University of Turin
Department of Computer Science

RESEARCH DOCTORATE IN COMPUTER SCIENCE
XXXIII Cycle

Modeling and analysis of biological complex systems

Simone Pernice

Tutor:
Prof. Marco Beccuti
Prof. Francesca Cordero

Supervisor of the Doctoral Program:
Prof. Marco Grangetto

Academic Year: **2020/2021**

Scientific Disciplinary Sector: **INF/01**

September 16, 2021

Acknowledgments

Vorrei ringraziare tutto il Qbio group e non solo (Giulio, Laura, Greta, Beatrice, Nicola, Prof. Sereno e Paolo) per avermi accompagnato durante il mio percorso di dottorato.
Un ringraziamento speciale va al Prof. Marco Beccuti, la Prof. Francesca Cordero, e al Prof. Gianfranco Balbo per i loro preziosissimi insegnamenti, consigli e commenti.

Grazie a tutta la mia famiglia che mi ha sempre supportato in ogni scelta e decisione, permettendomi di diventare ció che sono. In fine, l'ultimo ringraziamento lo dedico ad una persona a me molto cara, Andrea, che in questi anni mi ha sempre sostenuto, supportato e sopportato su ogni fronte.

Abstract

Computational modeling studies the behavior of complex systems by exploiting computer simulations and mathematical models. In the last decades computational modeling has become increasingly common in Life Science research, opening up new computation challenges to deal with the model complexity, the lack of knowledge and the missing parameters. In literature several modeling approaches were proposed according to the questions addressed during the analysis. All these approaches might be classified into two groups, the *statistical* and *mechanistic*, depending on the type of model exploited. In the *statistical* modeling approaches no mechanisms of the system under study is explicitly modeled, and they are thus mainly exploited to analyse time series data (such as growth data). Differently the *mechanistic* approaches are used to study the system dynamic considering different levels of abstraction, and modeling the interactions characterizing the system under study. These last approaches are able to better reproduce the dynamics characterizing the system behaviour enhancing the understanding of the mechanistic factors. Mechanistic approaches can be further divided into: *interaction-based*, *constraint-based* and *mechanism-based* modeling approaches, depending on the *size* and the *level of abstraction* of the model. The interaction-based approaches are commonly exploited to study topological properties of the system with methods based on the network analysis, therefore with few information, such as the stoichiometry of reactions, it is possible to study large and complex systems achieving qualitative knowledge only. The constraint-based approaches rely on the assumption of steady state and typically exploit Flux Balance Analysis (and its extensions) to study the system at equilibrium. Finally, the mechanism-based approaches describe the system behaviour and dynamics with high level of details by specifying all its characteristics, like the structure of the system, the mechanism of the events that produces changes in variables and the mathematical expressions used to represent them, with all their parameters defined.

In this thesis we focused principally on the *mechanistic* approaches, in particular to the mechanism-based ones, proposing new modeling and analysis techniques to increase the employment of these approaches to large complex networks. Indeed, the applicability of these approaches to these networks is mainly limited because of the lack of quantitative information on model parameters (due to cost and technical reasons) and the complexity of the systems. To deal with these aspects, we investigated

and developed different techniques exploiting i) *hybrid models* in which some component are modeled with different level of details and abstraction, ii) *model symmetries*, and iii) *approximation methods* for stochastic simulation.

These theoretical results were then implemented into a general modeling framework, called $\mathcal{G}reat\mathcal{M}od$. It represents a new way of facing the modeling analysis, exploiting the high-level graphical formalism, called Petri Net (PN), and its generalizations, which provide a compact, parametric and intuitive graphical description of the system and automatically derivation of the low-level mathematical processes (either deterministic and stochastic) characterizing the system dynamics. The framework novelties and strengths can be summarized into four points: (1) the use of a graphical formalism to simplify the model creation phase exploiting the *GreatSPN* GUI; (2) the implementation of an R package, `Epimod`, providing a friendly interface to access the analysis techniques (from the sensitivity analysis and calibration of the parameters to the model simulation); (3) a high level of portability and reproducibility granted by the containerization of all analysis techniques implemented in the framework; (4) a well-defined schema and related infrastructure to allow users to easily integrate their own analysis workflow in the framework.

Its effective was showed on different complex systems, such as the Multiple Sclerosis with and without specific therapies, the spread of the COVID19 in Piedmont region, and the Italian vaccination policy of the Pertussis disease.

Contents

Chapter 1

Introduction

In recent years the software and hardware improvements in computer science allowed to continuously enhance the simulation techniques based on computational models, making possible the deal with more complex systems considering different and new scales of analysis. In this contest, **modeling** means to describe the complex reality in a simple but meaningful way, requiring a balance between complexity and accuracy. A computational model uses a mathematical language first to reproduce experimental findings and next to help researchers to formulate new working hypotheses about the mechanisms that generated the observed phenomena, and to predict new phenomena. Indeed, the agreement between model results and observed data represents a necessary step of any modeling effort, but in order to prove its utility, a model should shed light on the system under study and offer a deeper understanding of its behaviour. In particular computational modeling has become increasingly common in Life Science research, thanks to a rapid development in sense of data availability, given by the next generation sequencing improvement and by major quantities of multiomic data for deep analyses and predictions, and either thanks to the availability of more powerful machines and techniques, opening up new computation challenges to deal principally with the model complexity, the lack of information and the missing parameters. Although the available modeling tools and frameworks are in a phase of rapid expan-

sion, modeling biological systems still remains a serious challenge.

Roughly speaking, modeling approaches can be classified into two groups: *statistical* and *mechanistic* ones [78]. The major differences between these two groups are: i) the aims of the analysis, ii) how the data (e.g., time series) are treated, and iii) the typology of models developed. Indeed, the former group is mainly exploited to analyse time series data (such as growth data) to obtain a set of descriptors of the dynamics by exploiting *no-parametric* functions, such as splines, to capture the statistical features of the data, ignoring any mechanisms of the system under study. Differently, the *mechanistic* group needs a great amount of information about the system mechanisms to be specified. Indeed, these modeling approaches describe explicitly the dynamics enhancing the understanding of the mechanistic factors and the causes of a generic phenomenon. From the literature it is possible to classify the mechanistic approaches into three groups: *interaction-based*, *constraint-based* and *mechanism-based* approaches. This classification depends principally on 1) the *size of the model* and 2) the *level of abstraction* [149, 28, 13]. The former aspect is characterized by the amount of mathematical details (e.g., number of parameters and variables) exploited to model the system. Instead, the latter aspect is suited to formally describe the functioning of the system, which should be identified taking into account the known biochemical, physical, or regulatory properties of system components, from qualitative (interaction-based models) to quantitative (mechanism-based models). We can briefly describe the main feature for these three main classes of models as follows:

- **Interaction-based models.** Interaction-based models are typically the results of network reconstruction processes that yield representations capturing structural information only. These models are commonly exploited to study topological properties of the network with methods based on the network analysis. In a given network, several topological features can be investigated: degree distribution (statistical indexes on the number of arcs connecting nodes), centrality measures (indexes that indicate the relative importance of nodes and arcs), presence of hubs (highly connected components), motifs (repeated architectures), and clusters (portions of the network with a high node density)[28, 2].

- **Constraint-based models.** The definition of constraint-based models

requires to introduce the teleonomy term. In contrast to the philosophical concept of teleology, teleonomy explains these goal-directed behaviours as an expression of genetic programs shaped by evolution. Flux Balance Analysis (FBA) was the first method proposed to perform this task [67]. The goal of Flux Balance Analysis (FBA) is to compute the distribution of fluxes in a system at the equilibrium. The equilibrium, or steady state, is a situation that follows a perturbation with no more net production or consumption over time. The transient period occurring after a perturbation of the system and before its steady state behavior is instead neglected. In this context, the choice among a set of feasible flux distributions is performed optimizing for a specific system goal according to the teleonomic assumption.

- **Mechanism-based models.** Mechanism-based models have the potential to reproduce the dynamics of the system. After listing all the entities and events characterizing the system, the system dynamics can be obtained by defining: a scheme of the mechanism of interactions among components, a mathematical representation of these interactions, some specific values for the kinetic parameters appearing in this representation, an initial set of values for the entities concentrations. All these model elements have to be defined with caution, starting from a review of the information available in the literature.

Furthermore, a schematic overview of these three groups is depicted in Fig. 1.1, where approaches vary with respect to: (i) the size of the model, defined in terms of the number of components and respective interactions, (ii) the mathematical complexity (computational costs) required for the analysis of the model, and (iii) the amount of detail needed to each approach, from simple adjacency matrices to all the kinetic parameters and the stoichiometry characterizing the system.

