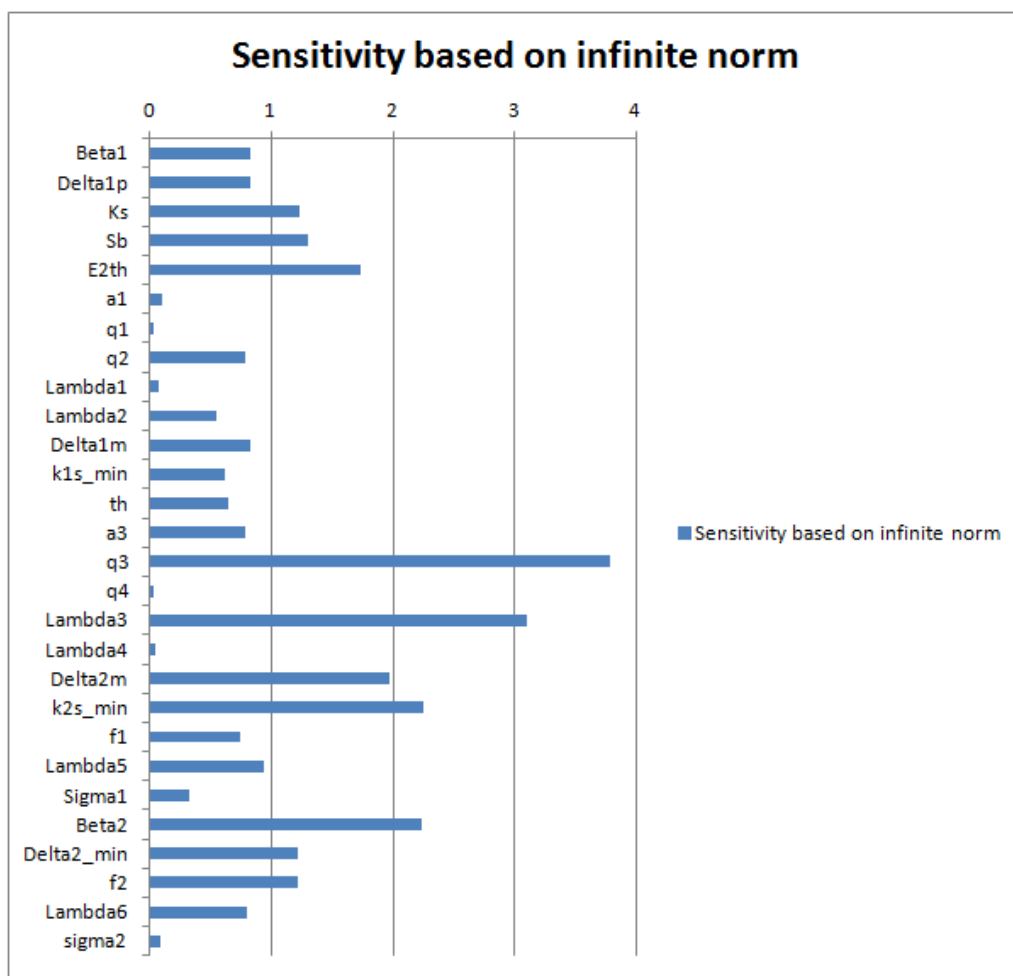


Figure 5.14: Sensitivity based on infinite norm of model parameters for x .

Sensitivity based on infinite norm depicts the importance of parameters belonging to the gene regulatory network that leads to E_2 expression, demonstrating again the importance of E_2 feedback for the state variable x .

5.2.4 mRNA mE_1 encoding for protein E_1

Sensitivity metrics of Equation 2.26 were implemented for the state variable mE_1 , representing the spliced mRNA that encodes for E_1 protein.

In this case the results of sensitivity analysis for E_1 are perfectly reproduced. As evident from Figure 5.15 showing the time integrated sensitivity coefficients, Figure 5.16 showing sensitivity at the particular time $t = 12.3$ day and Figure 5.17 showing sensitivity based on infinite norm, the three parameters with higher sensitivity are λ_3 , q_3 and t_h . These are the same parameters that

present the highest sensitivities for E_1 and it is not surprising. mE_1 concentration is regulated by the effects of keratinocytes differentiation ($k_{1s}(t)$) and for this reason the controlling factor for this state variable belong reasonably to this specific process.

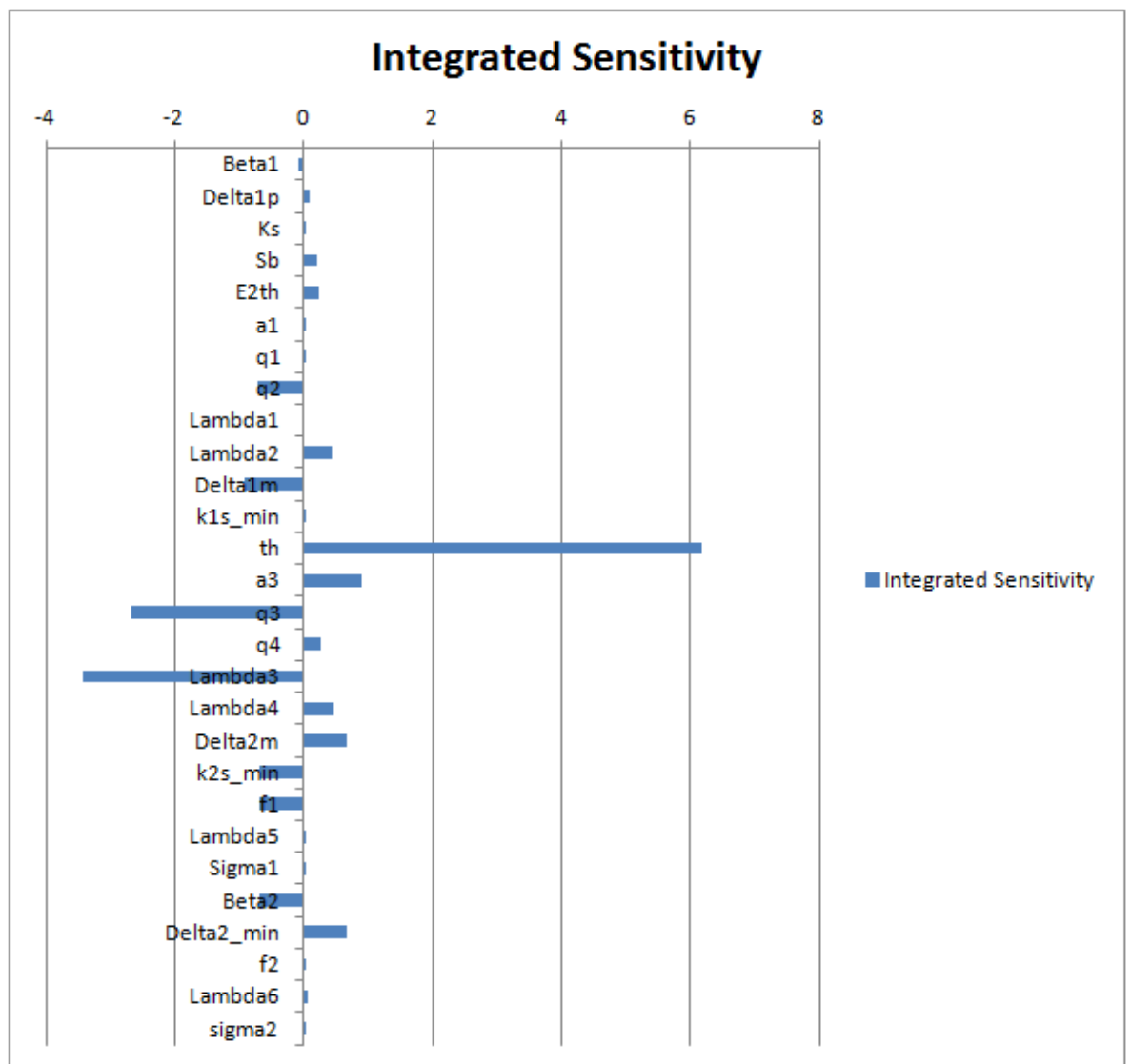


Figure 5.15: Time integrated sensitivity for mE_1 .

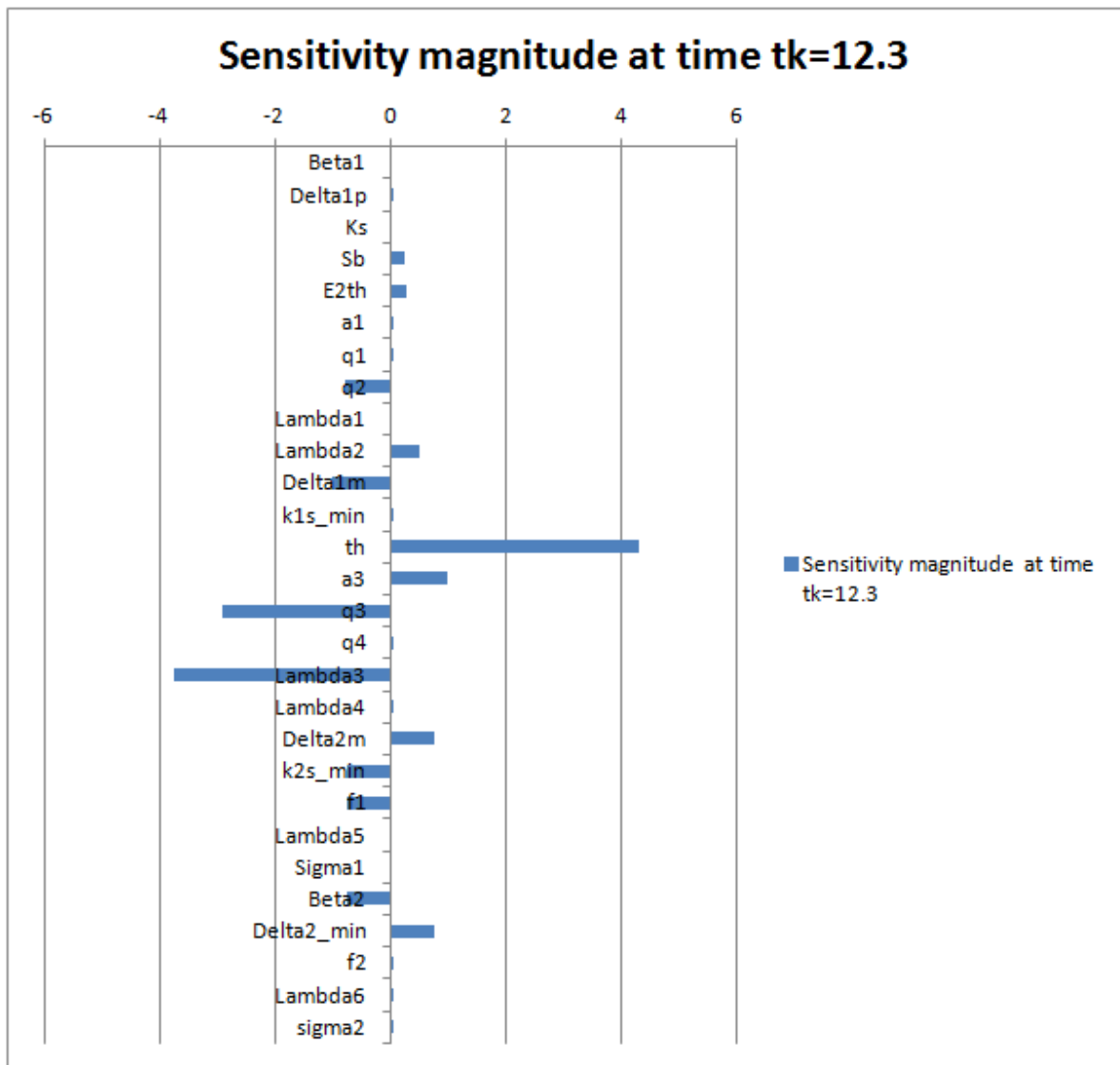


Figure 5.16: Sensitivity at $t = 12.3$ days for model parameters for $mE1$.

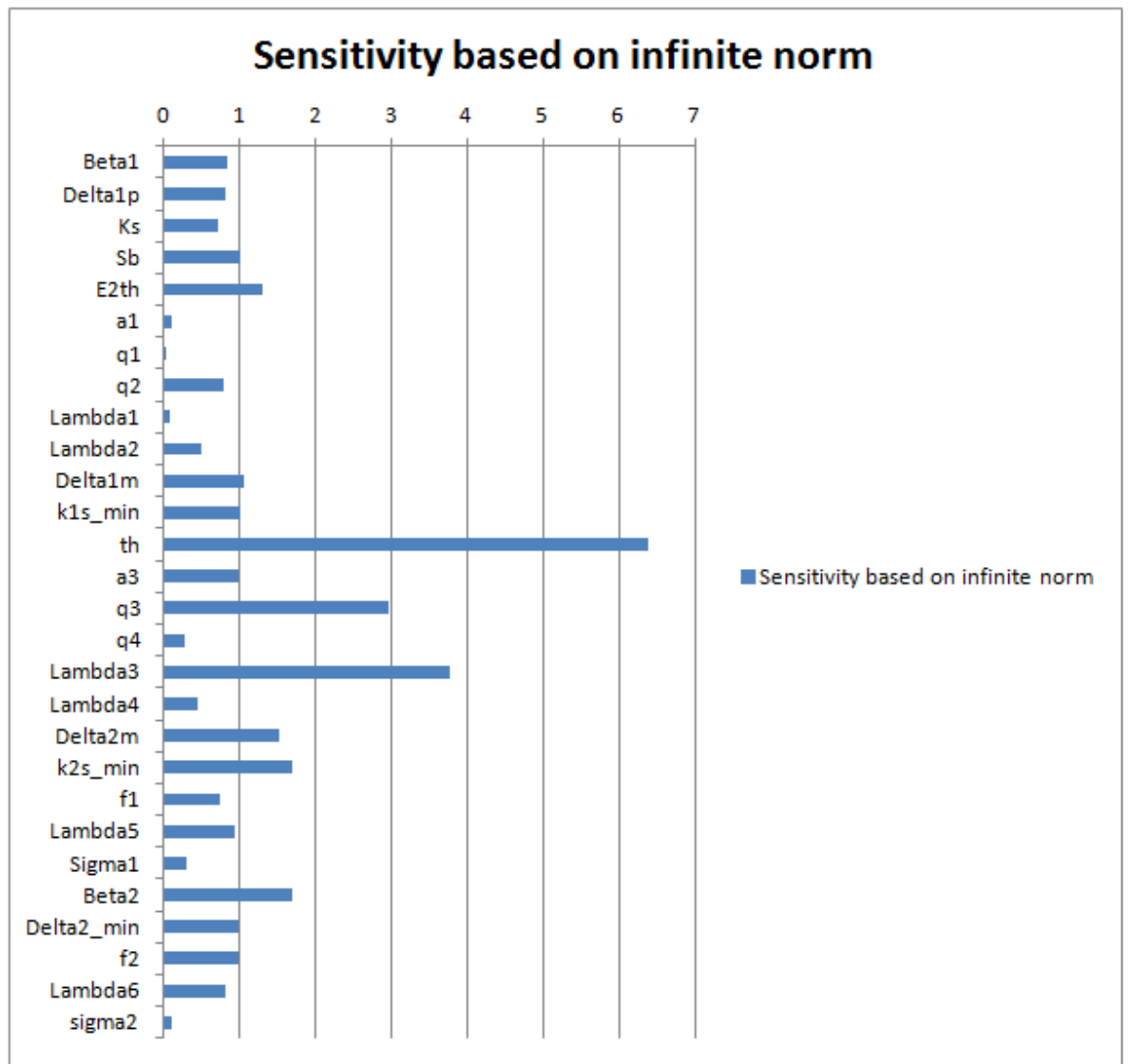


Figure 5.17: Sensitivity based on infinite norm of model parameters for mE_1 .