In this thesis we focused principally on the *mechanistic* approaches, considering in particular the mechanism-based ones, which describe the system behaviour and dynamics with high level of detail by specifying all its characteristics, therefore these approaches are the most likely candidates to provide a complete understanding of the system under study. Indeed to devise high precision models, a great amount of details and information regarding the analysed system must be known. However, more details do not always imply

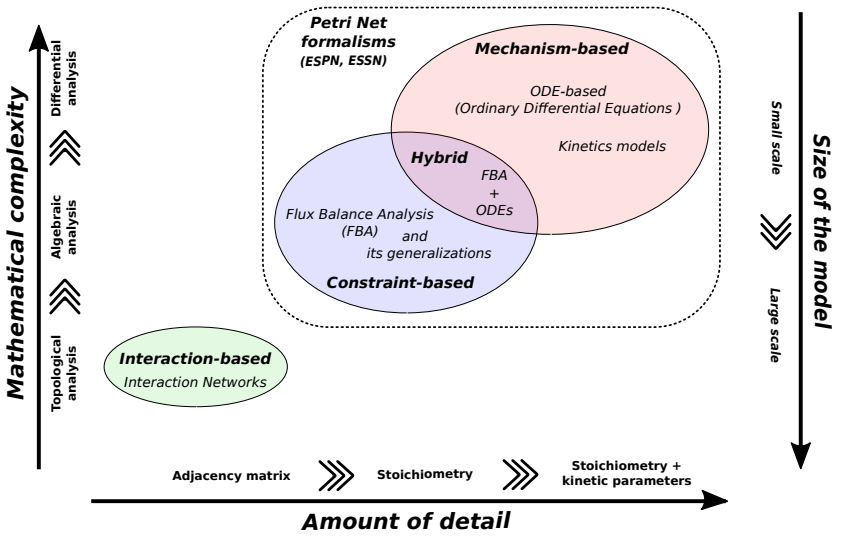


Figure 1.1: The three classes of modeling approaches are positioned basing on their level of detail and the computational cost. The box represents the high level Petri Net formalism and its extensions which we exploited to represent the mechanism, constraint and the hybrid based models.

more precision, and we must be aware that the computational cost increases together with the number of "free" parameters to be estimated. To cope with the increasing quantity of details and complexity, we investigated and developed different modeling techniques, exploiting the high level graphical formalism, namely Petri Net (PN) [84], to model the system in a more compact and parametric manner. PNs are a well-known graphical mathematical formalism for the description of the flow of activities in complex systems. With respect to other similar formalisms for system representation (e.g. block diagrams, logical trees, queuing networks, etc.), PNs are particularly suited to represent in a natural way interactions among system components (i.e. synchronization, sequentiality, concurrency and conflict). Indeed, the PNs are able to be both powerful and capable of accommodating a team-work with scientists without computer science and mathematical skills to translate ideas and hypotheses into equations and coding, given that the model definition and analysis represent a mathematical, computational, and resource intensive challenge.
In the literature, PNs have been extensively used to study a wide range of applications ranging from chemical processes to man-made systems (communication networks, computational distributed systems, manufacturing systems, etc.). In particular, the first application of PNs to modeling biological path-

ways was published by [126]; afterward, many other research works highlighted the advantages of using PNs to model biological systems [71, 34, 68, 152].

Starting from the PN formalism during my PhD we defined two new graphical formalisms, namely Extended Stochastic Petri Net (Extended Stochastic Petri Net (ESPN)) [115, 114] and the Extended Stochastic Symmetric Net (Extended Stochastic Symmetric Net (ESSN)) [116], to make easier the model creation phase in a compact and parametric manner. Thanks to the definition of these two new formalisms we provide a general purpose tool able to simulate the differently kinetic laws governing the biological reactions and events. Exploiting these formalisms, it is possible to automatically generate the stochastic (i.e., the Continuous Time Markov Chain (CTMC)) and deterministic (i.e., Ordinary Differential Equation (ODE)s) processes underlying the model. The choice between a deterministic model and a stochastic one depends on the application under study and the questions leading the analysis. Indeed, the stochastic process is able to reproduce relevant random features like variance, bimodality, and tail behavior that cannot be captured by a deterministic quantity, but with a higher computational cost than the solution of the ODEs system characterizing the deterministic process. For instance, considering large systems (i.e., systems with a large numbers of interacting elements) Stochastic Simulation Algorithm (SSA) [57], an exact stochastic method widely used to simulate chemical systems whose behaviour can be described by the Master equations, could be computationally too slow, and thus approximation methods must be used. For this reason we adapted the *τ-leaping algorithm* [58] to be applied to the general model that can be generated using PN formalism. This method speeds up the stochastic simulation of system by approximating the number of system events during a chosen time increment (i.e., τ) as a Poisson random variable, providing a good compromise between the solution execution time and its quality.

Another important issue characterizing both the deterministic and stochastic processes is the model complexity, for this reason we studied new solutions approaches to scale up the model analysis. Hence, to reduce the model complexity we proposed 1) to exploit the model symmetries, 2) to define hybrid systems combining different levels of details to model the different parts of the system. In the former case we proposed a new method to exploit the symmetries of an ESSN model to reduce the complexity of the system analysis.

Thus, in according to this method, we developed an algorithm to automatically derive from the model a reduced ODEs system exploiting the systems symmetries, from which the same measures derived by the original ODEs can be computed [12, 11]. Differently, in the latter we exploited the ESPN formalism to define hybrid systems in which parts of the net are modeled using different level of details. In particular we proposed to combine **mechanism and constraint models**, i.e., some components are modeled with a high level of precision exploiting mechanism-based model (e.g., ODEs system), and other by a low level of precision given by the constraint-based models (e.g. FBA), as shown in Fig. 1.1. In particular, we showed the applicability of this new approach considering the **metabolic pathways altered in Pancreatic Ductal Adenocarcinoma** cells, [115].

These theoretical results introduced above were then implemented into a new framework to study complex systems: ***GreatMod***. It combines the high-level graphical formalism of the PNs and its generalizations (e.g., ESPN and ESSN) with classical and new model analysis techniques. We can summarize its novelties and strengths into four points: (1) the use of a graphical formalism to simplify the model creation phase exploiting the *GreatSPN* GUI [5]; (2) the implementation of an R package, called `Epimod` providing a friendly interface to access the analysis techniques implemented in the framework; (3) a high level of portability and reproducibility granted by the containerization (into Docker images) of all analysis techniques implemented in the framework; (4) a well-defined schema and related infrastructure to allow users to easily integrate their own analysis workflow in the framework. The effectiveness of this framework is shown through three applications: **Multiple Sclerosis** (Multiple Sclerosis (MS)) [120, 116, 114, 117], **Italian Pertussis vaccination policy** [27], and **COVID-19 Epidemic in Piedmont** [119].

Finally, this thesis is organized into three parts:

1. **Background.** In this first part the notations used in the rest of the thesis are introduced. In detail, Chapter 2 introduces the Petri Net formalism and its generalizations (i.e., Stochastic Petri Net, Stochastic Symmetric Net, Symbolic Net), describing how the stochastic and deterministic processes can be generated by a model described trough these formalisms. Chapter 3 gives the general details behind the sim-

ulations of stochastic processes, and their deterministic approximation given by Kurtz's Theorem. Finally, in Chapter 4 we focused on a specific constraint-based approach, i.e., the Flux Balance Analysis, which will be exploited to define the hybrid modeling approach.

2. **Theoretical results.** In the second part, Chapter 5 focuses on the introduction of two new formalisms, the Extended Stochastic Petri Net (ESPN) and the Extended Stochastic Symmetric Net (ESSN), obtained by extending the Stochastic Petri Net and Stochastic Symmetric Net respectively. In addition, in Chapter 6 three new developed solutions techniques were introduced to scale up the analysis of ESPN and ESSN models. Firstly, we described an algorithm by which the complete net unfolding is avoided when the reduced ODEs system is automatically derived from the (E)SSN model. Then, we defined the hybrid modeling approach which combine the constraint-based approach, given by the Flux Balance Analysis, and the mechanistic-based approach, given by the system of ODEs. Finally, we extended the $\tau-$leaping algorithm, widely used to approximate the stochastic simulation of the Master equations, to deal with general model.

3. **Tools and applications.** In the last part we introduced the the $\mathcal{G}reat\mathcal{M}od$ framework, implementing the above theoretical results. In Chapter 7 all $\mathcal{G}reat\mathcal{M}od$ novelties and functionalities are explained in details. In particular, in Section 7.1 we introduce the R package, namely `Epimod`, in which the functions characterizing the modeling analysis steps are implemented. Chapter 8 focuses on the three case studies: Italian Pertussis vaccination policy (Section 8.1), Multiple Sclerosis (Section 8.2), and COVID-19 Epidemic in Piedmont (Section 8.3).

Part I

Background

Chapter 2

Petri Net formalism and its generalizations

Petri Nets (PNs) [99] and their extensions are widely recognized to be a powerful tool for modeling and studying complex systems thanks to their ability of representing systems in a natural graphical manner and of allowing the computation of qualitative and quantitative information about the behavior of these systems. They have been satisfactorily applied to fields such as communication networks, computer systems, manufacturing systems, etc. The first application of PNs to modeling biological pathways was published by Reddy et al in [126]; afterward, many other research works highlighted the advantages of using PNs to model biological systems [71, 34, 68, 152].

In this chapter we recall the formal definitions of the PN formalism and its extensions, namely Stochastic Petri Net (SPN) and Stochastic Symmetric Net (Stochastic Symmetric Net (SSN)). In detail, we show how to derive from the net both the CTMC and the ODEs underlying the model. Finally, the definition of symmetries of an SSN model is formally introduced.

2.1 Petri Net formalism

PNs are bipartite directed graphs with two types of nodes, namely *places* and *transitions*. The former ones correspond to state variables of the system and are graphically represented as circles. The latter ones correspond to the events that can generate a state change and are graphically represented as boxes. Nodes of different types are connected by *arcs*, which express the relation between states and event occurrences. A specific cardinality (multiplicity) is associated with each arc, and it describes the number of tokens removed from (or added to) the corresponding place upon the firing of the transition the arc is connected to. Graphically, it is written beside the arc, but the default value of one is omitted. Finally, places can contains *tokens* drawn as black dots. Then, the number of tokens in each place defines the state of a PN, called *marking*.
Let introduce the formal definition of a PN.

Definition 1 (Petri Nets)**.** *A PN system is a tuple* $\mathcal{N} = (P,\ T,\ I,\ O,\ \mathbf{m}_0)$ *where:*

- $P = \{p_i\}$ *is a finite and non empty set of* places*, with* $i = 0, \ldots, n_p$*, where* n_p *is the number of places.*
- $T = \{t_j\}$ *is a finite and non empty set of* transitions *with* $P \cap T = \emptyset$ *and* $j = 0, \ldots, n_t$*, where* n_t *is the number of transitions.*
- $I, O : P \times T \to \mathbb{N}$ *are the* input*,* output*, that define the arcs of the net and that specify their multiplicities.*
- $\mathbf{m}_0 : P \to \mathbb{N}$ *is a multiset on* P *representing the* initial marking *of the net.*

Functions I and O describe the input and output arcs of transitions, respectively. For convenience, these can represented by $n_t \times n_p$ matrices of natural numbers. The matrix $L = O - I$ is called the *incidence matrix*, and it represents the overall effect of the transitions.
Let us introduce the following shorthand notation: given a transition $t \in T$, we denote with ${}^\bullet\mathbf{t} = \{p \in P : I(p,t) > 0\}$, and $\mathbf{t}^\bullet = \{p \in P : O(p,t) > 0\}$ the subset of P containing all the places in input and output to transition t, respectively.

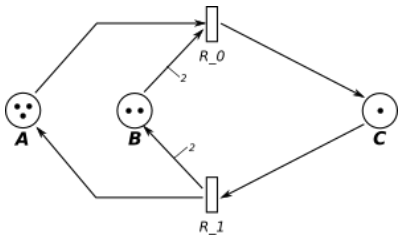


Figure 2.1: Representation of the PN describing the chemical reactions $A + 2B \xrightarrow{R_0} C$ and $C \xrightarrow{R_1} A + 2B$. Places represent the enzymes or compounds while transitions represent the events/reactions among the compounds. Tokens within the places stand for the molecules taking part in the reaction.

Finally, the evolution of the system is given by the firing of enabled transitions, where a transition is enabled if each input place contains a number of tokens greater or equal than a given threshold defined by the multiplicity of the corresponding input arc. Formally, a transition t is *enabled* in marking $\mathbf{m}$ iff $\mathbf{m}(p) \geq I(p,t), \forall p \in P$ where $\mathbf{m}(p)$ represents the number of tokens in place p in marking $\mathbf{m}$. Enabled transitions may *fire*, so that the firing of transition t in marking $\mathbf{m}$ yields a new marking $\mathbf{m'} = \mathbf{m} - I(t)^T + O(t)^T = \mathbf{m} + L(t)^T$. So, a transition occurrence/firing removes a fixed number of tokens from its input places ($^\bullet\mathbf{t}$) and adds a fixed number of tokens into its output places ($\mathbf{t}^\bullet$) (according to the multiplicity of its input/output arcs). Finally, the set of all the markings reachable through the transition firings from an initial marking is called the Reachability Set (RS). The behaviour of the net is described by means of the Reachability Graph (RG), a directed graph whose nodes are the markings of the RS and whose arcs are tagged with the names of the transitions that induce the corresponding marking changes.

As an example, in Fig. 2.1 the PN model of the chemical reactions $A + 2B \xrightarrow{R_0} C$ and $C \xrightarrow{R_1} A + 2B$ is showed, where $P = \{A,\ B,\ C\}$ are places while $T = \{R_0,\ R_1\}$ are transitions. Observe that the arcs connecting B to R_0 and R_1 to B have multiplicity 2 to capture the stoichiometry of these reactions. Places can contain tokens (e.g., molecules of the corresponding entities) drawn as black dots. The marking is *A(3)+B(2)+C(1)* and corresponds to the system state in which there are three molecules of A, two of B and one of C. Finally the input and output matrices are defined as:

$$I = \begin{bmatrix} 1 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ and } O = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 2 & 0 \end{bmatrix}.$$

Thus, the transition R_0 is enabled only if at least one token is in A and two tokens are in B. The firing of R_0 removes one token from A and two tokens from B, while it adds one token in C.

Let us note that the PN formalism can be exploited to model different biological systems, from reaction systems (interactions among compounds, such as the example showed in Fig. 2.1), cellular systems (interactions among cells) to epidemiological systems (interactions among individuals). For instance, a further example is reported in Fig. 2.4, in which is represented the classical Susceptible-Infected-Recovered (SIR) model (susceptible, infected, and recovered individuals).

2.2 Stochastic Petri Net formalism

The introduction of time into the PN formalism allows to model the temporal dynamics of the biological system under investigation. Several time dependency extensions have been proposed in the literature; among them SPNs [75] assume that exponentially distributed random delays (interpreted as duration of certain activities) are associated with transition firings. In this way the temporal behaviour of the system can be described as a stochastic process (i.e. Continuous Time Markov Chain - CTMC) governed by the so-called Chapman-Kolmogorov (CK) equations [52], whose differential form corresponds to the Chemical Master Equation (CME) [150] typically used to describe the behaviour of biological system. It follows that the CMEs can be automatically derived from an SPN model by observing that the underlying CTMC can be represented as a graph isomorphic to the RG of the net. However, in case of very complex models, the generation and solution of the underlying CTMC could be unfeasible, due to the well-known state space explosion problem, thus Monte Carlo simulation can be exploited to study the system behaviour. Let us underline that each trajectory obtained by Monte Carlo simulation represents one sample of the probability mass function that solves the CME. Among the simulation algorithm we mainly exploited the Gillespie algorithm, called *Stochastic Simulation Algorithm* (SSA) [57], and the τ-leaping method [58]. The SSA is an exact stochastic method widely used to simulate chemical systems whose behaviour can be described by the CME, Eq.s 2.2. In case of very large systems (i.e., systems with a large num-

bers of interacting elements) SSA could be computationally too slow, and then approximation methods must be used. Among these approaches the *τ-leaping algorithm* provides a good compromise between the solution execution time and its quality. Indeed, this method speeds up the stochastic simulation of system by approximating the number of system events during a chosen time increment (i.e., τ) as a Poisson random variable.
A second method to approximate the CTMC, when the system stochasticity is negligible and under specific assumptions [10], is defined by the so-called deterministic approach [83] which approximates the system behaviours through a deterministic process based on ODEs which can be written for each place of the net. This allows to derive and solve an ODEs system that is much smaller than that obtained through CME in which one CK differential equation is needed for each state of the system. More details regarding these simulation techniques are reported in Chapter 3.

Hereafter we recall the formal definition of SPN and how CME and ODE systems can be written in terms of SPN elements.

Definition 2 (Stochastic Petri Nets)**.** *A SPN system is a tuple* $(P, T, I, O, \mathbf{m}_0, \lambda)$*:*

- $P,\ T,\ I,\ O,\ \mathbf{m}_0$ *are introduced in the Definition 1;*
- $\lambda : T \to \mathbb{R}$ *gives the firing rate of each transition.*

In SPNs each transition is associated with a specific intensity, representing the parameter of the exponential distribution that characterises its firing time. By supposing that all the transitions intensity are defined in according to Mass Action (MA) law [50], the parameter associated with an enabled transition t is given by the function

$$\varphi(m, t, \nu) = \frac{\lambda(t)}{I[p,t]!} \prod_{\langle p| \ p \in {}^\bullet \mathbf{t} \rangle} m_{p_j}(\nu)^{I[p,t]} \tag{2.1}$$

with $\lambda(t)$ representing the rate of the enabled transition t.

In the SPNs the stochastic firing delays, sampled from negative exponential distributions, allow to automatically derive the underlying CTMC that can be studied to quantitatively evaluate the system behaviour [99]. In details, the CTMC state space, $\mathbb{S}$, corresponds to the RS of the corresponding SPN,

i.e. all the possible markings that can be reached from the initial marking. Thus, the CMEs for the CTMC are defined as follows:

$$\frac{d\pi(m_i,\nu)}{d\nu} = \sum_{m_k} \pi(m_k,\nu) q_{m_k,m_i} \qquad m_i, m_k \in \mathbb{S} \tag{2.2}$$

where $\pi(m_i,\nu)$ represents the probability to be in marking m_i at time ν, and q_{m_k,m_i} the velocity to reach the marking m_i from m_k, defined as

$$q_{m_k,m_i} = \sum_{\substack{t\in T \wedge \\ t\in E(m_k)_{|m_i}}} \varphi(m_k,t,\nu) L[p,t]. \tag{2.3}$$

where $L[p,t]$ is the overall effect of the transition t on place p (obtained from the incidence matrix), and $E(m_k)_{|m_i}$ is the set of all transitions enabled in marking m_k whose firing brings to the marking m_k.

The ODEs that describe the (approximate) deterministic behaviour of the SPN have the following form:

$$dx_i(\nu) = \sum_{j=1}^{n_T} \varphi(x,t_j,\nu) L[p_i,t_j] d\nu \qquad \forall i \in \{1,\dots,n_P\}. \tag{2.4}$$

where $x_i(\nu) \in \mathbb{R}^+$ is the continuous approximation of the number of tokens at time $\nu \in \mathbb{R}^+$ in place p_i, and $\varphi(x,t_j,\nu)$ is the rate of the reaction t_j which may depend on the marking of the SPN. In particular, assuming that all the system reactions follow the MA law, the Eq. 2.1 becomes

$$\varphi(x,t,\nu) = \lambda(t) \prod_{p_k \in {}^\bullet \mathbf{t_j}} x_k^{I[p_k,t]}, \qquad \forall t \in T. \tag{2.5}$$

For example, referring to the system represented in Fig. 2.1, its dynamics are described by the following system of ODEs

$$\begin{cases} \dfrac{dx_A(\nu)}{d\nu} = \overbrace{-\lambda(R_0) x_A(\nu) x_B(\nu)^2}^{\text{reaction } R_0} \overbrace{+\lambda(R_1) x_C(\nu)}^{\text{reaction } R_1} \\ \dfrac{dx_B(\nu)}{d\nu} = \overbrace{2\lambda(R_0) x_A(\nu) x_B(\nu)^2}^{\text{reaction } R_0} \overbrace{+2\lambda(R_1) x_C(\nu)}^{\text{reaction } R_1} \\ \dfrac{dx_C(\nu)}{d\nu} = \overbrace{\lambda(R_0) x_A(\nu) x_B(\nu)^2}^{\text{reaction } R_0} \overbrace{-\lambda(R_1) x_C(\nu)}^{\text{reaction } R_1} \end{cases}$$

where dx_A, dx_B, dx_C represent the changes of the number of molecules of type A, B, C respectively.