Sensitivity time curves for mE_1 most sensitive parameters are shown in Figure 5.18.

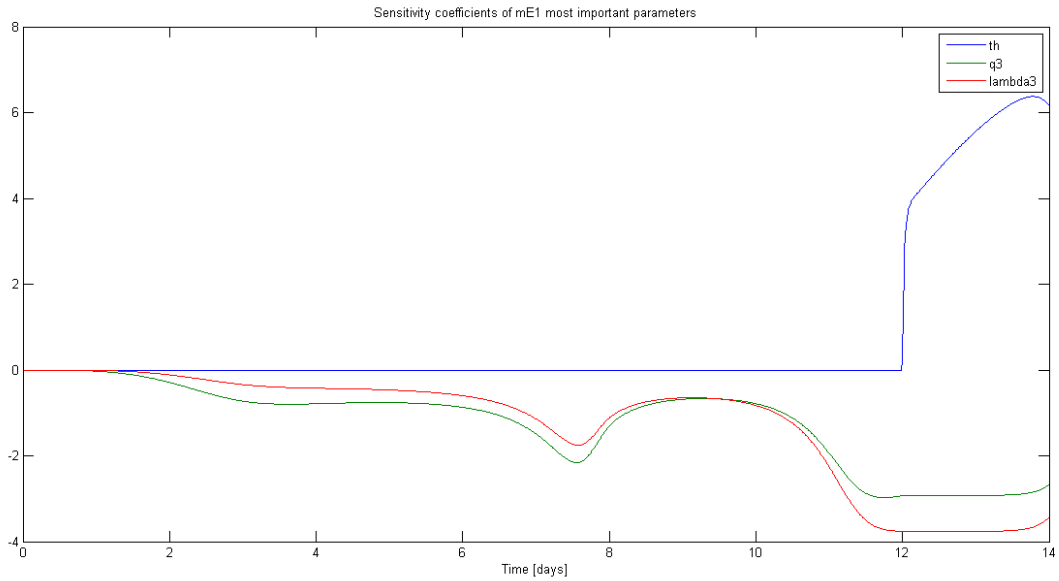


Figure 5.18: Time course of mE_1 sensitivity to λ_3 , q_3 and t_h calculated with DM.

The sensitivity time curves are almost coincident to the ones of E_1 analysis, confirming the strong correlation between the two variables.

5.2.5 mRNA mE_2 encoding for protein E_2

Sensitivity metrics of Equation 2.26 were implemented for the state variable mE_2 , representing the spliced mRNA that encodes for E_2 protein.

As evident from Figure 5.19 and Figure 5.20 both time integral sensitivity and sensitivity at $t = 12.3$ day reveal that the parameters with higher sensitivity for mE_2 are q_2 , β_2 and δ_2^{min} . It is important also to point out, the high sensitivities of λ_2 and of f_1 and k_{2s}^{min} belonging to Equation 3.11.

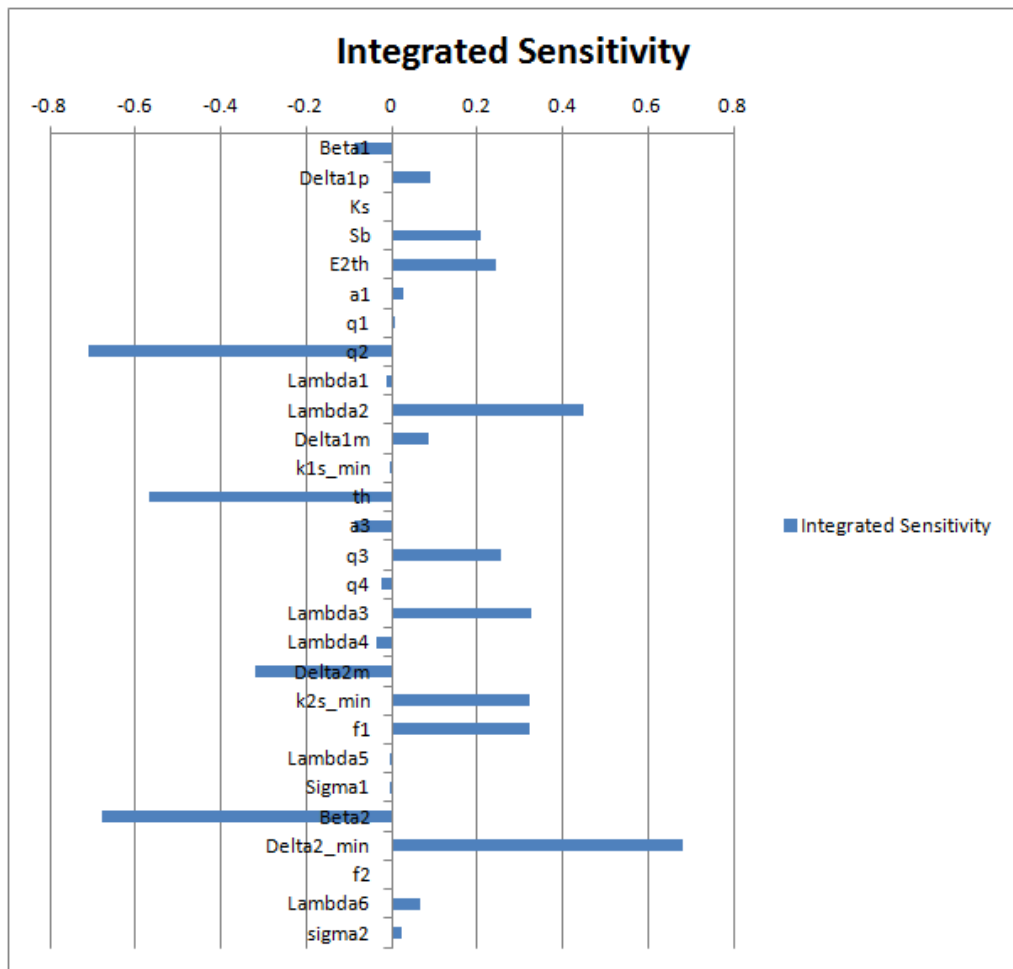


Figure 5.19: Time integral sensitivity for mE_2 .

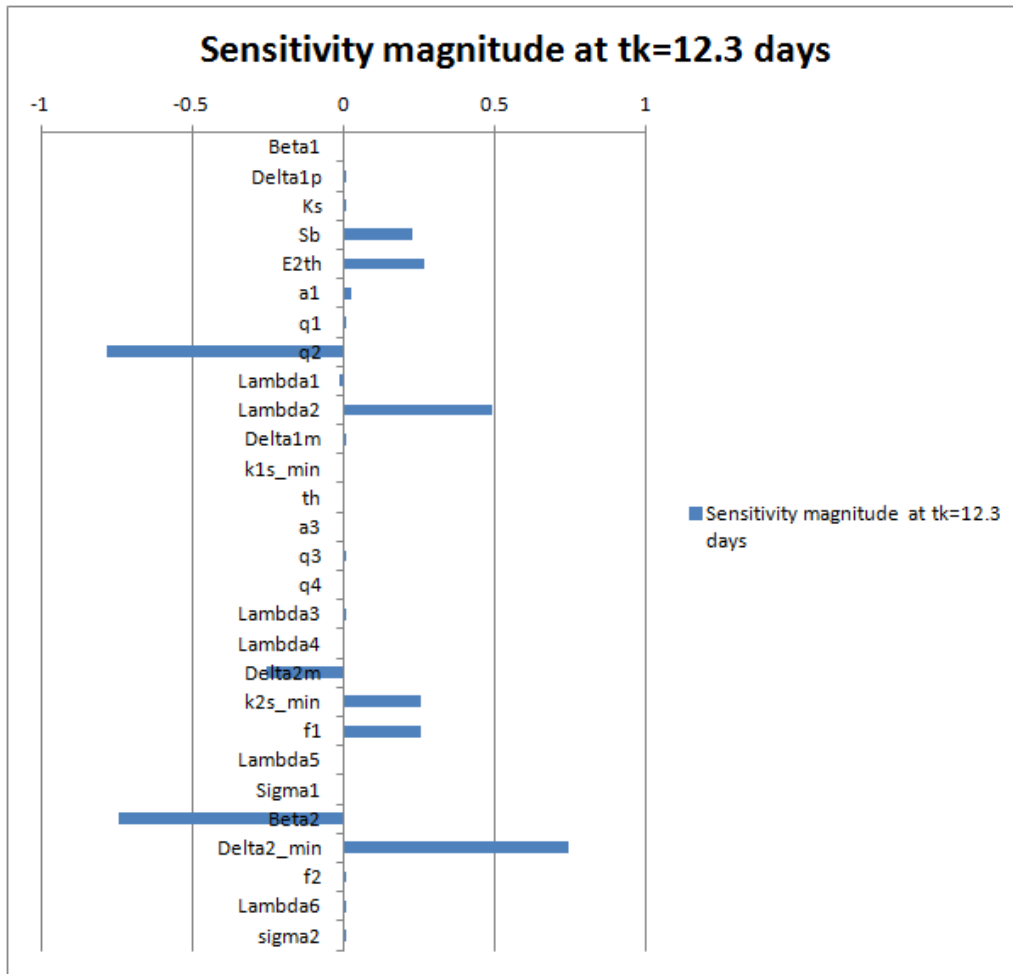


Figure 5.20: Sensitivity at $t = 12.3$ days for model parameters for mE_2 .

In Table 5.6 the meanings of the most sensitive parameters for mE_2 are depicted .

Parameter	Description
q_2	Hill coefficient of E_2 negative feedback
δ_2^{min}	Basal E_2 degradation rate without E_1 negative regulation
β_2	E_2 translation rate

Table 5.6: Model parameters with higher sensitivity for mE_2 . First column contains the names of the parameters; second column shows their meanings.

The three parameters of Table 5.6 are all involved in the regulation of E_2 and their sensitivity time curves are presented in Figure 5.21.

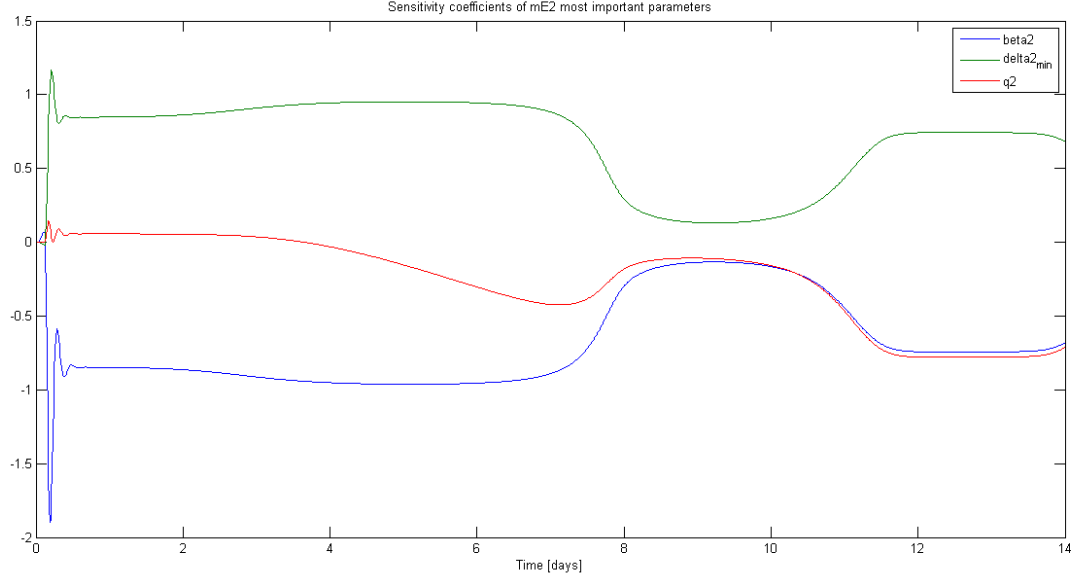


Figure 5.21: Time course of mE_2 sensitivity to q_2 , β_2 and δ_2^{min} calculated with DM.

As evident from Figure 5.21 the three parameters present initial peaks, similar to the ones of the other analysis and attributable to the fast phases that characterize the trends of the state variables. At the beginning of the simulation time, β_2 has the highest sensitivity, then the parameters reach the steady state where β_2 and δ_2^{min} present more or less symmetrical values. Around days 8-10 all parameters show a variation (positive for q_2 and β_2 and negative for δ_2^{min}) and another steady state until day 12 followed by another opposite variation and a steady state after day 12. Again there is a confirmation of the importance of days 8 and 12 in the life cycle of the virus, periods where E_1 and E_2 mRNA synthesis increase. Moreover the importance of β_2 is explained by the fact that it regulates the expression of E_2 that turns down x concentrations and consequently mE_2 concentration. For parameter q_2 (and also for λ_2) the considerations made for E_2 are still valid while the importance of δ_2^{min} points out that no reduction can be applied to model equation (3.12). Concerning the parameters mentioned in the beginning, f_1 and k_{2s}^{min} , their results are interesting because, even if their correct nominal values are still under investigation, their role is connected to the steady state value that E_2 assumes which is regulated by E_1 feedback (Equation 3.12).