2.3 Stochastic Symmetric Net formalism

Among the PN generalisations proposed in literature, Stochastic Symmetric Nets (SSNs) [30] extend PNs providing a more compact and readable representation of the system, thanks to the possibility of distinguish different type of tokens and thus graphically represented in the models as dots of different colors.
In SSNs each place $p \in P$ has an associated color domain (a data type) denoted $cd(p)$ which defines the type of tokens stored in such place. Color domains are defined by the Cartesian product of elementary types called *color classes* $\mathcal{C} = \{C_1, \ldots, C_n\}$, so that $cd(p) = C_1^{e_1} \times C_2^{e_2} \times \ldots \times C_n^{e_n}$ where e_i is the number of times C_i appears in $cd(p)$. Color classes are finite and disjoint sets. They can be ordered (in this case a successor function $(++)$ is defined on the class, inducing a circular order among the elements in the class), and can be partitioned into (static) subclasses (e.g $C_{i,j}$ is the ith static subclass of the jth color class).
The example model represented in Fig. 2.2 extends the PN model introduced previously in Fig. 2.1 to take in account different positions where the A, B, and C molecules might be arranged. For simplicity in this example we omitted the movements, and we considered only three positions represented by one color class, namely *Position*, which is divided into three static subclasses (i.e. $Pos1$, $Pos2$ and $Pos3$). In this case the color domain of all the three places is defined by the only color class, representing that one molecule might be in position 1,2 or 3 (token colored in red, blue and green respectively).

An instance of a given transition t is an assignment of the transition variables to a specific color of proper type defined by its color domain $cd(t)$. Hence, we can define the function var which assigns to each transition $t \in T$ a set of variables, each taking values in a given color class C_i of $\mathcal{C}$ (the variable's type). For example, t with $var(t) = \{x, y, z\}$, $x, y : C$ and $z : C'$, is a transition with three parameters, x, y, z, taking values on sets C (x, y) and C' (z), respectively, hence $cd(t) = C \times C \times C'$. If there are n variables of the same type C, an indexed family of variable symbols $\{v_i\}$, $i : 1, .., n$, may be used: thus in the example above, we might use $\{x_1, x_2, z\}$, with $x_i : C$, $z : C'$. In the example, x=c_1, y=c_2, z=c', with $c_i \in C$, $c' \in C'$, is an instance of t, also denoted as a tuple $\langle c_1, c_2, c' \rangle \in cd(t)$ (this tuple is the transition

instance's color)[1]. An explicit notation may be used, i.e., $\langle t, c\rangle$ to denote an instance, where c is the assignment, also called binding. The transition variables appear in the functions labeling its arcs. For instance, the color domain of transition R_0 is $Position \times Position$, the variables characterizing its input arc are $x,\ y \in Position$, the binding is defined as $c = \langle x, y\rangle$ and consequentially the instance is denoted as $\langle t, x, y\rangle$. Moreover, a guard can be used to define restrictions on the allowed instances of a transition. A guard is a logical expression defined on the color domain of the transition, and its terms, called basic predicates, allow (i) to compare colors assigned to variables of the same type $(x = y,\ x \neq y)$; (ii) to test whether a color element belongs to a given static subclass $(x \in C_{i,j})$; (iii) to compare the static sub-classes of the colors assigned to two variables $(d(x) = d(y), d(x) \neq d(y))$. An example is given by the guard associated with the transition R_0, which denies the A molecules in the third position to participate at the reaction R_0.
The *marking* of an SSN is defined by the number of colored tokens in each place. For instance, a possible marking of the place A in Fig. 2.2 is $2\langle pos1\rangle + 1\langle pos2\rangle$, indicating that in place A there are two (red) tokens in the color class *Pos1*, and one (blue) in *Pos2*.

Each arc connecting a place p to a transition t, namely an input arc of t, is labeled with an expression defined by the function $I[p, t] : cd(t) \rightarrow Bag[cd(p)]$, with domain $cd(t)$ and codomain $Bag[cd(p)]$, where $Bag[A]$ is the set of multisets built on set A, and if $b \in Bag[A] \wedge a \in A$, $b[a]$ denotes the multiplicity of a in the multiset b. Similarly, each arc connecting a transition t to a place p, namely an output arc of t, is denoted by the function

$$O[p, t] : cd(t) \rightarrow Bag[cd(p)].$$

Thus, the evaluation of $I[p, t]$ (resp. $O[p, t]$), given a legal binding of t, provides the multiset of colored tokens that will be withdrawn from (input arc) or will be added to (output arc) the place connected to that arc by the firing of such transition instance. Moreover, we denote with $^\bullet\mathbf{t}$ the set of input places of the transition t and with $\mathbf{t}^\bullet$ the set of output places of t, i.e. $^\bullet\mathbf{t} := \{p \in P|\ \exists\ c \in cd(p)\ s.t.\ I[p, \mathbf{t}](c')[c] > 0, \forall c' \in cd(t)\}$ and $\mathbf{t}^\bullet := \{p \in P|\ \exists\ c \in cd(p)\ s.t.\ O[p, \mathbf{t}](c')[c] > 0, \forall c' \in cd(t)\}$. In details, a transition instance $\langle t, c\rangle$ is enabled and can fire in an marking m, iff: (1) its guard evaluated on c is true; (2) for each place p we have that $I[p, t](c) \leq m(p)$, where $\leq$ is the

[1] If $var(t) = \{\}$ then by convention $cd(t) = E = \{e\}$, E being the neutral color class

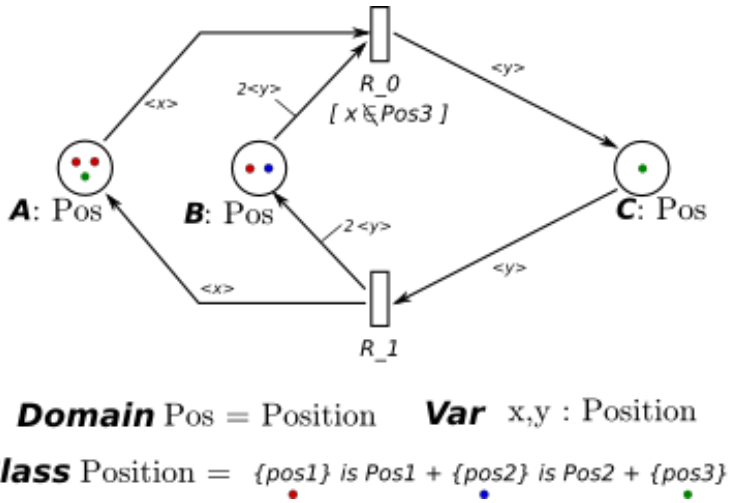


Figure 2.2: Representation of the SSN of the chemical reactions introduced in Section 2.1, considering different positions. The color domain defined by the color class *Position* is associated with each place. Let us observe that *Position* is divided into three static subclasses, which are represented, for clarity, by the red, blue and green colors.

comparison operator among multisets. We use the notation $E(t, m)$ to denote the set of all instances of t enabled in marking m. The firing of the enabled transition instance $\langle t, c\rangle$ in m produces a new marking m' such that, for each place p, we have $m'(p) = m(p) + O[p, t](c) - I[p, t](c)$.
In SSNs, the firing time of an enabled transition instance $\langle t, c\rangle$ is sampled from a negative exponential distribution whose rate is given by the function ω, i.e.

$$\omega(t, c) = \begin{cases} r_i & cond_i(c)\ i = 1, \ldots, n, \\ r_{n+1} & otherwise, \end{cases}$$

where $cond_i$ are boolean and mutually exclusive expressions comprising standard predicates on the transition color instance. In this manner, the firing rate r_i of a transition instance can depend only on the static sub-classes of the objects assigned to the transition parameters and on the comparison of variables of the same type. Thus, these stochastic firing delays, sampled from a negative exponential distribution, allow to automatically derived the stochastic process, i.e. the CTMC, that describes the dynamics of the SSN model. Specifically, the CTMC states are isomorphic to SSN markings and the state changes correspond to the marking changes in the model.
Hereafter we recall the formal definition of SSN.

Definition 3 (Stochastic Symmetric Net)**.** *An SSN is a nine-tuple:*

$$\mathcal{N}_{SSN} = \langle P, T, \mathcal{C}, I, O, cd, \Theta, \omega, \mathbf{m}_0 \rangle$$

where

- P, T *are introduced in the Definition 1.*
- $\mathcal{C} = \{C_1, \ldots, C_n\}$ *is the finite set of basic color classes.*
- $I, O[p,t] : cd(t) \to Bag[cd(p)]$ *are the input and output matrices (similarly as in Definition 1), whose elements are in the form of the arc functions defined above.*
- $cd : \bigotimes_{i=1}^{n} \bigotimes_{j}^{e_i} C_i^j$ *is a function defining the color domain of each place and transition (where* $e_i \in \mathbb{N}$ *is the number of occurrences of the class* C_i*); for places it is expressed as Cartesian product of basic color classes, for transitions it is expressed as a list of variables with their types. Observe that a place may contain undistinguished tokens only or a transition may have no parameters, in this case their domain is* neutral*.*
- Θ *is the vector of guards and maps each element of* T *into a standard predicate (*$\Theta(t)$ *may be the constant* true*, which is also a standard predicate).*
- $\omega : T \times cd(t) \to \mathbb{R}$ *is the function returning the rate of transition* t *assuming the firing of the instance* $\langle t, \mathbf{c} \rangle$*.*
- $\mathbf{m}_0 : P \to Bag[cd(p)]$ *is the initial marking, mapping each place* p *on a multiset on* $cd(p)$*.*

Assuming that all the transitions of the SSN are characterized by a MA law, the intensity defined in Eq. 2.1 of $\langle t, c \rangle$ in marking m becomes:

$$\varphi(m,t,c) = \omega(t,c) \prod_{\langle p,c' \rangle |\ p \in {}^\bullet\mathbf{t}\ \wedge\ c' \in cd(p)} m[p][c']^{I[p,t](c)[c']} \tag{2.6}$$

where $m[p][c']$ denotes the marking of place p for color c'.

Let us note that the translation of an SSN model into an equivalent SPN is always possible by means of a procedure called **unfolding**, which consists of

replicating places and transitions as many times as the cardinalities of the corresponding color domains. Therefore, colors disappear in the unfolded model and the complex behavior due to color combinations, color arc functions, and color transition guards, is encoded with a net structure in which tokens are indistinguishable entities and new transitions, places, and arcs are introduced to account for the different actions performed by instances of the same transition on different colored tokens. In particular, for the places containing tokens of different classes, the cross-product of the color domains are computed and each resulting element associated with the original name of the place becomes the name of a corresponding new place in the unfolded net. These place replicas are thus named $p_{c_1,\ldots,c_n}$, where $\forall\ \langle c_1,\ldots,c_n\rangle \in cd(p)$. Similarly, for what concerns the transitions, the new names in the unfolded net start with the name of the transition in the SSN model, followed by a list expressing the binding. The possible bindings are derived accounting for the restrictions possibly imposed by the associated transition guards (when present). These transition replicas are thus named $t_{c_1,\ldots,c_n}$ where $\forall\ \langle c_1,\ldots,c_n\rangle \in cd(t)$. For instance, in our example model Fig. 2.2, if we consider the A place, which has color domain Pos, then the number of places obtained from the unfolding is $(n_{Pos1}+n_{Pos2}+n_{Pos3})$, where $n_{Pos1},\ n_{Pos2},\ n_{Pos3}$ are the cardinalities of the sub classes $Pos1,\ Pos2,\ Pos3$ respectively. Since, $n_{Pos1}=n_{Pos2}=n_{Pos3}=1$ then the number of unfolded places of A is three: $A_{pos1},\ A_{pos2},\ A_{pos3}$, representing the number of tokens of color $pos1,\ pos2,\ pos3$ respectively, as depicted in Fig. 2.3, in which we showed the unfolding of the sub SSN of the model introduced in Fig. 2.2 considering only the *R_0* transition. Similarly, the unfolded transitions derived from R_0 are defined by all the possible combinations of the color variables (which are associated with the transition input arcs) $x,\ y$ varying among the sub color classes defining the $cd(R\_0)$. For instance, the transition $R\_0\_x\_pos1\_y\_pos2$ is obtained because of $x \in Pos1$ and $y \in Pos2$. Let us note that because of the guard associated with R_0 the place A_pos3 is not connected with any transitions, and transitions with $x \in Pos3$ (e.g., $R\_0\_x\_pos3\_y\_pos2$) can not be derived.
In general, let $\widetilde{C_i} = \{C_{i,h}, h : 1 \ldots |\widetilde{C_i}|\}$ be the set of static subclasses of color class C_i. The number of places resulting from the unfolding of a place p with color domain $cd(p) = C_{i_1}^{n_1} \times \ldots \times C_{i_k}^{n_k}$ is given by the product $\prod_{j=1}^{k} (\sum_{h=1}^{|\widetilde{C_{i_j}}|} |C_{i_j,h}|)^{n_j}$. Where, the notation $C_{i_j}^{n_j}$ means that class C_{i_j} occurs n_j times in the Cartesian product.

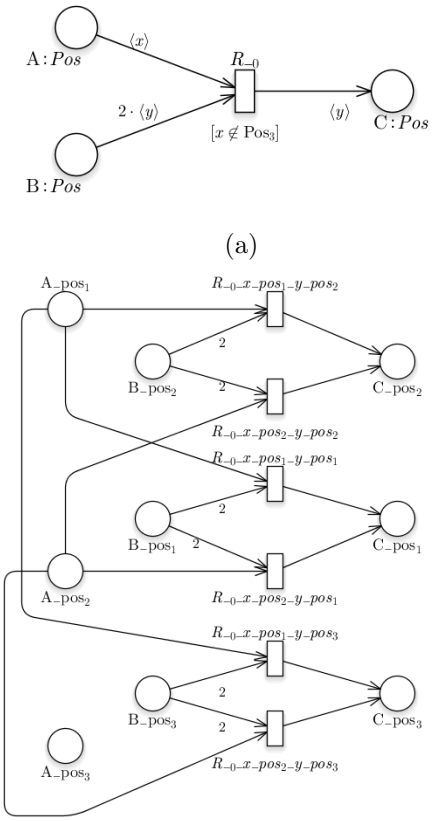


Figure 2.3: Unfolding example considering the first transition of the SSN model depicted in Fig. 2.2.

Therefore, to represent the behavior of an SSN model with an ODEs system consists of translating it into a standard SPN via an *unfolding* procedure, and then of applying the method discussed above which requires one ODE for each place of the resulting model [92]. According to this, the ODEs system introduced in Eq.s 2.4, which well approximates the stochastic behavior of an SSN model [9], becomes:

$$\frac{dx_{p,c}(\nu)}{d\nu} = \sum_{\langle t',c'\rangle \in E(t',x(\nu)) \ \wedge \ t'\in T \wedge \ c'\in cd(t)} \varphi(x(\nu),t',c')(L[p,t'](c')[c]) \tag{2.7}$$

where $x_{p,c}(\nu)$ is the average number of tokens of color c in the place p at time ν, $L[p,t'](c')[c] = O[p,t'](c')[c] - I[p,t'](c')[c]$, T is the set of transitions of the SSN, and $E(t',x(\nu))$ the set of the enabled instances of $\langle t',c'\rangle$ in $x(\nu)$, i.e. the

vector of the average number of tokens at time ν for each place and possible color tuple. In this case Eq. 2.6 becomes

$$\varphi(x(\nu), t', c') = \omega(t', c') \prod_{\langle p_j, c'' \rangle |\ p \in {}^\bullet\mathbf{t}' \ \wedge\ c'' \in cd(p_j)} x_{p_j, c''}(\nu)^{I[p_j, t'](c'')[c']}. \tag{2.8}$$

2.4 Exploiting symmetries in Symmetric Net

The SSN syntax has been devised to exploit the symmetries present in the modeled system to simplify its analysis. In this section, to show how system symmetries can be automatically derived by SSN, we introduced as example an extension of the SIR model, which is more suitable to show the advantages of using the symmetries [10]. In detail, this model, depicted in Fig. 2.4, has three places **S**, **I**, and **R** representing the three types of individuals: susceptible, infected, and recovered. Furthermore we defined two color classes *Age* and *Loc* modeling the age and the locations of the members of the population, respectively. The former color class is defined by three static sub-classes, $Age = \{Newborn,\ Young,\ Old\}$, identifying people of the same age, differently the latter class is partitioned into two static sub-classes $Loc = \{Loc1,\ Loc2\}$ defining two different zones. In such manner, each token is associated with an age and a location information. Therefore, the events that might occur are (i) the infection of a susceptible after the contact with an infected one, modeled by the transition *Infection*, (ii) the recovery from the disease, represented by the transition *Recovery*, (iii) the aging through the transition *Aging*, and (iv) the movements between the two zones through the transition *Moving*. All the arcs have cardinality one, except the arc connecting the transition *Infection* to place **I** which has cardinality 2 (which is expressed by the sum of the two color instances representing the infected and susceptible become infected), representing the susceptible becoming infected. Let us note that the infection may occur only if the individuals are in same locations, which is represented by the transition *Infection* guard. Finally, the color domain of all places is defined by $Age \times Loc$.

The constraints on the syntax of SSNs allow the automatic exploitation of the symmetries of the model through a symbolic representation of markings. This idea is based on the contest of the Symbolic Marking (SM) which is a compact representations for sets of equivalent ordinary markings, where the

actual color of tokens is abstracted away, but the ability to distinguish tokens with different colors and to establish their static subclass is retained. In detail, we defined that a SM, $\widehat{\boldsymbol{m}}$, is an equivalence class of ordinary markings, where two markings are equivalent if one can be obtained from the other by applying a color permutation preserving static subclasses.