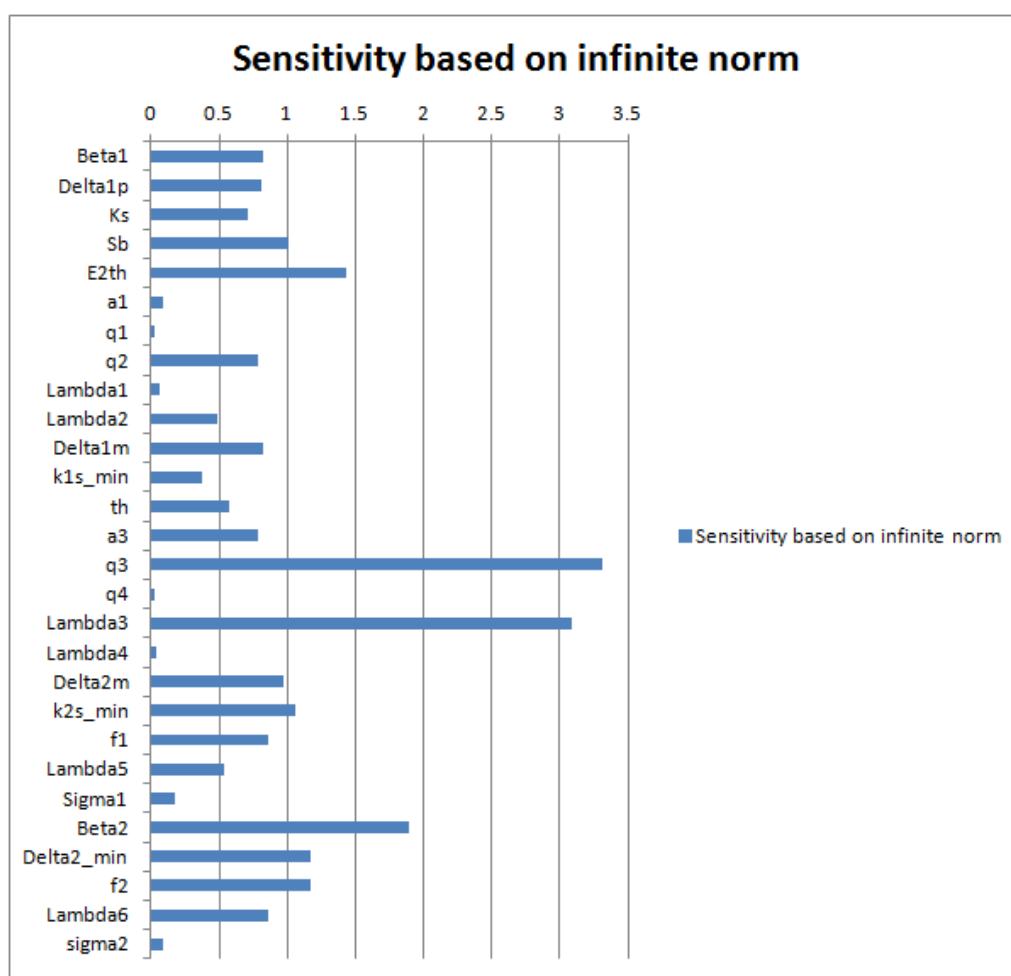


Figure 5.22: Sensitivity based on infinite norm of model parameters for mE_2 .

On the other hand, sensitivity based on infinite norm (Figure 5.22) shows the importance of parameters like q_3 and λ_3 belonging to the splicing flux for mE_1 transcript. These results can be explained as a confirmation of the importance of E_1 regulation path for mE_2 since the splicing flux for mE_2 transcript is positively regulated by E_1 . Again, looking at integrated sensitivity results, one can see that no contradictory results are highlighted in Figure 5.22 with respect to Figure 5.19.

5.3 Impulse parametric sensitivity analysis

The sensitivity coefficients produced by iPSA have a double time dependence, from the time of perturbation τ and from the observation time t (with $t \geq \tau$).

Hence, the coefficients $iS_{j,i}$ produced by the code implementation consist in matrices where the rows represent the observation times t and the columns the perturbation time τ and an intuitive possible graphical representation is constituted by the heatmap with rainbow colormap. Since iPSA coefficient matrices calculated for this analysis have huge dimension (1401x1401) and the computational effort for the plot is very heavy, heatmaps were generated only for some selected parameters and state variables, considered interesting regarding PSA analysis results and biological meanings. The chosen parameters are q_2 , λ_2 , f_1 , S_b , E_2^{th} , δ_2^{min} , k_{2s}^{min} and the state variables that were considered for this analysis are x , mE_2 and E_2 . This choice is justified by the importance of E_2 gene regulatory network, from primary transcript x to transcript mE_2 to E_2 protein. The chosen parameters are of paramount importance for this network, in particular q_2 and λ_2 address the negative feedback of E_2 for the regulation of x transcription, S_b accounts for the basal transcription of x , E_2^{th} corresponds to the value of E_2 that turns the feedback from positive to negative and vice versa, f_1 and k_{2s}^{min} are interesting in the optics of investigating E_1 positive regulation of mE_2 splicing flux while δ_2^{min} is interesting for the negative regulation operated by E_1 in the degradation of E_2 .

When heatmaps were produced with both numerical and analytical approach, smooth colored images were expected, but it was evident that some oscillation were present as depicted in Figure 5.23 and 5.24.

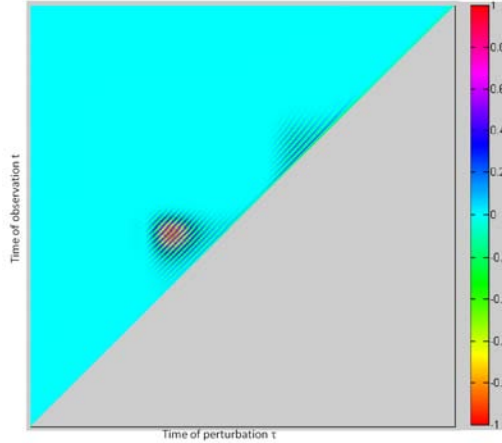


Figure 5.23: Heatmap illustrating iPSA coefficients ($\frac{\partial E_2(t)}{\partial q_2(\tau)}$) of state variable E_2 with respect to parameter q_2 (numerical approach). The x-axis gives the time at which impulse perturbation ($\tau \in [0 - 14]days$) is applied to the parameter, while the y-axis indicates the observation time t ($t \in [0 - 14]days$). The plot is scaled to have values between -1 and +1.

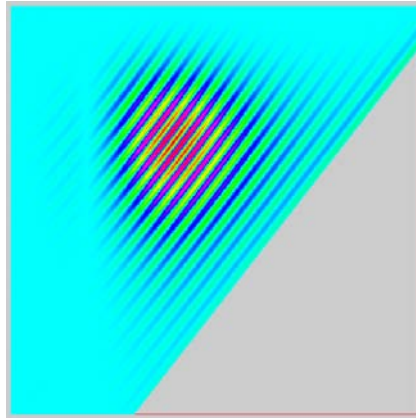


Figure 5.24: Particular of Heatmap presented in Figure 5.23 illustrating iPSA coefficients of state variable E_2 for the parameter q_2 (numerical approach).

Since similar oscillations are present in all heatmaps, the origin of this behavior was investigated. To better describe the characteristics of these oscillations, let's fix the time of observation at $t = 6.41$ day (see Figure 5.25) and plot the time course curve of iPSA coefficient as a function of τ . The curve is depicted in Figure 5.26.

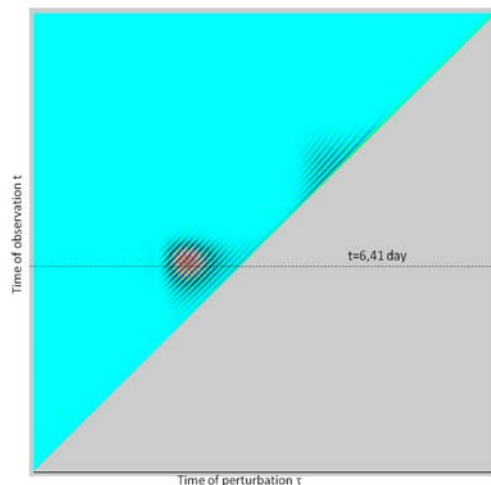


Figure 5.25: Selection of $t = 6.41$ day in Figure 5.23 illustrating iPSA coefficients of state variable E_2 for the parameter q_2 (numerical approach).

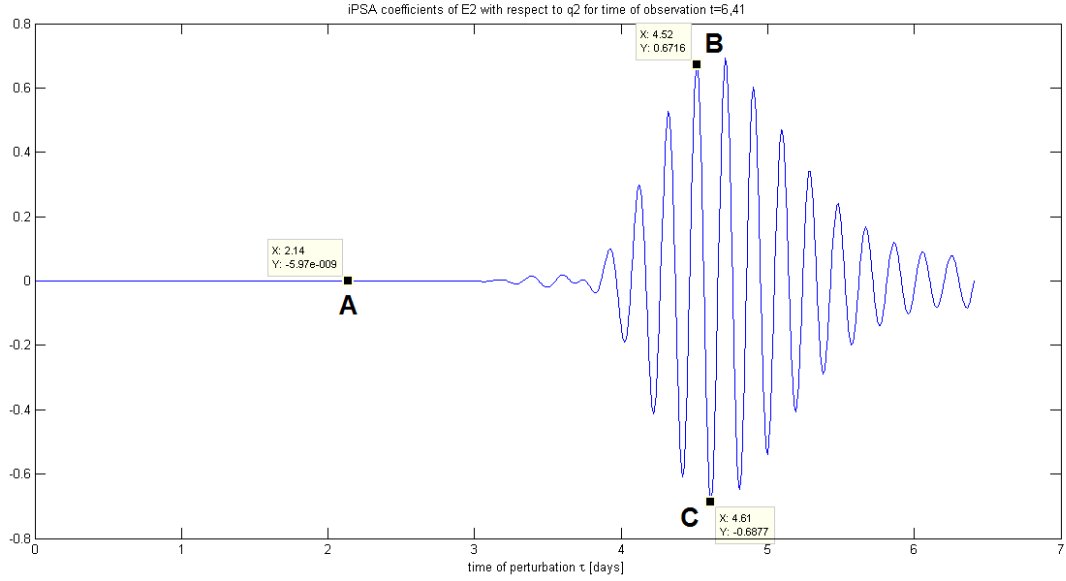


Figure 5.26: iPSA analysis for E_2 with respect to parameter q_2 (numerical approach). The curve illustrates the iPSA coefficients at different perturbation times τ for time of observation $t = 6.41$ day.

It seems that for early times of perturbation, no effect is visible and iPSA coefficients have zero value while from $\tau \approx 3 - 4$ to $\tau \approx 6.5$ day the systems reacts with oscillating behavior. This means that, when perturbing parameter q_2 before day 3, E_2 perturbed curve should show no differences with the nominal one, however when perturbing parameter q_2 between 3 and 6.5 days some variations should be observed. Moreover oscillations appear to have a period of ≈ 0.2 days. In order to investigate this behavior three times of perturbation were selected from Figure 5.26: $\tau_1 = 2.14$, $\tau_2 = 4.52$, $\tau_3 = 4.61$ day. The three τ correspond to three cases:

- case **(A)** $\tau_1 = 2.14$ day: iPSA coefficient is zero (steady region)
- case **(B)** $\tau_2 = 4.52$ day: iPSA coefficient = 0.6716, it belongs to a positive peak (oscillating region)
- case **(C)** $\tau_3 = 4.61$ day: iPSA coefficient = -0.6877 , it belongs to a negative peak (oscillating region)

E_2 nominal and perturbed curves (in the three times of perturbation) were plotted together for comparison as depicted in Figure 5.27.

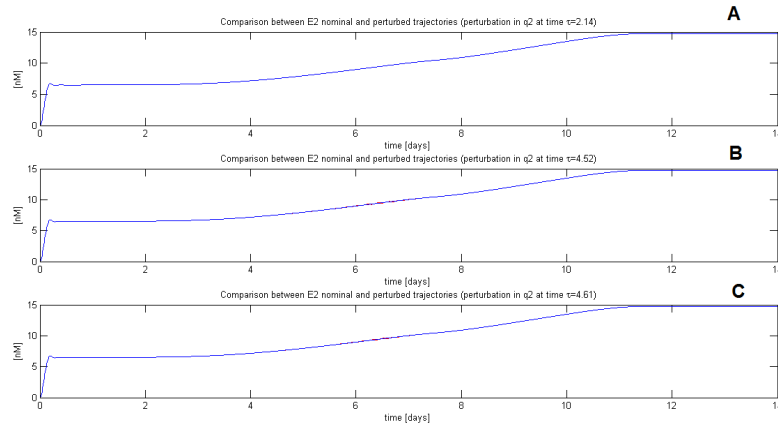


Figure 5.27: Comparison between E_2 nominal (blue) and perturbed curve (red) for three perturbation times (A) $\tau_1 = 2.14$, (B) $\tau_2 = 4.52$, (C) $\tau_3 = 4.61$ day.

As expected a perturbation at time previous to day 3 as in case (A) produces no effect on E_2 trajectory while perturbations after day 3 produce a perturbed E_2 trajectory that presents oscillatory behavior (cases (B) and (C)). Let's zoom around the perturbation times and the perturbed part for cases (B) and (C) obtaining the following figures.