For instance considering the color class Loc partitioned in $Loc_1 = \{l1, l2\}$

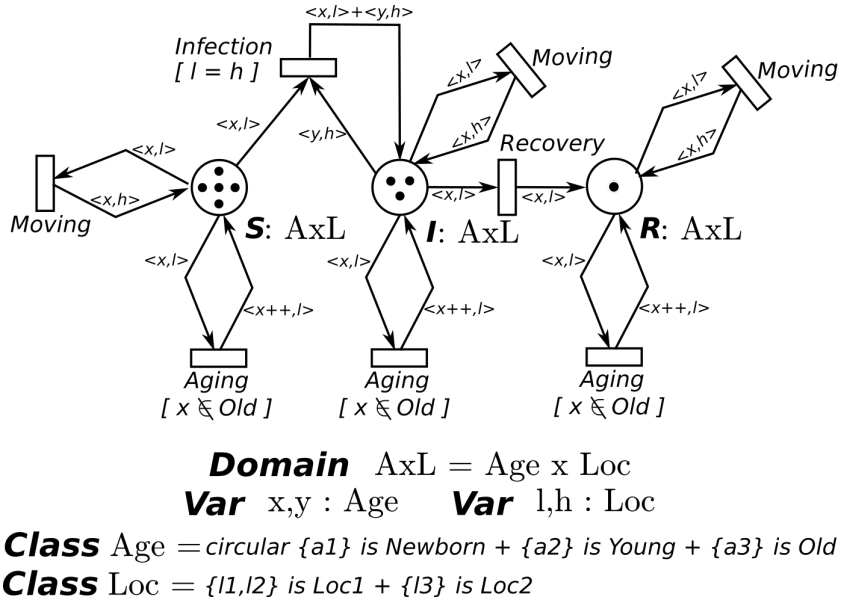


Figure 2.4: The SSN representing the SIR model where the age and locations of the population are taken in account by the two color classes *Age* and *Loc*, respectively.

and $Loc_2 = \{l3\}$, the two markings representing two infected individuals $(a1, a2)$ in the same zone (Loc_1) and one $(a3)$ in the second zone (Loc_2): $\boldsymbol{m} = I(\langle a1, l2\rangle + \langle a2, l2\rangle + \langle a3, l3\rangle)$ and $\boldsymbol{m'} = I(\langle a1, l1\rangle + \langle a2, l1\rangle + \langle a3, l3\rangle)$ belong to the same SM, since we may obtain $\boldsymbol{m'}$ from $\boldsymbol{m}$ by applying the permutation that exchanges the colors $l1$ and $l2$, and vice versa. Observe that instead, marking $\boldsymbol{m''} = I(\langle a1, l3\rangle + \langle a2, l3\rangle + \langle a3, l1\rangle)$ does not belong to the same SM of $\boldsymbol{m}$ and $\boldsymbol{m'}$ since it is not possible to permute $l1$ and $l3$ because they belong to different static subclasses.

Given two equivalent markings $\boldsymbol{m}_1$ and $\boldsymbol{m}_2$ (belonging to the same SM, $\widehat{\boldsymbol{m}}$) it is possible to show that there is a 1:1 correspondence between the transition instances enabled in $\boldsymbol{m}_1$ and those enabled in $\boldsymbol{m}_2$. Indeed, if s is the permutation that allows to obtain marking $\boldsymbol{m}_2$ from $\boldsymbol{m}_1$, i.e. $\boldsymbol{m}_2 = s.\boldsymbol{m}_1$ (where

the notation $s.x$ represents a permutation of elements of the vector x), for each transition instance $\langle t, c\rangle$ enabled in $\boldsymbol{m}_1$ there exists a transition instance $\langle t, s.c\rangle$ enabled in $\boldsymbol{m}_2$ with the same rate. Moreover markings $\boldsymbol{m}_1'$ reached by firing $\langle t, c\rangle$ in $\boldsymbol{m}_1$, and $\boldsymbol{m}_2'$ reached by firing $\langle t, s.c\rangle$ in $\boldsymbol{m}_2$ are equivalent (i.e. they belong to the same SM $\widehat{\boldsymbol{m}}'$: indeed $\boldsymbol{m}_2' = s.\boldsymbol{m}_1'$).
In particular, given this aggregation induced by the markings equivalence on the CTMC describing the behaviors in time of the SSN model, it is possible to prove that the strong and exact lumpability conditions are satisfied, so that all the states within an aggregate are equally likely for every initial distribution. Thus, by exploiting the system symmetries in such a way that not only qualitative analysis but also quantitative analysis can be performed on the reduced graph, called Symbolic Reachability Graph (SRG). In detail, by defining a symbolic firing rule on the symbolic marking it is possible to build the SRG directly starting from a symbolic initial marking, without building the RG and then grouping markings into equivalence classes. The formal description of the symbolic firing rule is given in [30], but the idea is that ordinary transition instance firings are grouped in a symbolic transition instance firing, where the ordinary firing instance are obtained by valid assignments of colors to the subclasses defined by tokens with the same distribution and belonging to the same static subclass. Finally from the SRG it is possible to directly derive the isomorphic lumped CTMC [30], from which the most of quantitative proprieties computed from the original RG and CTMC can still be obtained. Hereafter, for simplicity, we will focus on the ODEs system which can be derived from the CTMC, but equivalent definitions can be made considering the Kolmogorov equations.

Let us now define the *static partition* of a place color domain: if $\widetilde{C_i}$ is the set of static subclasses of class C_i, the static partition of color domain $C_1 \times C_2 \times C3$ is $\tilde{C}_1 \times \tilde{C}_2 \times \tilde{C}3$; moreover, abusing notation, if $c = \langle c_1, c_2, c_3\rangle$ is a color in $C_1 \times C_2 \times C3$, let us denote $d(c) = \langle d(c_1), d(c_2), d(c_3)\rangle$ the static partition it belongs to.
Given a place p with color domain $cd(p)$, the average number of tokens of color $c \in cd(p)$ (which is denoted as $\bar{m}(p)(c)$) has the following property: $\forall c, c' : d(c) = d(c')$, $\bar{m}(p)(c) = \bar{m}(p)(c')$, i.e. the average number of colored tokens in p is the same for all colors in the same static partition element. This

can be proved as follows:

$$\bar{m}(p)(c) = \sum_{m \in RS} \pi(m)\ m(p)(c) \tag{2.9}$$

(where $\pi(m)$ is the probability of marking m, either at a given time t or in steady state). Let $c' : d(c) = d(c')$ be a color in the same static partition element: i.e. $c' = s.c$ for some permutation s preserving static partitions; now applying permutation s to equation (2.9) we obtain

$$\bar{m}(p)(s.c) = \sum_{s.m : m \in RS} \pi(s.m)\ s.m(p)(s.c) \tag{2.10}$$

Due to the hypothesis of initial symmetric marking, $\{s.m | m \in RS\} = RS$, moreover $s.m$ is in the same SM as m and hence have the same probability, i.e. $\pi(m) = \pi(s.m)$, finally $s.m(p)(s.c) = m(p)(c)$ by definition of $s.m$.

Recalling the same idea behind the definition of symbolic marking, we can define the notion of *symbolic* ODE (Symbolic ODE ($\widehat{ODE}$)): a compact representation for a set of equivalent ODE, where the actual color identity is abstracted away, but the ability to distinguish different colors and to establish their static subclass is retained. Thus, due to symmetries, each summation over the color domain of a given transition t in an ODE may be computed efficiently by grouping instances with "similar" rate and same number of tokens moved into or out of the place. In detail, a $\widehat{ODE}$ is an equivalence class of ODEs, such that two ODEs are equivalent if one can be obtained from the other by applying a color permutation preserving static subclasses. Its representation can be formalized as follows:

Definition 4 (Symbolic ODE representation)**.** *A representation $\mathcal{R}$ of a* $\widehat{\text{ODE}}$ *is a tuple:*

$$\mathcal{R} = \langle \widetilde{c}, \widetilde{s}, \mathcal{M} \rangle$$

where

- $\widetilde{c} : C \to \mathbb{N}$ *is a mapping from a color to the index of its corresponding color class.*
- $\widetilde{s} : C \to \mathbb{N}$ *is a mapping from a color to the index of its corresponding static subclass.*

- $\mathcal{M} : \{x\} \to \{\widehat{x}\}$ *is mapping from an ODE variable to the corresponding symbolic variable in* $\widehat{\text{ODE}}$ *such that:*

$$\mathcal{M}(x_{p_{i,c_1,\ldots,c_n}}) = \widehat{x}_{p_{i,Z^1_{\widetilde{c}(c_1),\widetilde{s}(c_1)},\ldots,Z^k_{\widetilde{c}(c_n),\widetilde{s}(c_n)}}}$$

where $\forall Z^j_{\widetilde{c}(c),\widetilde{s}(c)}, Z^k_{\widetilde{c}(c'),\widetilde{s}(c')}\ k = j \Leftrightarrow c = c'$

For instance, if we consider the two differential equations, given in Eq. 2.7, representing the susceptible individuals in the first color classes, i.e. $\{a1\} \in Newborn$ and $\{l1,\ l2\} \in Loc_1$, which are defined as follow

$$\begin{aligned}
\frac{dx_{S,a1,l1}(\nu)}{d\nu} =& \varphi(x(\nu), Moving, \langle a1, l2, a1, l1\rangle) + \varphi(x(\nu), Moving, \langle a1, l3, a1, l1\rangle) - \\
& \varphi(x(\nu), Moving, \langle a1, l1, a1, l2\rangle) - \varphi(x(\nu), Moving, \langle a1, l1, a1, l3\rangle) - \\
& \varphi(x(\nu), Aging, \langle a1, l1, a2, l1\rangle) - \varphi(x(\nu), Infection, \langle a1, l1, a1, l1\rangle) - \\
& \varphi(x(\nu), Infection, \langle a1, l1, a2, l1\rangle) - \varphi(x(\nu), Infection, \langle a1, l1, a3, l1\rangle) \\
\frac{dx_{S,a1,l2}(\nu)}{d\nu} =& \varphi(x(\nu), Moving, \langle a1, l1, a1, l2\rangle) + \varphi(x(\nu), Moving, \langle a1, l3, a1, l2\rangle) - \\
& \varphi(x(\nu), Moving, \langle a1, l2, a1, l1\rangle) - \varphi(x(\nu), Moving, \langle a1, l2, a1, l3\rangle) - \\
& \varphi(x(\nu), Aging, \langle a1, l2, a2, l2\rangle) - \varphi(x(\nu), Infection, \langle a1, l2, a1, l2\rangle) - \\
& \varphi(x(\nu), Infection, \langle a1, l2, a2, l2\rangle) - \varphi(x(\nu), Infection, \langle a1, l2, a3, l2\rangle)
\end{aligned}$$

the corresponding $\widehat{ODEs}$ is constituted by only one equation, that is:

$$\begin{aligned}
\frac{d\widehat{x}_{S_{Z^1_{1,1},Z^2_{2,1}}}(\nu)}{dv} =& \varphi(\widehat{x}(\nu), Moving, \langle Z^1_{1,1}, Z^2_{2,1}, Z^3_{1,1}, Z^4_{2,1}\rangle) + \\
& \varphi(\widehat{x}(\nu), Moving, \langle Z^1_{1,1}, Z^2_{2,2}, Z^3_{1,1}, Z^4_{2,1}\rangle) - \\
& \varphi(\widehat{x}(\nu), Moving, \langle Z^1_{1,1}, Z^2_{2,1}, Z^3_{1,1}, Z^4_{2,1}\rangle) - \\
& \varphi(\widehat{x}(\nu), Moving, \langle Z^1_{1,1}, Z^2_{2,1}, Z^3_{1,1}, Z^4_{2,2}\rangle) - \\
& \varphi(\widehat{x}(\nu), Aging, \langle Z^1_{1,1}, Z^2_{2,1}, Z^3_{1,2}, Z^4_{2,1}\rangle) - \\
& \varphi(\widehat{x}(\nu), Infection, \langle Z^1_{1,1}, Z^2_{2,1}, Z^3_{1,1}, Z^4_{2,1}\rangle) - \\
& \varphi(\widehat{x}(\nu), Infection, \langle Z^1_{1,1}, Z^2_{2,1}, Z^3_{1,2}, Z^4_{2,1}\rangle) - \\
& \varphi(\widehat{x}(\nu), Infection, \langle Z^1_{1,1}, Z^2_{2,1}, Z^3_{1,3}, Z^4_{2,1}\rangle) \\
=& \varphi(\widehat{x}(\nu), Moving, \langle Z^1_{1,1}, Z^2_{2,2}, Z^3_{1,1}, Z^4_{2,1}\rangle) - \\
& \varphi(\widehat{x}(\nu), Moving, \langle Z^1_{1,1}, Z^2_{2,1}, Z^3_{1,1}, Z^4_{2,2}\rangle) - \\
& \varphi(\widehat{x}(\nu), Aging, \langle Z^1_{1,1}, Z^2_{2,1}, Z^3_{1,2}, Z^4_{2,1}\rangle) - \\
& \varphi(\widehat{x}(\nu), Infection, \langle Z^1_{1,1}, Z^2_{2,1}, Z^3_{1,1}, Z^4_{2,1}\rangle) - \\
& \varphi(\widehat{x}(\nu), Infection, \langle Z^1_{1,1}, Z^2_{2,1}, Z^3_{1,2}, Z^4_{2,1}\rangle) - \\
& \varphi(\widehat{x}(\nu), Infection, \langle Z^1_{1,1}, Z^2_{2,1}, Z^3_{1,3}, Z^4_{2,1}\rangle).
\end{aligned}$$

Where the place $x_{S,a1,l1}$ becomes $\widehat{x}_{S_{Z^1_{1,1},Z^2_{2,1}}}$ since $a1$ belongs to the first color class Age ($\widetilde{c}(a1)$) and its respective first subclass $Newborn$ ($\widetilde{s}(a1)$), while $l1$ belongs to the second color class Loc and its respective first subclass $Loc1$. Similar reasoning could be done considering the intensity function defined in Eq. 2.8, for instance $\varphi(x(\nu), Moving, \langle a1, l2, a1, l1\rangle)$ becomes $\varphi(\widehat{x}(\nu), Moving, \langle Z^1_{1,1}, Z^2_{2,1}, Z^3_{1,1}, Z^4_{2,1}\rangle)$ since $a1$ belongs to the first color class Age and its respective first subclass $Newborn$, and $l1$, $l2$ belong to the second color class Loc and its respective first subclass $Loc1$. The mathematical details and the algorithm to derive from an ODE system the corresponding $\widehat{ODE}$ are given in [10]. Roughly, the algorithm takes in input all the variables of an ODE and replaces them with the corresponding representative symbolic variables. Where an ODE variable, $x_{p_{c_1,\ldots,c_n}}$, is defined by the corresponding place name (e.g. p) and the list of the associated color (e.g. $c_1, \ldots, c_n$). While an $\widehat{ODE}$ variable is defined so that it encodes for each $Z^k_{i,j}$, their indexes i, j, and k (i.e. color, subclass and index respectively). Hence, for each ODE variable the algorithm derives the corresponding symbolic variable.

Now we can observe that symbolic variables representing the distribution of equivalent tokens (w.r.t. a colored permutation preserving static subclasses) in a place are the same so that we can substitute all these symbolic variables with a representative one. This leads automatically to a reduction in the number of $\widehat{ODEs}$ and a reduction on the terms which constitute the equation.

2.4.1 Arc functions syntax

In this section we introduce some of the notations and definitions exploited in the new method for deriving the set of $\widehat{ODE}$ in Sec. 6.1, without the complete unfolding of the SSN. This method is based on the symbolic manipulation of expressions of a language $\mathcal{L}$ through a set of operators (difference, transpose). Before explaining the elements of language $\mathcal{L}$, let us recall some objects introduced in Sec. 2.3, starting with the function var which assigns to each transition $t \in T$ a set of variables, each taking values in a given color class C_i of $\mathcal{C}$ (the variable's type). A transition t may have an associated *guard* $g = \Theta(t)$, i.e., a predicate defined on $cd(t)$ (the default/implicit guard being the constant $true$): in that case the transition color domain $cd(t)$ is restricted to those instances of t verifying the guard. In details, Θ is a function assigning a guard to each transition t, where a guard is defined as boolean function defined on $var(t)$, $\Theta(t) : cd(t) \rightarrow \{true, false\}$, and it is denoted through a boolean expression whose terms are basic predicates on $var(t)$. In this contest, I and O associate each pair $(p,t) \in P \times T$ with a map $I[p,t]$, $O[p,t]$ (called arc function), annotating a corresponding oriented arc connecting t and p (if the arc does not exist the corresponding function is the empty constant).
Let us define an SSN function labeling an arc connecting transition t and place p as a mapping $W(p,t) : cd(t) \rightarrow Bag(cd(p))$ whose form is:

$$W(p,t) = \sum_i \lambda_i T_i[g_i], \quad \lambda_i \in \mathbb{N}, \tag{2.11}$$

in which the sum is a multiset sum and $[g_i]$ is the guard, λ_i are scalars and T_i are function-tuples. Where a function-tuple T, denoted by $\langle f_1, \ldots, f_k \rangle$, as the Cartesian product of *class functions* f_i. Each class-function f is a linear function defined on a subset of variables of $var(t)$ of the same type. Let $var_{C_i}(t) = \{v_1, \ldots v_m\}$ be the subset of variables in $var(t)$ of type C_i, and $\widetilde{C_i}$

the set of static subclasses of C_i, then $f : C_i^m \to Bag[C_i]$ is so defined:

$$f = \sum_{k=1}^{m} \alpha_k v_k + \sum_{q=1}^{|\widetilde{C_i}|} \beta_q S_{i,q} \tag{2.12}$$

where $\alpha_k, \beta_k \in \mathbb{Z}$. $S_{i,q}$ is a constant functions called *diffusion/synchronization*, which maps its argument to the set $C_{i,q}$. In particular, the following equivalence holds: $\sum_q S_{i,q} = S_i$. Scalars must be such that no negative coefficient result from the evaluation of f for any color satisfying the guard possibly associated with the function-tuple or transition. Fixed an order on $var(t)$ defining $cd(t)$, consistent with the local ordering on $var_{C_i}(t)$, and assuming that the color domain of class-functions naturally extends to the tuple's color domain, the semantics of a function-tuple $T := \langle f_1, \ldots, f_k \rangle$ is $T(c) = \langle f_1(c), \ldots, f_k(c) \rangle$, where the Cartesian product of multisets is defined as: let $m_1 \in Bag[A], m_2 \in Bag[B]$, then $\langle m_1, m_2 \rangle$ is an element of $Bag[A \times B]$ such that $\forall a \in A, b \in B$, $\langle m_1, m_2 \rangle(\langle a, b \rangle) = m_1(a) \cdot m_1(b)$. A guarded tuple $T[g]$, where g is a predicate defined similarly to transition guards, maps each element $c \in cd(t)$ in $T(c)$ if $g(c) = true$, in $\emptyset$ otherwise.