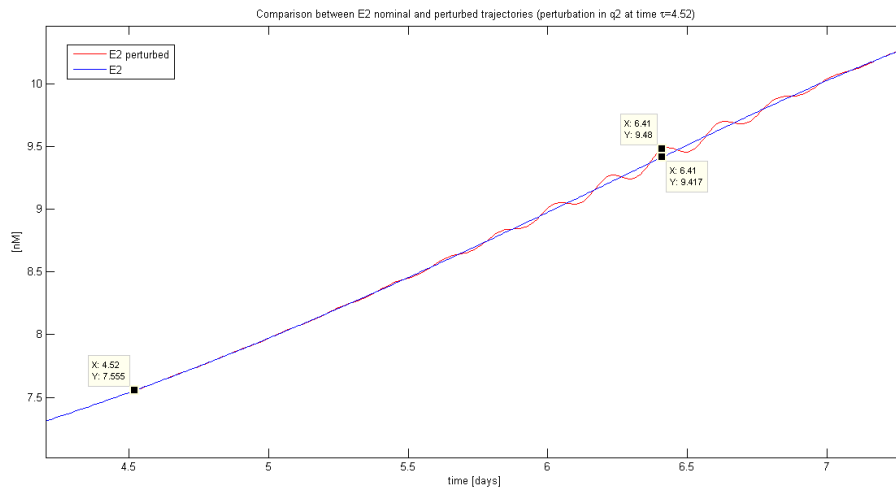


Figure 5.28: Particular of Figure 5.27 (B) illustrating the comparison between nominal E_2 trajectory and perturbed trajectory.

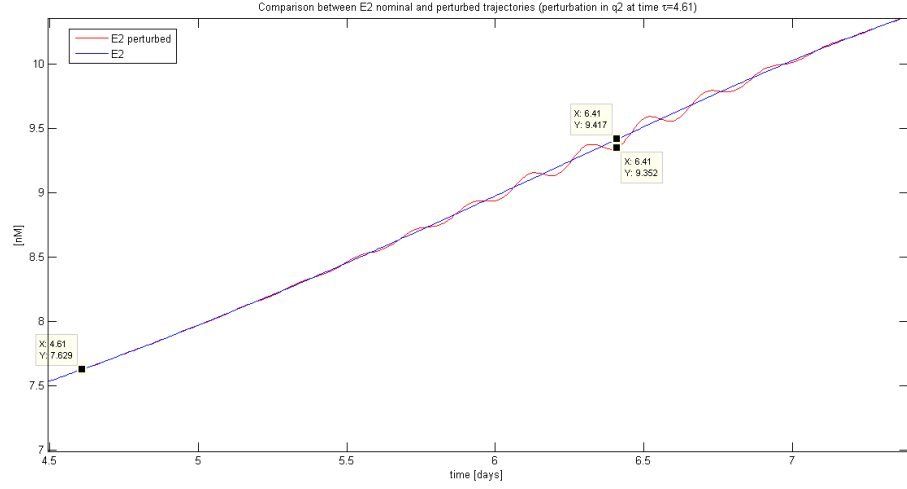


Figure 5.29: Particular of Figure 5.27 (C) illustrating the comparison between nominal E_2 trajectory and perturbed trajectory.

It is evident that the state trajectories have an oscillating behavior consequent to the impulse perturbation even if these oscillation do not appear immediately at the time of perturbation. In fact the higher variations are concentrated in the region going from $\tau \approx 5.5 - 6$ to $\tau \approx 7.5$. This region correspond to the τ values of the coloured region present in the heatmaps. These considerations suggest that the oscillation are originated by the system reaction to the stimulus that is consequently re-conducted to the iPSA coefficients. In order to effectively prove this hypothesis one needs to compare the variation between perturbed E_2 and nominal E_2 for case (B) and (C) at the time corresponding to the observation time $t = 6.41$ day. In particular, selecting $t = 6.41$ in the curves of Figure 5.28 and 5.29 one obtains:

- (B) $\tau_2 = 4.52$ day and $t = 6.41$ day:

$$E_2^\Delta - E_2 = 9.48 - 9.417 = 0.063nM.$$

- (C) $\tau_3 = 4.61$ day and $t = 6.41$ day:

$$E_2^\Delta - E_2 = 9.352 - 9.417 = -0.065nM.$$

where E_2^Δ refers to E_2 perturbed values. A correlation between these two values and the values of iPSA coefficients in case (B) and (C) respectively (Figure 5.26) should be found. Looking at the way in which iPSA coefficients

were calculated one can see that they correspond to the values of $E_2^\Delta(t) - E_2(t)$ up to a multiplying constant ($\frac{1}{\Delta p_j}$) and to the normalization factor:

$$iS_{i,j}(t, \tau) = \frac{E_2^\Delta(t) - E_2(t)}{\Delta p_j} \cdot \frac{p_j}{E_2(t)}$$

Hence dividing the values of iPSA coefficients for $\frac{1}{\Delta p_j} \cdot \frac{p_j}{E_2(t)}$ one should obtain the variation between E_2 perturbed and nominal values. Dividing iPSA coefficients of cases **(B)** and **(C)** for this factor, calculated at the time of observation $t = 6.41$ day, one obtains:

- case **(A)**:

$$\frac{iPSA(6.41; 4.52)}{\frac{p_j}{\Delta p_j \cdot E_2(6.41)}} = \frac{0.6716}{10.6192} = 0.063nM$$

- case **(B)**:

$$\frac{iPSA(6.41; 4.61)}{\frac{p_j}{\Delta p_j \cdot E_2(6.41)}} = \frac{-0.6877}{10.6192} = -0.065nM$$

that are exactly the two variations between E_2 perturbed and nominal value in $t = 6.41$ day. Hence the study demonstrated that the oscillations of the heatmaps and iPSA coefficients are derived from the behavior of the state variables after the perturbation and the magnitude of iPSA coefficients is dependent from the variation of the state variable and from the applied normalization. From this point of view iPSA revealed at first an important property of the model: it has an oscillatory behavior under the application of an impulsive stimulus.

In the following iPSA results with the two approaches are shown for state variables x , mE_2 and E_2 with respect to parameters q_2 , λ_2 , f_1 , S_b , E_2^{th} , δ_2^{min} , k_{2s}^{min} .

5.4 iPSA results with numerical approach

In the following, heatmaps representations with values normalized between $[-1, 1]$ are depicted for the parameters of interest for the state variables x , mE_2 and E_2 . From Figure 5.31, 5.32 and 5.33 it appears that in all heatmaps non zero values are concentrated in a more or less rounded region (with different shapes depending on the state variable and parameter) centered in $(t, \tau) \approx (6day, 4.5day)$. In the case of parameter q_2 , there is the presence

of a region different from zero also around $(t, \tau) \approx (9\text{day}, 9\text{day})$ as depicted in Figure 5.30.

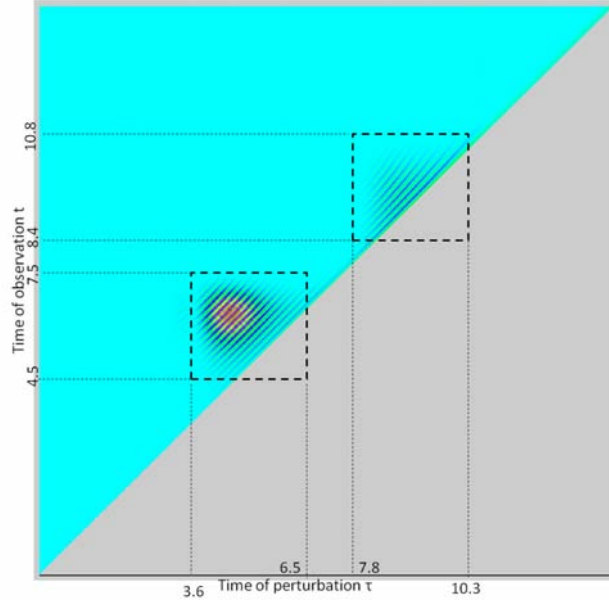


Figure 5.30: Heatmap illustrating iPSA coefficients of state variable E_2 with respect to parameter q_2 (numerical approach). The two regions with values different from zero are defined in their time ranges.

In particular the first regions are all characterized by higher values of iPSA coefficients inside the rounded part that become smaller in the borders.

The results suggest that, for every parameter, if the perturbation is applied from $\tau \approx 3.5$ day to $\tau \approx 5.5$ (and also from $\tau \approx 7.8$ to $\tau \approx 10.3$ for q_2) the model reacts to the perturbation, however no reactions are visible. The reason why the regions are located exactly between that values of τ is enigmatic and it results difficult to give an interpretation. Looking at state variable predictions (Figure 3.4) of the model it seems that around $t = 2 - 3$ days the state variables leave their steady state however it is not possible to directly re-conduct this behavior to iPSA results after time 3 days. From the biological point of view, there is a possible interpretation related to the time positions of the rounded regions centered more or less at $\tau = 4.5$ day. From the study of article [18], day 4 is demonstrated to be the first day in which raft tissue culture systems show the starting of viral DNA amplification, that begins to increase until reaching a peak around day 10-12. Regarding the other colored region that appears in the heatmaps of q_2 , it is located around days 8 - 10, days that are assumed to be the peak of vDNA amplification.

Hence, a possible interpretation is that the corresponding parameter play a considerable role also in the process of increasing amplification to reach the peak value. However these aspects need to be deeply analyzed and further study are needed for better elucidate the real behavior of the model and for give a better interpretation of the results. Despite the previous considerations, it is important to notice that, for all state variables, the parameters with higher iPSA coefficients values are in the order E_2^{th} , λ_2 , S_b , q_2 , confirming the importance, among all, of the parameters that play a role in E_2 regulation for transcription and partially for viral replication processes.

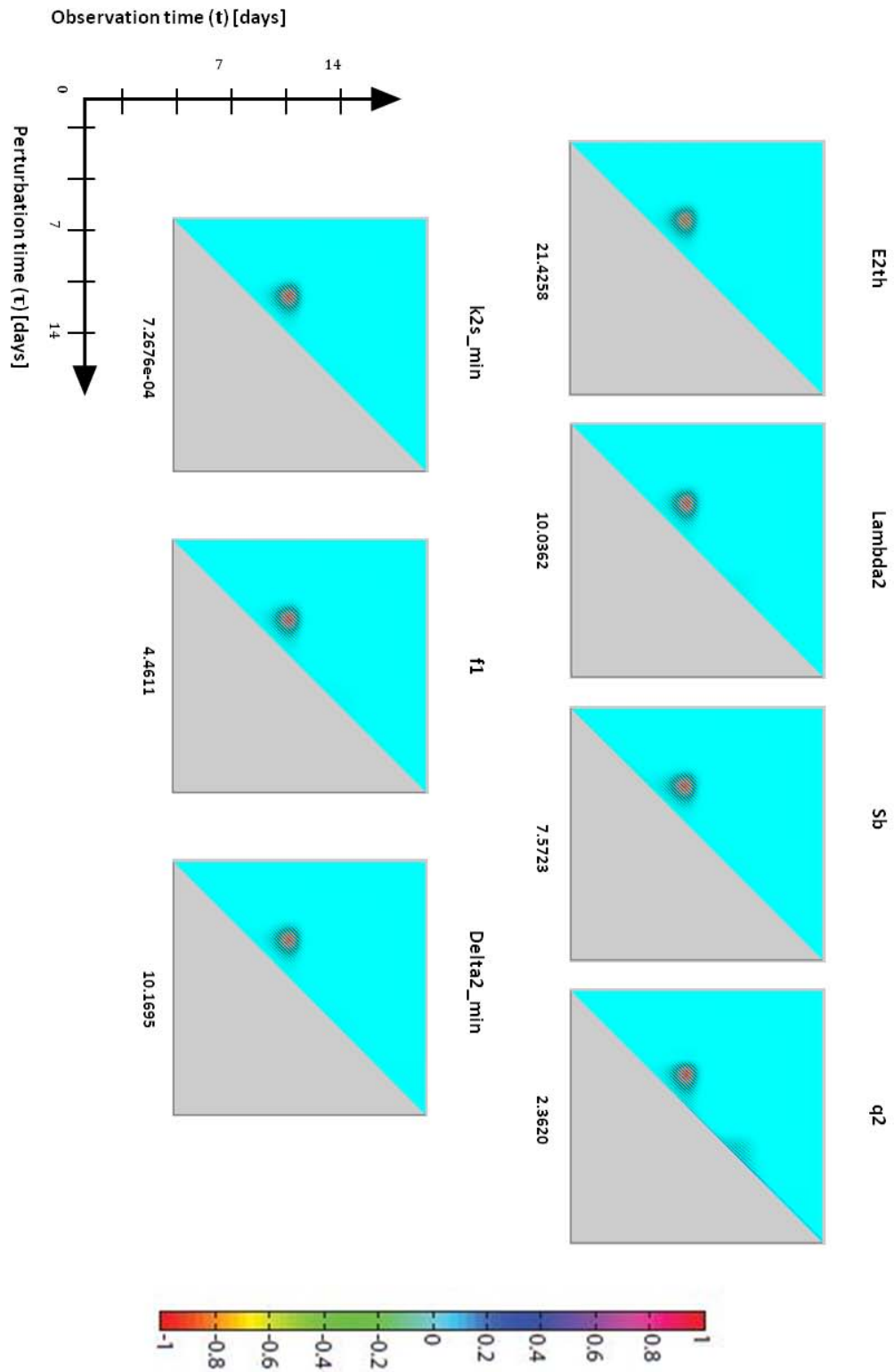


Figure 5.31: iPSA analysis (numerical approach) for state variable x . Each heatmap illustrates the iPSA coefficient of x with respect to perturbations on one parameter in the network, indicated in the title. The x-axis gives the time at which impulse perturbation is applied to the parameter, while the y-axis indicates the observation time x . For comparison purpose, each plot is scaled to have values between -1 and +1 by the scaling factor reported in the abscissa.

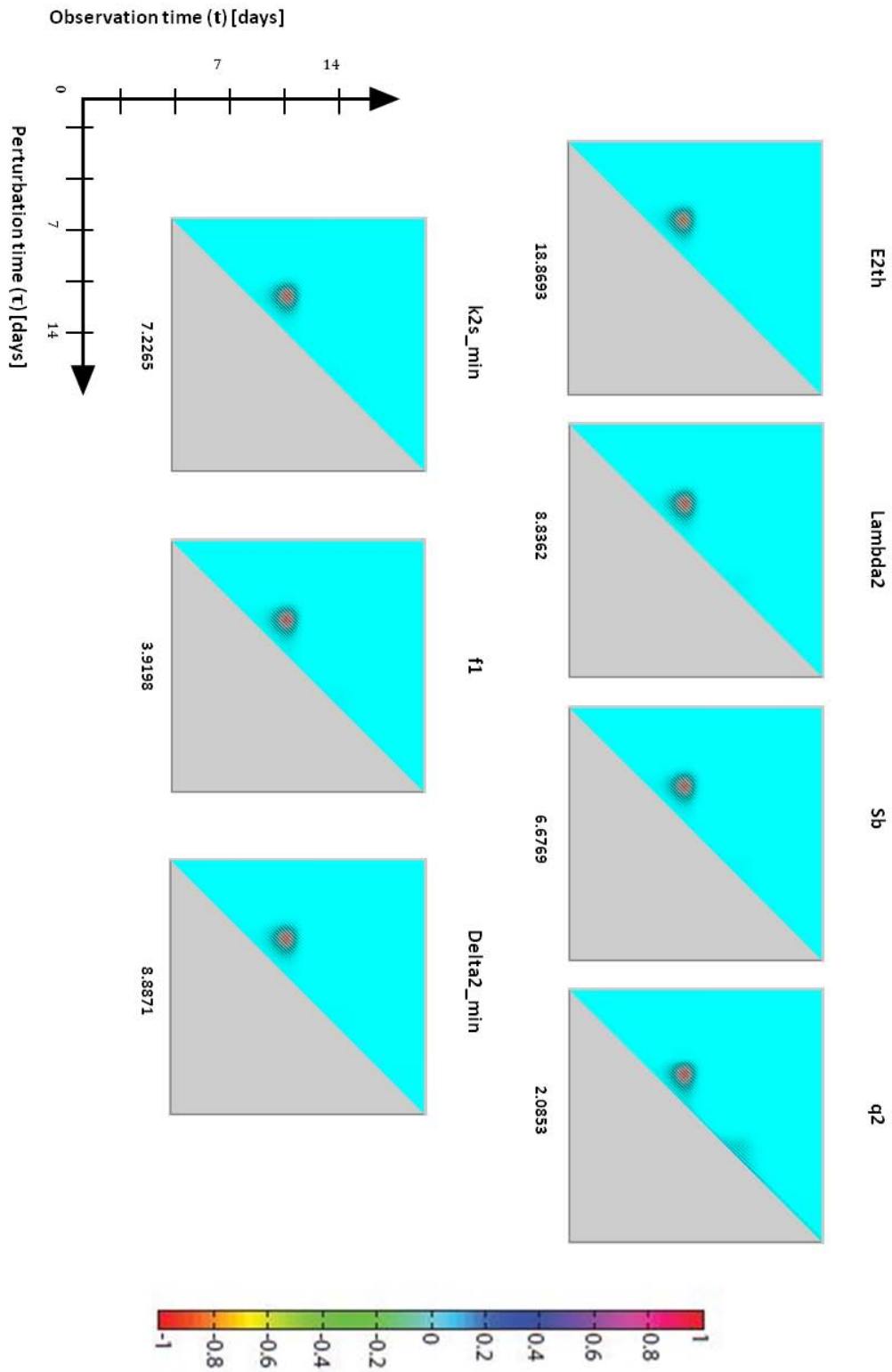


Figure 5.32: iPSA analysis (numerical approach) for state variable mE_2 . Each heatmap illustrates the iPSA coefficient of x with respect to perturbations on one parameter in the network, indicated in the title. The x-axis gives the time at which impulse perturbation is applied to the parameter, while the y-axis indicates the observation time mE_2 . For comparison purpose, each plot is scaled to have values between -1 and +1 by the scaling factor reported in the abscissa.

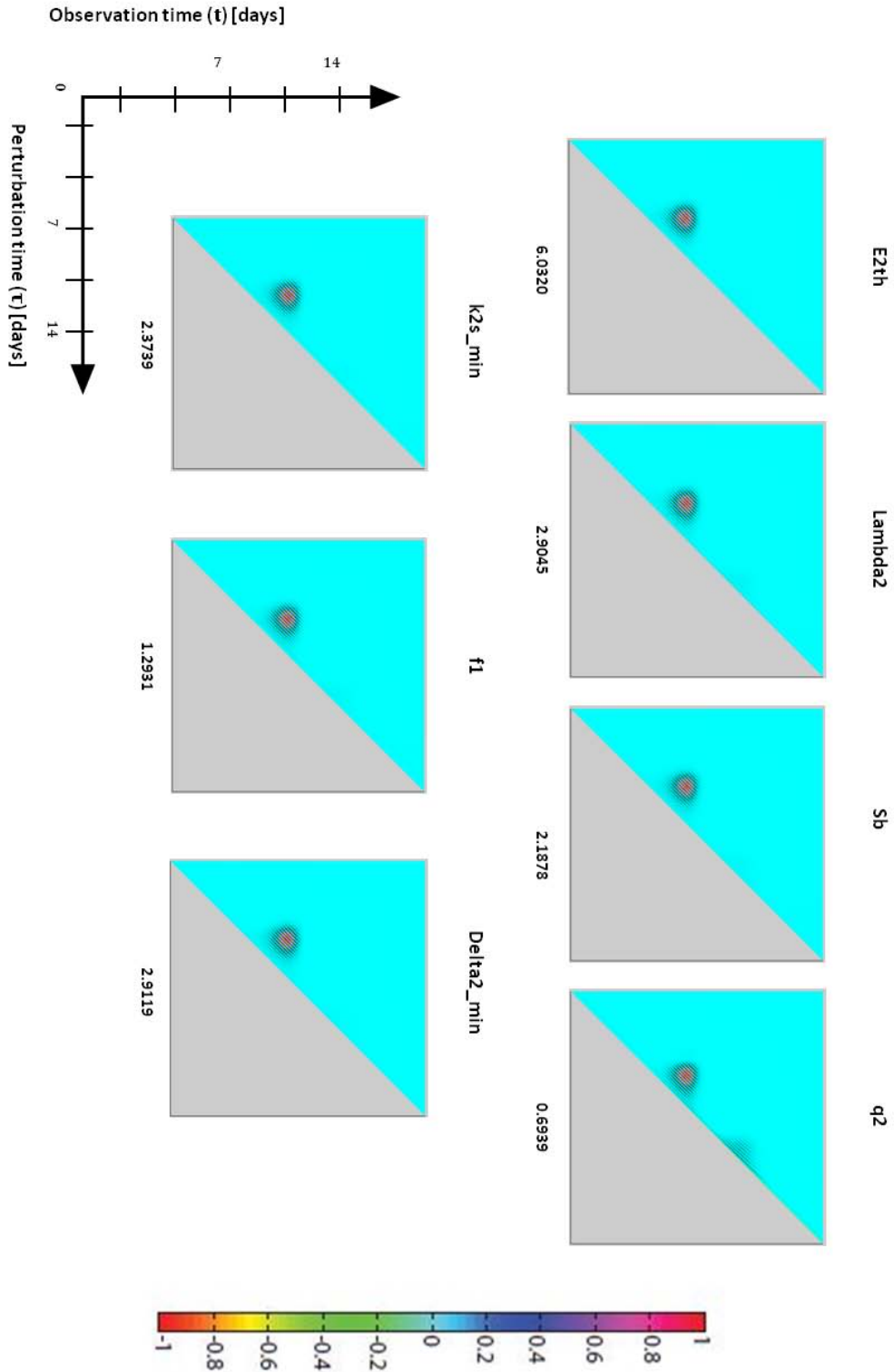


Figure 5.33: iPSA analysis (numerical approach) for state variable E_2 . Each heatmap illustrates the iPSA coefficient of x with respect to perturbations on one parameter in the network, indicated in the title. The x-axis gives the time at which impulse perturbation is applied to the parameter, while the y-axis indicates the observation time E_2 . For comparison purpose, each plot is scaled to have values between -1 and +1 by the scaling factor reported in the abscissa.

5.5 iPSA results with analytical approach

In the following, heatmaps representations with values normalized between $[-1, 1]$ are depicted for the parameters of interest for the state variables x , mE_2 and E_2 . From Figure 5.34, 5.35 and 5.36 it appears that the heatmaps representations are similar to the ones obtained with numerical approach but present some important differences. Oscillations are still present confirming the oscillatory nature of the model under impulsive stimulus. In this case not only parameter q_2 shows two coloured regions in the heatmaps but also k_{2s}^{min} , f_1 and δ_2^{min} and only for state variable x . Moreover the bigger difference regards the normalization coefficients, that are much higher than the ones of the previous approach in the case of mE_2 and E_2 state variables. In other words, analytical approach produced very high iPSA coefficients that are not close to the ones of numerical approach. It is necessary to underline that the two approaches should not lead to identical results since in numerical approach the perturbation is set as 1% of the parameter nominal value while in analytical approach as 1% of the maximum value of each state variable but however no such differences were expected between the two analysis. For sure further study for determining which perturbation magnitudes should be set to obtain similar results must be conducted.

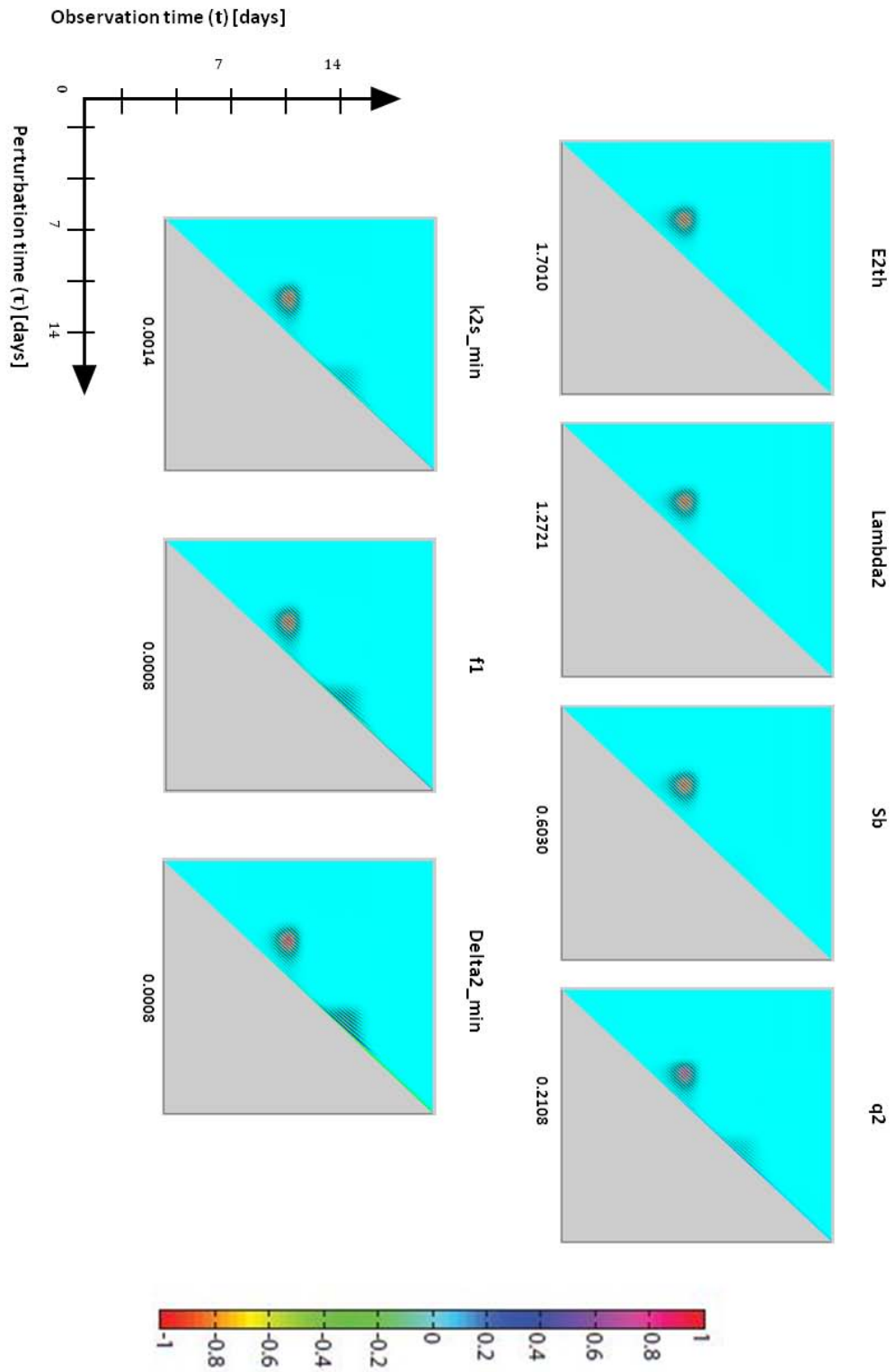


Figure 5.34: iPSA analysis (analytical approach) for state variable x . Each heatmap illustrates the iPSA coefficient of x with respect to perturbations on one parameter in the network, indicated in the title. The x-axis gives the time at which impulse perturbation is applied to the parameter, while the y-axis indicates the observation time x . For comparison purpose, each plot is scaled to have values between -1 and +1 by the scaling factor reported in the abscissa.

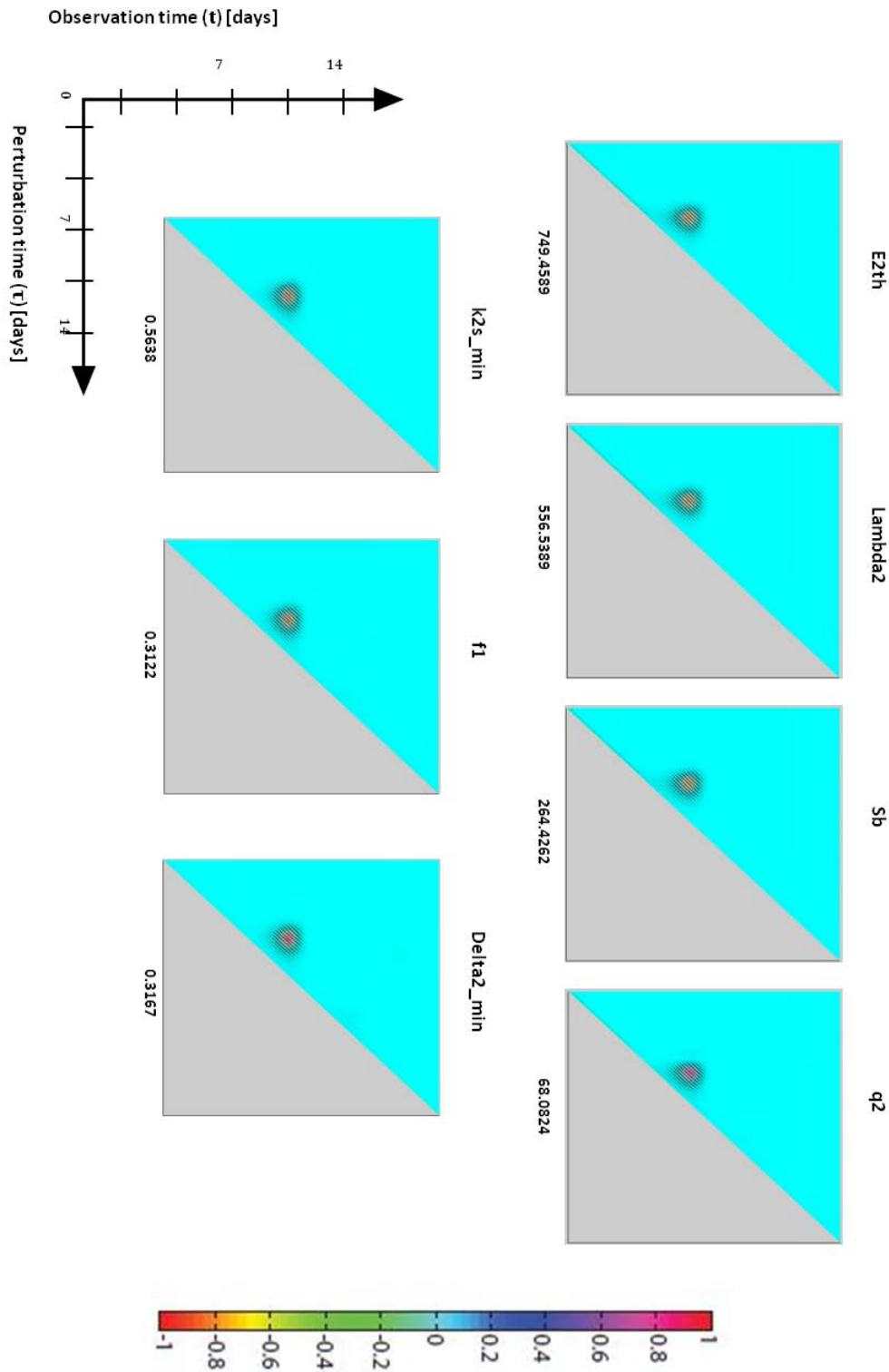


Figure 5.35: iPSA analysis (analytical approach) for state variable mE_2 . Each heatmap illustrates the iPSA coefficient of x with respect to perturbations on one parameter in the network, indicated in the title. The x-axis gives the time at which impulse perturbation is applied to the parameter, while the y-axis indicates the observation time mE_2 . For comparison purpose, each plot is scaled to have values between -1 and +1 by the scaling factor reported in the abscissa.

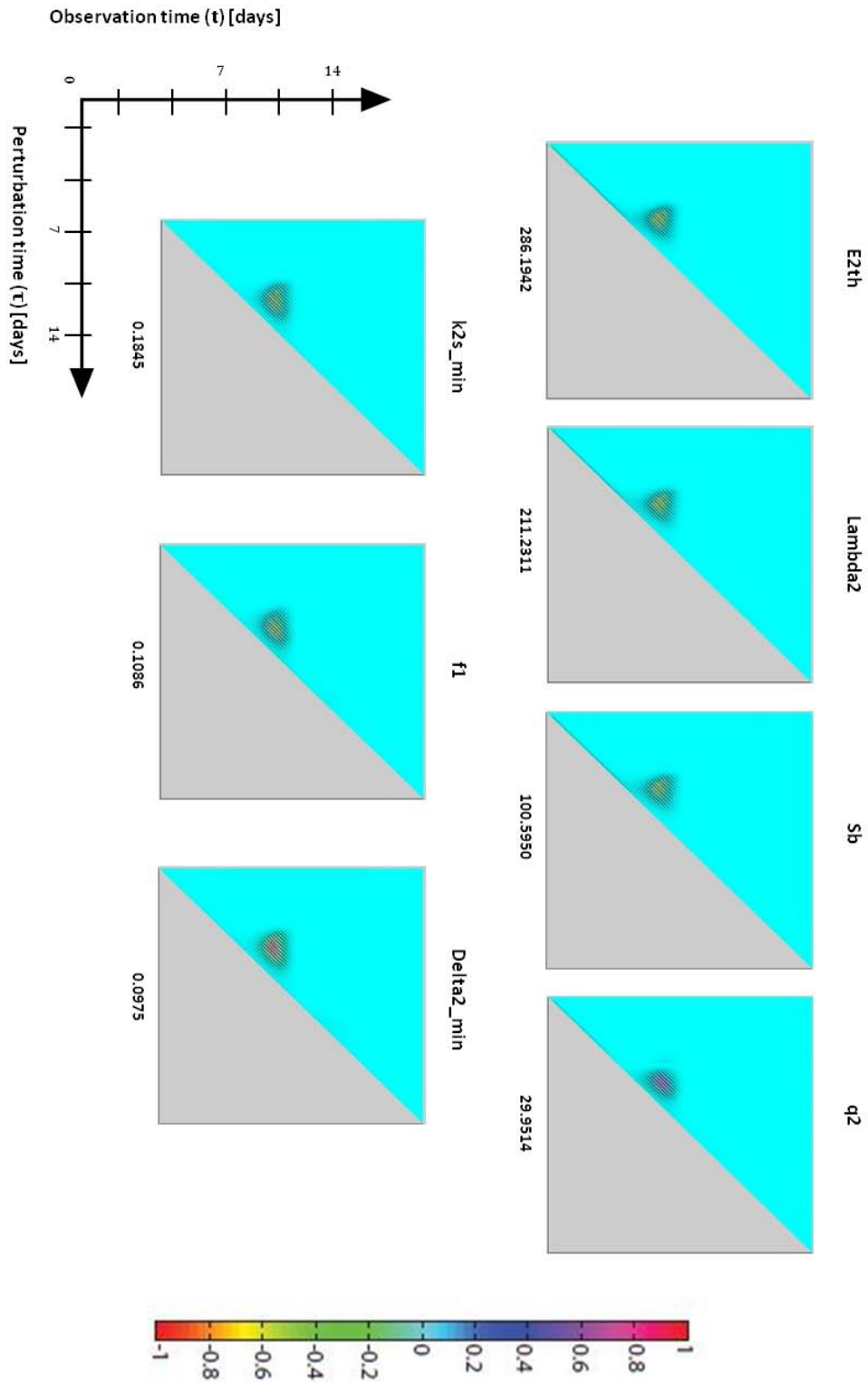


Figure 5.36: iPSA analysis (analytical approach) for state variable E_2 . Each heatmap illustrates the iPSA coefficient of x with respect to perturbations on one parameter in the network, indicated in the title. The x-axis gives the time at which impulse perturbation is applied to the parameter, while the y-axis indicates the observation time E_2 . For comparison purpose, each plot is scaled to have values between -1 and +1 by the scaling factor reported in the abscissa.

Chapter 6

Conclusions

In this work different sensitivity methods of conventional PSA were investigated along with the novel sensitivity analysis iPSA. Each method was presented along with advantages and disadvantages which were then explored in a real application of the HPV early promoter regulation model.

As regards conventional PSA, Direct Method and Indirect Method were compared considering results reliability, complexity of implementation and time of computation. In the case under examination DM appeared to be more adequate concerning results accuracy and complexity of implementation even if it is important to underline that the model under consideration has not a high complexity. DM sensitivity results were in accordance with literature from the biological point of view and no contradictory revelations appeared. In particular conventional PSA showed the importance of the parameters belonging to the transactivation/repression feedback operated by protein E_2 on the transcription of the primary transcript x . The results revealed the dominance of the parameters belonging to the negative feedback, accordingly to biological knowledge that assesses that repression feedback is stronger than transactivation one.

iPSA application to the HPV model was performed only for the most interesting model parameters and the analysis was applied using two approaches, numerical and analytical. Both matlab implementations resulted to be computationally heavy giving the need to produce huge size matrices of data with different ode solver applications. Hence further study will have the aim of finding a better computation method for iPSA. Moreover heatmaps (with values scaled between $[-1,1]$) were produced to represent two dimension iPSA data which presented oscillations thus revealing the oscillatory nature of the model under impulsive stimulus. This behavior was confirmed by investigating the time course of the state variables after the impulse perturbation and verifying that the perturbed trajectories were actually characterized by

oscillations after the applied perturbation. All heatmaps presented non zero values around a specific located area, suggesting that all state variables are sensitive to parameter changes only during that time range. This aspect didn't appear with conventional PSA and needs to be deeply investigate since it could give important information about the behavior of the model during this time range. Moreover, both approaches produced similar normalized heatmap representations with the exception of their scaling factor that were much higher for analytical approach. A likely reason is that the levels of perturbation that were introduced were not suitable for obtaining similar results giving the fact that in numerical approach the perturbation is applied directly to the parameter while in analytical approach to the state variable. From the biological point of view the results were partially in accordance with conventional PSA: again there was a confirm of the importance of the parameters belonging to E_2 transcriptional feedback but high sensitivities were shown also for the parameters acting inside the positive feedback. Further analysis and studies are mandatory in order to elucidate the problems faced during iPSA matlab implementation and iPSA result interpretation both for the model under study and for other general applications.

Besides the need of further investigations, the sensitivity analysis operated in this study underlined that iPSA can be considered an interesting complementary analysis for conventional PSA.

Acronyms

DM Direct Method. 4, 6, 8, 9, 25, 31, 32, 37, 38, 42, 43, 50, 55, 58, 75, 76, 83, 86, 87

DQM Different Quotient Method. 4, 25, 26, 31, 32, 38, 43, 76, 83, 85

GFM Green's Function Method. 4, 9, 10, 76

GFm Green's Function Matrix. 8, 12, 28, 76

HPV Human Papillomavirus. 4, 76

iPSA Impulse Parametric Sensitivity Analysis. 4, 5, 31, 59–62, 64–74, 76

LSA Local Sensitivity Analysis. 4, 5, 76

PSA Parametric Sensitivity Analysis. 1, 4, 76

vDNA Viral DNA. 38, 66, 76

Appendices

Chapter A

Green's function method derivation

Green's Function general definition

A Green's function, $G(t, \tau)$, of a linear differential operator $L = L(t)$ acting on distributions over a subset of the Euclidean space \mathbf{R}^n , at a point τ , is any solution of

$$LG(t, \tau) = \delta(t - \tau) \quad (\text{A.1})$$

where δ is the Dirac delta function. If the Kernel of L is non-trivial the Green's function is not unique, however in practice, some combination of symmetry and boundary conditions will give an unique Green's function. This property of a Green's function can be exploited to solve differential equations of the form

$$Lu(t) = f(t) \quad (\text{A.2})$$

If such a function G can be found for the operator L , then first multiplying the equation (A.1) by $f(\tau)$, and then performing an integration in the τ variable, the following equation is obtained [19]:

$$\int LG(t, \tau)f(\tau)d\tau = \int \delta(t - \tau)f(\tau)d\tau = f(t) \quad (\text{A.3})$$

Using equation (A.3) into equation (A.2) it follows that

$$Lu(t) = \int LG(t, \tau)f(\tau)d\tau. \quad (\text{A.4})$$

Green's function applied to sensitivity analysis

Let's consider the generic ODE model described by Equation 2.1

$$\begin{cases} \frac{d\mathbf{x}(t, \hat{\mathbf{p}})}{dt} = \mathbf{f}(\mathbf{x}, \hat{\mathbf{p}}) \\ \mathbf{x}(t_0, \hat{\mathbf{p}}) = \mathbf{x}^0 \end{cases}$$

where \mathbf{x} is the n -vector of concentrations ($x \in \mathbb{R}^n$) and \mathbf{p} is the m -vector of system parameters ($\hat{\mathbf{p}} \in \mathbb{R}^m$) with initial conditions $\mathbf{x}(t_0, \hat{\mathbf{p}}) = x^0$. The first-order sensitivity coefficients can be obtained differentiating the ODE model equations in Equation 2.1 with respect to the state vector $\mathbf{x}(\tau)$, obtaining the following equation:

$$\frac{d}{dt} \frac{\partial x_i(t)}{\partial x_j(\tau)} = \mathbf{J}(t) \cdot \frac{\partial x_i(t)}{\partial x_j(\tau)} \quad i, j = 1, \dots, n, \quad (\text{A.5})$$

where $\mathbf{J}(t) = \frac{\partial \mathbf{f}}{\partial \mathbf{x}(t)}$ is the Jacobian, τ is the time of perturbation and t the observation time ($t \geq \tau$). Equation (A.5) can be re-written in matricial form as follows

$$\frac{d}{dt} \mathbf{S}^{\mathbf{x}}(t, \tau) = \mathbf{J}(t) \mathbf{S}^{\mathbf{x}}(t, \tau) \quad (\text{A.6})$$

where $\mathbf{S}^{\mathbf{x}}(t, \tau)$ is known as Green's function matrix (GFm) or Kernel and contains the elements $\mathbf{S}_{i,j}^{\mathbf{x}}(t, \tau) = \frac{\partial x_i(t)}{\partial x_j(\tau)}$ and \mathbf{J} is the Jacobian matrix. Solving Equation A.6 is equivalent to solve the homogenous part of Equation 2.6 of the DM. Since eq. (2.6) is a linear inhomogeneous equation, its solution can be obtained by firstly solving the homogeneous part (Equation A.6) and then determining the particular solutions for each parameter through a linear integral transform of the nonhomogeneous term $\frac{\partial f}{\partial \mathbf{p}(\tau)}$ [1] [19]:

$$\frac{\partial \mathbf{x}(t)}{\partial p_j(\tau)} = \int_{\tau}^t \mathbf{S}^{\mathbf{x}}(t, s) \frac{\partial f(s)}{\partial p_j} ds. \quad (\text{A.7})$$

When conducting PSA with time of perturbation equal to the initial time t_0 Equation A.7 becomes:

$$\frac{\partial \mathbf{x}(t)}{\partial p_j} = S_{[1..n],j}(t, t_0) = \mathbf{S}^{\mathbf{x}}(t, t_0) \cdot \frac{d\mathbf{x}(t_0)}{dp_j} + \int_{t_0}^t \mathbf{S}^{\mathbf{x}}(t, s) \frac{\partial f(s)}{\partial p_j} ds \quad (\text{A.8})$$

where

$$\frac{d\mathbf{x}(t_0)}{dp_j} = \begin{cases} \delta_i & p_j \text{ is an initial condition for one of the components } x_i \\ 0 & \text{otherwise} \end{cases} \quad (\text{A.9})$$

where δ_i is the vector $\delta_i = [0, 0, \dots, 1, 0, \dots, 0]$ with 1 at the i -th position [8]. There are several variants of the Green function method and they differ from each other in the calculation of the matrix $S^{\mathbf{x}}(t, \tau)$. Hwang et al. (1978) [8] proposed to express the Green's function in terms of an adjoint Green's function, $\hat{\mathbf{S}}^{\mathbf{x}}(\tau, t)$, that solves the following equation:

$$\frac{d}{dt} \hat{\mathbf{S}}^{\mathbf{x}}(\tau, t) = -\hat{\mathbf{S}}^{\mathbf{x}}(\tau, t) \cdot \mathbf{J}(\tau) \quad 0 \leq \tau \leq t \quad (\text{A.10})$$

with

$$\hat{\mathbf{S}}^{\mathbf{x}}(t, t) = \mathbf{I} \quad (\text{A.11})$$

It is possible to demonstrate that $\mathbf{S}^{\mathbf{x}}(t, \tau) = \hat{\mathbf{S}}^{\mathbf{x}}(\tau, t)$ and instead of solving equation (A.7) or (A.8), one can solve (A.10) and obtain $\hat{\mathbf{S}}^{\mathbf{x}}(\tau, t)$. However (A.10) only gives $\hat{\mathbf{S}}^{\mathbf{x}}(\tau, t)$ for a certain fixed t . In order to obtain results for different times t the group property can be applied thus $\hat{\mathbf{S}}^{\mathbf{x}}(t_3, t_1) = \hat{\mathbf{S}}^{\mathbf{x}}(t_3, t_2)\hat{\mathbf{S}}^{\mathbf{x}}(t_2, t_1)$ where $t_1 < t_2 < t_3$.

Chapter B

MATLAB code implementations

In this chapter MATLAB codes used for the implementation of the three methods of sensitivity analysis are presented. To produce state variables trajectories, a solver for differential equations is needed and in this case ODE45 MATLAB function was used. It consists in a medium order method able to solve nonstiff differential equations. This method solves differential equations with a variable time step for efficient computation and it needs as arguments an ODE function, the vector of initial condition and if desired, the *options*. The *options* set for this analysis were:

- AbsTol: absolute error tolerances that apply to the individual components of the solution vector. It was set with value= 1e-3.
- MaxStep: upper bound on solver step size. It was always set equal to the time span used for the simulation time.

Trajectories vectors need then to be re-conducted to the time sequence corresponding to the simulation time through the use of the method "*interp1*", a linear interpolation calculator that returns interpolated values of a 1-D function at specific query points.

It is evident that each method requires at the beginning the creation of the vector of the system parameters and the creation of the model, in particular DQM and iPSA need a common initial code while DM solves system equations and sensitivity calculation simultaneously. DQM and iPSA common initial code for the creation of model equations and calculation of concentration-time curves of the state variables is summarized below.

```
1 %%structure containing model parameters nominal values
3 struct.Sb=10; %[nm/min]...
```

```

...
5 struct.beta2=2e-2; %[1/min]

7 %%creation of time vector [0-14 days]
Tspan=0.01*24*60;
9 tfine=14*24*60; %end simulation time (min)
t=[0:Tspan:tfine]; %minutes
11 n=length(t);
tplot=[0:0.01:14]; %days
13
%%initial conditions
15 y0=[0 0 0 0 0];

17 %%options for ODE45
options=odeset('AbsTol', 1e-3, 'MaxStep', Tspan);
19
%%calculation of model state variable concentration-time curve
21 [T Y]=ode45('mod_HPV', [t(1) t(end)], y0, options, struct);
x=interp1(T, Y(:, 1), t);
23 mE1=interp1(T, Y(:, 2), t);
mE2=interp1(T, Y(:, 3), t);
25 E1=interp1(T, Y(:, 4), t);
E2=interp1(T, Y(:, 5), t);

```

Listing B.1: MATLAB main section for the creation of the model.

The model equations are implemented in the function "*mod_HPV*" that needs to be set as a argument for the ODE solver. The code of "*mod_HPV*" is shown below.

```

1 function dy=mod_HPV(T,y,FLAG,struct)
3 %% parameters depending on state variables concentrations
%% k_{2s}(E_1)
5 k2s_E1=((struct.f1-1)*struct.k2s_min)/(1+exp((struct.lambda5-y(4))/struct.
sigma1))+struct.k2s_min;
%% \delta_{2p}(E_1)
7 delta2p_E1=((struct.f2-1)*struct.delta2_min)*(1-1/(1+exp((struct.lambda6-y
(4))/struct.sigma2)))+struct.delta2_min;
%% S_x(E_2)
9 if y(5) <= struct.E2th
SxE2=struct.Sb +(struct.a1*(y(5))^struct.q1)/(struct.lambda1^struct.q1+(
y(5))^struct.q1);
11 else
struct.a2 = struct.Sb +(struct.a1*(struct.E2th)^struct.q1)/(struct.
lambda1^struct.q1+(struct.E2th)^struct.q1); %for the continuity of the
function!
13 SxE2=((struct.a2*struct.lambda2^struct.q2)/(struct.lambda2^struct.q2+(y
(5)-struct.E2th)^struct.q2));
end
15 %% k_{1s}(t)
if T <= struct.th
17 k1s=struct.k1s_min+(struct.a3*T^struct.q3)/(struct.lambda3^struct.q3
+T^struct.q3);
else
19 struct.a4 = struct.k1s_min+(struct.a3*struct.th^struct.q3)/(struct.
lambda3^struct.q3+struct.th^struct.q3); %for the continuity of the
function!
k1s=(struct.a4*struct.lambda4^struct.q4)/(struct.lambda4^struct.q4+(T-
struct.th)^struct.q4);

```

```

21 end
   %% model equations
23 dy(1)=SxE2-struct.ks*y(1); %x
   dy(2)=kis*struct.ks*y(1)-struct.delta1m*y(2); %mE1
25 dy(3)=k2s_E1*struct.ks*y(1)-struct.delta2m*y(3); %mE2
   dy(4)=struct.beta1*y(2)-struct.delta1p*y(4); %E1
27 dy(5)=struct.beta2*y(3)-delta2p_E1*y(5); %E2
   dy=dy';

```

Listing B.2: MATLAB mod_HPVP function for the implementation of model equations.

B.1 Difference quotient method

The code that was used for the DQM is depicted below with an example: the parameter q_2 for the state variable E_2 .

```

2  %% sensitivity coefficients-ks DIFFERENCE QUOTIENT METHOD
   % creation of the 3 magnitudes of perturbation 1%,5%,10%
4  dpvect=[struct.q2*0.01,struct.q2*0.05,struct.q2*0.1];
   m=length(dpvect);
6  E2_corr=E2(2:end);
   q2=struct.q2;
8
   %%calculation of the perturbed trajectories
10 for i = 1:m
       struct.q2=q2+dpvect(i);
12   [T Y]=ode45('mod_HPVP',[t(1) t(end)],y0,options,struct);
       E2_q2(i,:)=interp1(T,Y(:,5),t); %Y(:,5) because is for E2
14   E2_q2_corr(i,:)=E2_q2(i,2:end);
       s_q2(i,:)=((E2_q2_corr(i,:)-E2_corr)/dpvect(i)).*(q2./E2_corr); %
       normalized sensitivity curves
16 end
18 %integrated sensitivities (all perturbations)
   [k,1]=size(s_q2);
20   for i=1:k
       for j=1:1
22         S_q2_all_delta(i,1)=sum(s_q2(i,j));
       end
24   end

```

Listing B.3: MATLAB main section for the implementation of the DQM for parameter q_2 and for state variable E_2 .

B.2 Direct method

The necessary main code and ODE function are reported below for parameter q_2 and for state variable E_2 .

```

2 %% sensitivity coefficients q2- DIRECT METHOD
3 %calculation of state variables and sensitivity simultaneously
4 [T Y]=ode45('mod_direct_q2',[t(1) t(end)],y0,options,struct);
5 x=interp1(T,Y(:,1),t);
6 mE1=interp1(T,Y(:,2),t);
7 mE2=interp1(T,Y(:,3),t);
8 E1=interp1(T,Y(:,4),t);
9 E2=interp1(T,Y(:,5),t);
10
11 % sensitivity curves
12 S_q2=interp1(T,Y(:,10),t); %Y(:,10) because is for E2
13 S_q2=S_q2.*(struct.q2./E2); %normalized
14 %integrated sensitivity
15 for i=1:length(S_q2)
16     S_q2_dir=sum(S_q2(i));
17 end

```

Listing B.4: MATLAB main section for the implementation of DM for parameter q_2 and for state variable E_2 .

As it is evident from Listing B.4, each parameter needs a specific ODE function, in this case "mod_direct_q2" where the $2n$ differential equations are implemented. The code of the ODE function for q_2 for E_2 is presented below.

```

1 function dy=mod_direct_q2(T,y,FLAG,struct)
2 %parameters depending on state variables
3 k2s_E1=..
4 delta2p_E1=..
5 SxE2=..
6 k1s=..
7 %model equations
8 dy(1)=SxE2-struct.ks*y(1);
9 dy(2)=k1s*struct.ks*y(1)-struct.delta1m*y(2);
10 dy(3)=k2s_E1*struct.ks*y(1)-struct.delta2m*y(3);
11 dy(4)=struct.beta1*y(2)-struct.delta1p*y(4);
12 dy(5)=struct.beta2*y(3)-delta2p_E1*y(5);
13
14 %derivatives of E1 dependent parameters in E1
15 delta2p_der=-((struct.f2-1)*struct.delta2_min*((1/struct.sigma2*exp((struct.
16     lambda6-y(4))/struct.sigma2))))/((1+exp((struct.lambda6-y(4))/struct.
17     sigma2))^2);
18 k2s_der=((struct.f1-1)*struct.k2s_min*((1/struct.sigma1*exp((struct.lambda5-
19     y(4))/struct.sigma1))))/((1+exp((struct.lambda5-y(4))/struct.sigma1))^2)
20 ;
21 %derivatives of E2 dependent parameters in E2
22 if y(5)<=struct.E2th
23     SxE2_der=((struct.a1*struct.q1*(y(5))^(struct.q1-1)*(struct.lambda1^
24     struct.q1+y(5)^struct.q1))-struct.q1*struct.a1*y(5)^(struct.q1-1)*y(5)^
25     struct.q1)/(struct.lambda1^struct.q1+(y(5))^struct.q1)^2;
26 else
27     struct.a2 = struct.Sb +(struct.a1*(struct.E2th)^struct.q1)/(struct.
28     lambda1^struct.q1+(struct.E2th)^struct.q1);%for the continuity of the
29     function!

```



```

    SxE2_der=- (struct.a2*struct.lambda2^struct.q2*(struct.q2*(y(5)-struct.
    E2th)^(struct.q2-1)))/(struct.lambda2^struct.q2+(y(5)-struct.E2th)^
    struct.q2)^2;
25 end
27 %derivatives in q2
    if y(5)<=struct.E2th
29     SxE2_q2=0;
    else
31     SxE2_q2=(struct.a2*struct.lambda2^struct.q2*log(struct.lambda2)*(struct.
    lambda2^struct.q2+(y(5)-struct.E2th)^struct.q2)-struct.a2*struct.lambda2
    ^struct.q2*(struct.lambda2^struct.q2*log(struct.lambda2)+(y(5)-struct.
    E2th)^struct.q2*log(y(5)-struct.E2th)))/(struct.lambda2^struct.q2+(y(5)-
    struct.E2th)^struct.q2)^2;
    end
33
35 %Jacobian matrix
    J= [-struct.ks,0,0,0,SxE2_der
        k1s*struct.ks,-struct.delta1m,0,0,0
37         k2s_E1*struct.ks,0,-struct.delta2m,k2s_der*struct.ks*y(1),0
        0,struct.beta1,0,-struct.delta1p,0
39         0,0,struct.beta2,-delta2p_der*y(5),-delta2p_E1];
    % second addend of DM equation- specific for each parameter (df/dp)
41 dfdp=[SxE2_q2;0;0;0;0];
    %DM equations--> d/dt S(t)=J*S(t)+df/dp
43 dy(6)=J(1,:)*y(6:10)+dfdp(1);
    dy(7)=J(2,:)*y(6:10)+dfdp(2);
45 dy(8)=J(3,:)*y(6:10)+dfdp(3);
    dy(9)=J(4,:)*y(6:10)+dfdp(4);
47 dy(10)=J(5,:)*y(6:10)+dfdp(5);
49
    dy=dy';

```

Listing B.5: MATLAB ODE function for DM calculation for parameter q_2 .

B.3 iPSA

B.3.1 Numerical approach

Numerical approach applies the perturbation directly on the parameter using the gaussian approximation of an impulse. The perturbation is introduced inside the ODE solver through the ODE function that needs to be different for each parameter, this function generates the perturbed state trajectories which are then subtracted to the non-perturbed state trajectories. The level of perturbation was set as 1% of parameter nominal value. The following code presents the main code for the calculation of iPSA coefficients for parameter q_2 and for every state variable, at the end iPSA coefficients for E_2 are selected and a heatmap graphical representation is generated with normalized values between [-1;1] using the function `HeatMap(Data, ...'Colormap', ColormapValue, ...)`.

```

1  %perturbation amplitude 1% of nominal value
   struct.Aimpulse=0.01;
3  %% iPSA coefficients for q2
   for i=1:n %cycle for tau
5     disp(['i=', num2str(i)])
       struct.tau=tau(i);
7     [M Y_pert]=ode45('mod_iPSA_q2',[t(1):Tspan:t(end)],y0,options,struct);

9     struct.q2=2; %overwriting the parameter to the nominal value

11    iPSA.x_q2(:,i)=Y_pert(:,1)-Y(:,1)/(struct.Aimpulse*struct.q2);
       iPSA.mE1_q2(:,i)=Y_pert(:,2)-Y(:,2)/(struct.Aimpulse*struct.q2);
13    iPSA.mE2_q2(:,i)=Y_pert(:,3)-Y(:,3)/(struct.Aimpulse*struct.q2);
       iPSA.E1_q2(:,i)=Y_pert(:,4)-Y(:,4)/(struct.Aimpulse*struct.q2);
15    iPSA.E2_q2(:,i)=Y_pert(:,5)-Y(:,5)/(struct.Aimpulse*struct.q2);

17  end

19  for i=1:n %tau
       for j=1:n % t
21         if j>i
               iPSA.x_q2(i,j)=NaN;
23             iPSA.mE1_q2(i,j)=NaN;
               iPSA.mE2_q2(i,j)=NaN;
25             iPSA.E1_q2(i,j)=NaN;
               iPSA.E2_q2(i,j)=NaN;
27         end
       end
29  end

31  %% normalization of iPSA coefficients
   for i=1:n
33     iPSA.E2_q2(i,:)=iPSA.E2_q2(i,:)*(struct.q2/E2(i));
   end
35  HeatMap(iPSA.E2_q2,'ColorMap',colormap(hsv(512)))

```

Listing B.6: MATLAB code for iPSA numerical calculation.

The function "mod_iPSA_q2" used inside the ODE solver is necessary to calculate the perturbed state trajectories after a variation of the value of parameter q_2 . The perturbation is generated adding a gaussian impulse to the nominal value of the parameter using matlab function *gaussmf*($x, [\sigma c]$) that generates a gaussian impulse centered in τ and with $SD = 5$, thus having a total width of 0.02 days. After the model equation are implemented to produce the state variable trajectories.

```

36  function dy=mod_iPSA_q2(T,y,FLAG,struct)
38  % gaussian approximation of impulsive perturbation
       struct.q2=struct.q2+gaussmf(T,[5,struct.tau])*struct.Aimpulse*struct.q2;
40  %derivatives of E1 dependent parameters in E1
       k2s_E1=((struct.f1-1)*struct.k2s_min)/(1+exp((struct.lambda5-y(4))/struct.
           sigma1))+struct.k2s_min;
42  delta2p_E1=((struct.f2-1)*struct.delta2_min)*(1-1/(1+exp((struct.lambda6-y
           (4))/struct.sigma2)))+struct.delta2_min;
       %derivatives of E2 dependent parameters in E2
44  if y(5)<=struct.E2th

```

```

        SxE2=struct.Sb +(struct.a1*(y(5))^struct.q1)/(struct.lambda1^struct.q1+(
        y(5))^struct.q1);
46 else
        struct.a2 = struct.Sb +(struct.a1*(struct.E2th)^struct.q1)/(struct.
        lambda1^struct.q1+(struct.E2th)^struct.q1);%for the continuity of the
        function!
48 SxE2=((struct.a2*struct.lambda2^struct.q2)/(struct.lambda2^struct.q2+(y
        (5)-struct.E2th)^struct.q2));
        end
50
52 if T<=struct.th
        k1s=struct.k1s_min+(struct.a3*T^struct.q3)/(struct.lambda3^struct.q3
        +T^struct.q3);
54 else
        struct.a4 = struct.k1s_min+(struct.a3*struct.th^struct.q3)/(struct.
        lambda3^struct.q3+struct.th^struct.q3); %for the continuity of the
        function!
56 k1s=(struct.a4*struct.lambda4^struct.q4)/(struct.lambda4^struct.q4+(T-
        struct.th)^struct.q4);
        end
58 %model equations
        dy(1)=SxE2-struct.ks*y(1);
60 dy(2)=k1s*struct.ks*y(1)-struct.delta1m*y(2);
        dy(3)=k2s_E1*struct.ks*y(1)-struct.delta2m*y(3);
62 dy(4)=struct.beta1*y(2)-struct.delta1p*y(4);
        dy(5)=struct.beta2*y(3)-delta2p_E1*y(5);
64
        dy=dy';

```

Listing B.7: MATLAB code for "mod_ipsa_q2" of iPSA numerical calculation.

B.3.2 Analytical approach

In order to calculate iPSA coefficient with analytical approach is necessary to produce the $n \times n = 5 \times 5$ GFm matrix for each τ/t combinations, with $t \geq \tau$. For this reason the computation needs to cycle for each τ when perturbing the state trajectories and for each t when calculating the difference between the perturbed states and the non-perturbed ones. The level of perturbation is set as 1% of the maximum value of each state variable and the perturbation is added at each τ to the nominal value of the state variable at that specific time.

```

68 %parameters and time scale definition
        %calculation of state variables trajectories
        ...
70 n=length(t);
        %tau definition
72 tau=t;
        ...
74 %state variables perturbation (1% of the maximum of each curve)
        struct.delta_x=max(abs(x))*0.01;
76 struct.delta_mE1=max(abs(mE1))*0.01;
        struct.delta_mE2=max(abs(mE2))*0.01;

```

```

78 struct.delta_E1=max(abs(E1))*0.01;
   struct.delta_E2=max(abs(E2))*0.01;
80
   % calculation of the perturbed trajectories obtained
82 % by changing the initial conditions (y0_mod) at every cycle
   for i=1:n %cycle for tau
84     struct.tau=tau(i);
       y0_mod=[x(i)+struct.delta_x mE1(i) mE2(i) E1(i) E2(i)];
86     [M Y_pert_x]=ode45('mod_HPv',[t(i):Tspan:t(end)],y0_mod,options,struct);
       y0_mod=[x(i) mE1(i)+struct.delta_mE1 mE2(i) E1(i) E2(i)];
88     [M Y_pert_mE1]=ode45('mod_HPv',[t(i):Tspan:t(end)],y0_mod,options,struct
       );
       y0_mod=[x(i) mE1(i) mE2(i)+struct.delta_mE2 E1(i) E2(i)];
90     [M Y_pert_mE2]=ode45('mod_HPv',[t(i):Tspan:t(end)],y0_mod,options,struct
       );
       y0_mod=[x(i) mE1(i) mE2(i) E1(i)+struct.delta_E1 E2(i)];
92     [M Y_pert_E1]=ode45('mod_HPv',[t(i):Tspan:t(end)],y0_mod,options,struct)
       ;
       y0_mod=[x(i) mE1(i) mE2(i) E1(i) E2(i)+struct.delta_E2];
94     [M Y_pert_E2]=ode45('mod_HPv',[t(i):Tspan:t(end)],y0_mod,options,struct)
       ;
       % creation of the perturbed trajectories attaching the perturbed
96 %part of the curve to the non-perturbed part
       if i>1
98         Y_pert_x=[Y([1:i-1],:);Y_pert_x];
100        Y_pert_mE1=[Y([1:i-1],:);Y_pert_mE1];
102        Y_pert_mE2=[Y([1:i-1],:);Y_pert_mE2];
104        Y_pert_E1=[Y([1:i-1],:);Y_pert_E1];
106        Y_pert_E2=[Y([1:i-1],:);Y_pert_E2];
108
110        %creation of the structure data containing GFm (S_GFM)
112        data{i,j}.S_GFM(1,:)=(Y_pert_x(j,:)-Y(j,:))/struct.delta_x;
114        data{i,j}.S_GFM(2,:)=(Y_pert_mE1(j,:)-Y(j,:))/struct.delta_mE1;
116        data{i,j}.S_GFM(3,:)=(Y_pert_mE2(j,:)-Y(j,:))/struct.delta_mE;
118        data{i,j}.S_GFM(4,:)=(Y_pert_E1(j,:)-Y(j,:))/struct.delta_E1;
120        data{i,j}.S_GFM(5,:)=(Y_pert_E2(j,:)-Y(j,:))/struct.delta_E2;
122
124    end
126
128    for j=1:n %cycle for t
130        %saving the structure data for next implementation
132        save('data.mat','data')

```

Listing B.8: MATLAB code for GFm calculation.

Consequently iPSA coefficients $iS_{i,j}(t, \tau)$ are calculated applying the formula (2.15) and a heatmap graphical representation is generated with normalized values between [-1;1] using the function *HeatMap(Data, ...'Colormap', ColormapValue, ...)*. In the following the example for parameter q_2 and state variable E_2 is presented. $iS_{i,j}(t, \tau)$ elements were set as "Not a Number" (NaN) for $t < \tau$.

```

118 % iPSA coefficient of q2 for E2
120 % derivative in q2
122 for i=1:n
       if E2(i)<=struct.E2th
           SxE2_q2(i)=0;

```

```

124     else
125         SxE2_q2(i)=(struct.a2*struct.lambda2^struct.q2*log(struct.lambda2)*
126         struct.lambda2^struct.q2+(E2(i)-struct.E2th)^struct.q2)-struct.a2*struct
127         .lambda2^struct.q2*(struct.lambda2^struct.q2*log(struct.lambda2)+(E2(i)-
128         struct.E2th)^struct.q2*log(E2(i)-struct.E2th)))/(struct.lambda2^struct.
129         q2+(E2(i)-struct.E2th)^struct.q2)^2;
130     end
131 end
132 % iPSA coefficients calculation
133 for i=1:n
134     for j=1:n
135         if j>=i
136             data{i,j}.dfdp_q2=[SxE2_q2(i);0;0;0;0];
137             data{i,j}.iPSA_q2=data{i,j}.S_GFM_Statix*data{i,j}.dfdp_q2;
138             iPSA_q2_E2(j,i)=data{i,j}.iPSA_q2(5)*(struct.q2/E2(j));
139         else
140             iPSA_q2_E2(j,i)=NaN;
141         end
142     end
143 end
144 %creation of the normalized heatmap
145 HeatMap(iPSA.q2_E2/max(max(abs(iPSA.q2_E2))), 'ColorMap', colormap(hsv(512)))

```

Listing B.9: MATLAB code for iPSA coefficient calculation for q_2 parameter and E_2 state variable.

Bibliography

- [1] Varma A., Morbidelli M., Wu H., *Parametric Sensitivity in Chemical Systems*. Cambridge University Press, Cambridge, UK, 1999.
- [2] D. Hamby, “A review of techniques for parameter sensitivity analysis of environmental models,” *Environmental monitoring and assessment*, vol. 32, no. 2, pp. 135–154, 1994.
- [3] R. L. Iman and J. C. Helton, “An investigation of uncertainty and sensitivity analysis techniques for computer models,” *Risk analysis*, vol. 8, no. 1, pp. 71–90, 1988.
- [4] T. M. Perumal and R. Gunawan, “Understanding dynamics using sensitivity analysis: caveat and solution,” *BMC systems biology*, vol. 5, no. 1, p. 41, 2011.
- [5] A. Giaretta, B. Di Camillo, L. Barzon and G. M. Toffolo , “Modeling HPV early promoter regulation,” *37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society* , August 25-29 2015.
- [6] R. P. Dickinson and R. J. Gelinas, “Sensitivity analysis of ordinary differential equation systems—a direct method,” *Journal of computational physics*, vol. 21, no. 2, pp. 123–143, 1976.
- [7] T. Turányi, “Sensitivity analysis of complex kinetic systems. tools and applications,” *Journal of Mathematical Chemistry*, vol. 5, no. 3, pp. 203–248, 1990.
- [8] J. Hwang, E. Dougherty, S. Rabitz and H. Rabitz, “The Green’s Function Method of Sensitivity Analysis in chemical kinetics,” *The Journal of Chemical Physics* 69, 1978.

- [9] M. Bentele, I. Lavrik, M. Ulrich, S. Stößer, D.W. Heermann, H. Kalthoff, P.H. Krammer and R. Eils, “Mathematical modeling reveals threshold mechanism in CD95-induced apoptosis,” *The Journal of Cell Biology* 166(6), p. 839–851, Sep 13 2004.
- [10] Yue H., Brown M., Knowles J., Wang H., Broomhead, D. S. Kell D.B., “Insights into the behaviour of systems biology models from dynamic sensitivity and identifiability analysis: a case study of an NF-B signalling pathway.,” *Molecular BioSystems*, 2(12),, pp. 640–649., 2006.
- [11] Kreimer, R. Aimee et al., “Human papillomavirus types in head and neck squamous cell carcinomas worldwide: a systematic review,” *Cancer Epidemiology Biomarkers & Prevention* 14.2 , pp. 467–475, 2005.
- [12] A. Alba, M. Cararach and C. Rodríguez-Cerdeira, “The human papillomavirus (HPV) in human pathology: Description, pathogenesis, oncogenic role, epidemiology and detection techniques,” *The Open Dermatology Journal* 3 , pp. 90–102, 2009.
- [13] A. Alba, M. Cararach, and C. Rodríguez-Cerdeira, “The human papillomavirus (hpv) in human pathology: Description, pathogenesis, oncogenic role, epidemiology and detection techniques,” *The Open Dermatology Journal*, vol. 3, pp. 90–102, 2009.
- [14] H.-U. Bernard, “Gene expression of genital human papillomaviruses and considerations on potential antiviral approaches.,” *Antiviral therapy*, vol. 7, no. 4, pp. 219–237, 2002.
- [15] Centers for Disease Control and Prevention (CDC), “Human papillomavirus-associated cancers — United States, 2004–2008,” pp. 258–261, 2012.
- [16] C. Johansson and S. Schwartz, “Regulation of human papillomavirus gene expression by splicing and polyadenylation,” *Nature reviews Microbiology*, vol. 11, no. 4, pp. 239–251, 2013.
- [17] S. V. Graham, “Human papillomavirus: gene expression, regulation and prospects for novel diagnostic methods and antiviral therapies,” *Future microbiology*, vol. 5, no. 10, pp. 1493–1506, 2010.
- [18] M. A. Ozbun and C. Meyers, “Human papillomavirus type 31b e1 and e2 transcript expression correlates with vegetative viral genome amplification,” *Virology*, vol. 248, no. 2, pp. 218–230, 1998.

- [19] B. Friedman, *Principles and Techniques of Applied Mathematics*. New York: Wiley, 1956.

*Ringrazio la mia famiglia per il supporto incondizionato,
gli amici per portare la leggerezza ad ogni momento
e Luca, la mia colonna.*