Let us now introduce the syntax of the language used to express the SSN structural relations. The expressions of $\mathcal{L}$ have a syntax which resembles the arc function syntax defined in Eq.s 2.11 and 2.12, but there are some additional constraints on the functions used as elementary building blocks, however the expressive power is actually extended. Thus, the elements of $\mathcal{L}$ are an extension of the SSN arc functions, and have got the following syntax:

$$\sum_i \lambda_i [g_i'] T_i [g_i], \quad \lambda_i \in \mathbb{N} \tag{2.13}$$

where $[g_i]$ and $[g_i']$ are respectively called *guard* and *filter*, which denote functions $D \to Bag[D']$. D and D' are in turn defined as Cartesian products of color classes, and g_i and g_i' take the form of SSN standard predicates. Observe that the SSN arc function syntax may include guards but not filters.
The components in a tuple T_j are in one-to-one correspondence with the elements in the Cartesian product D': they are *intersections* ($\cap$) of basic class functions $D \to Bag[C]$ from set $BS = \{v, S - v, S_C, S_{C_k}\}$, where C is one of the basic color classes in D', v is a variable of type C, and C_k is a static subclass of C. When class C occurs n times in D' a family $\{v_i\}$, $i : 1, .., n$, of type

C variables may be used to highlight that v_i (also called *projection*) refers the i^{th} type C element on a tuple of D'. Symbols S, S_{C_k} are constants mapping to a global (sub)class. To keep the presentation simple, ordered classes are not considered here, but the extension is straightforward.
The functions in BS are a subset of SSN class-functions, however, any class-function f can be expressed[2] in terms of BS. A function tuple T_j, individually considered, maps to multisets with multiplicities ≤ 1 (i.e., *sets*) of colors. Filters and guards are boolean expressions whose terms are basic predicates: variable symbols occurring on the guard g_j and on the tuple T_j have exactly the same meaning, whereas a variable $v_i : C$ on g'_j refers the i^{th} type C element of any color tuple represented by T_j.
Language $\mathcal{L}$ is closed with respect *transpose* and *difference* operators, defined as follows:

Definition 5 (Transpose)**.** *Let $f : D \to Bag[D']$ be a function, its transpose $f^t : D' \to Bag[D]$ is defined as: $f^t(x)[y] = f(y)[x], \forall x \in D', y \in D$.*

Definition 6 (Difference)**.** *Let $f, g : D \to Bag[D']$ be two functions. The difference $f - g : D \to Bag[D']$ is defined as: $f - g(x) = f(x) - g(x), \forall x \in D$.*

The language, with its operators and properties, is the key formal tools to define the $\widehat{ODE}$ characterizing an SSN model without unfolding it. In particular, the difference and transpose operators allow the relations $\mathcal{R}$ and $\mathcal{A}$ to be defined and expressed in a symbolic form. Assuming that nodes p and t are connected, function $\mathcal{R}(p,t)$, called *Removed By*, defines which instances (t, c') withdraw tokens of color $c \in cd(p)$ from place p. Function $\mathcal{A}(p,t)$, called *Added By*, defines which instances (t, c') put tokens of color $c \in cd(p)$ into p. Both functions map to multisets: $\mathcal{R}(p,t)(c)[c']$ and $\mathcal{A}(p,t)(c)[c']$ are the number of tokens of color c withdrawn/added by an instance (t, c') from/to p.

$$\mathcal{R}(p,t) : cd(p) \to Bag[cd(t)];\ \mathcal{R}(p,t) = (I[p,t] - O[p,t])^t,$$

$$\mathcal{A}(p,t) : cd(p) \to Bag[cd(t)];\ \mathcal{A}(p,t) = (O[p,t] - I[p,t])^t.$$

The symbolic derivation of $\mathcal{A}(p,t)$ and $\mathcal{R}(p,t)$ is based on the calculus presented in [25, 24].
Let us show an example considering the SIR model depicted in Fig. 2.4B).

[2]This can be obtained through the intersection, which is not included in the SSN arc expressions syntax.

Place **S** (whose color domain is $Age \times Loc$) is connected to transition $Infection$ with color domain $x \in Age, y \in Age, l \in Loc, h \in Loc$ and with guard $l = h$. The expression for $\mathcal{R}(Infection, \mathbf{S}) = (\langle x, l \rangle [l = h])^t$ is $1\langle age_1, S_{Age}, loc_1, loc_1 \rangle$ denoting a function from $cd(\mathbf{S})$ to $Bag[cd(Infection)]$. Here the names age_1 and loc_1 indicate respectively the first occurrence of classes Age and Loc in $cd(\mathbf{S})$, while the color identifying an instance of $Infection$ is indicated as a 4-tuple $\langle x, y, l, h \rangle [l = h]$. As expected the instances of $Infection$ that remove tokens of color $\langle x, l \rangle$ from **S** are those with $x = age_1, l = loc_1, \forall y, h = loc_1$.
Let us now consider place **I** and transition $Moving$. $\mathcal{A}(Moving, \mathbf{I}) = ((\langle x, l \rangle)^t = 1\langle age_1, S_{Loc}, loc_1 \rangle$ where an instance of $Moving$ is indicated as a 4-tuple $\langle x, l, h \rangle$. The $\mathcal{A}(Infection, \mathbf{I})$ expression can be interpreted as follows: the instances of $Moving$ adding tokens of color $\langle age_1, loc_1 \rangle$ into **I** are those with any $l \in Loc$ and with $y = age_1$, $h = loc_1$.
Finally, if we consider the differential equation for place p and color $c \in cd(p)$, Eq. 2.7, we can rewrite it in terms of $\mathcal{A}()/\mathcal{R}()$ as follows:

$$\frac{dx[p,c]}{d\nu} = \sum_{(t,c'):p \in t^\bullet, c' \in \mathcal{A}(p,t)(c)} \varphi(x(\nu), t, c') \cdot \mathcal{A}(p,t)(c)[c'] - \sum_{(t,c'):p \in {}^\bullet t, c' \in \mathcal{R}(p,t)(c)} \varphi(x(\nu), t, c') \cdot \mathcal{R}(p,t)(c)[c'] \tag{2.14}$$

where $x[p,c]$ is the (average) number of c-colored tokens in place p at time ν (to keep notation simpler we will omit time dependency), then it is possible to observe the color dependencies of the intensities and the $\mathcal{A}()/\mathcal{R}()$ functions. Indeed, each sum spans over all instances (t, c') that add (positive terms) or withdraw (negative terms) tokens of color c to/from p. The *intensity* φ of (t, c'), which depends on the distribution of colored tokens in t's preset, multiplied by the number of tokens of color c added to or withdrawn from p by (t, c') ($\mathcal{A}(p,t)(c)[c']$ or $\mathcal{R}(p,t)(c)[c']$) gives the actual flow of tokens into or out of p.

Chapter 3

Solution techniques

In this Chapter we describe the main simulating techniques to obtain the temporal evolution of systems in which entities interacting with each other are involved.

Let us recall from Chapter 2 that the stochastic temporal evolution of a given system is derived by solving a set of CMEs (eq.s 2.2), one for each possible state of the system. However, this approach is unfeasible in case of complex systems, leading to the well-known *State-Space Explosion* problem. Two different approaches can be exploited to deal with this point: the stochastic simulation or the deterministic approximation.

In 1976 Daniel Gillespie proposed an exact stochastic algorithm to simulate chemical or biochemical systems of reactions, called Stochastic Simulation Algorithm (SSA) [57], that can be exploited to obtain trajectories distributed in according to the solution of the CME. Since this method explicitly simulates all the events that might occur in the system, it can become slower with an increasing number of system molecules. For this reason several algorithms [56, 58, 22] were proposed for obtaining approximations of the SSA with lower computational costs. One of the most common is the **τ-leaping algorithm** [58], which exploits a Poisson approximation to *leap over* many fast reactions and to approximate the stochastic behavior of the system. Indeed, it provides a natural connection between the SSA in the discrete stochastic regime and

the explicit Euler method applied in the continuous deterministic approximation, in which the system behaviour is approximated by a deterministic model described through an ODEs system [83].
The deterministic approximation is instead characterized by a computational cost significantly lower than the stochastic simulations, however in those systems in which randomness plays an important role, this approach is not able to provide a good approximation of the real system behaviour (for instance relevant random features like variance, bimodality, and tail behavior). In literature [63, 64, 66] there are approaches based on the co-simulation of discrete (i.e., stochastic) and continuous events (i.e., deterministic) to study the evolution of the system. These approaches will not be analysed in this thesis.
Thus, in this chapter we give the mathematical details regarding 1) the stochastic simulations given by the SSA and its approximation, **τ-leaping algorithm**, and 2) the deterministic approximation given by the Kurtz's theorem [83].

3.1 Stochastic Simulation Algorithm

The SSA [57] was proposed firstly as an algorithm to simulate chemical or biochemical systems of reactions, but it could be easily extended to simulated different systems. Since in this thesis the algorithm is directly applied to SPN models, here we describe it by using the PNs notation. Thus, let us consider for simplicity an SPN model of n_p places and n_t transitions, but all the formulas can be extended to the SSN formalism by adding the color dependencies to each transition and place. The state of the system at a specif time point ν is described by the marking m_i, i.e., the number of tokens in each place at time ν. We assume that the system is well stirred and in thermal but not chemical equilibrium. We recall that the dynamics of the system follows the CME introduced in Eq.s 2.2, that is:

$$\frac{d\pi(m_i,\nu)}{d\nu} = \sum_{m_k} \pi(m_k,\nu) q_{m_k,m_i} \qquad m_i, m_k \in \mathbb{S} \tag{3.1}$$

where $\pi(m_i,\nu)$ represents the probability to be in marking m_i at time ν, and q_{m_k,m_i} the velocity to reach the marking m_i from m_k.
Let us now define the key factor of the SSA, which is the *propensity function* $a_t(m_i)$ of the transition t with marking m_i such that $a_t(m_i)d\nu$ is the probability that the transition t will fire in the next time interval $[\nu, \nu + d\nu]$. We

underline that the propensity function of a transition is strictly connected to the velocity of changing markings. Indeed, defining the marking change vector associated with a transition t as $v_t \equiv (v_{1t}, \ldots, v_{n_p t})$, then for each markings m_i and m_k, with m_i reachable from m_k, there exists always and uniquely a transition t such that $m_i = m_k + v_t$. Thus, the Eq.s 2.2 can be rewritten in terms of propensity functions as follows:

$$\frac{d\pi(m_i, \nu)}{d\nu} = \sum_{t \in T} [\pi(m_i - v_t, \nu) a_t(m_i - v_t) - \pi(m_i, \nu) a_t(m_i)], \tag{3.2}$$

where the first term represents the probability to reach the state m_i given the firing of the transition t in the time interval $[\nu, \nu + d\nu]$, and the second term the probability to stay in m_i given no firing transitions in the same time interval.

Finally it is straightforward to show that the time necessary, $\hat{\Delta(\nu)}$, to the next occurring firing transition is the exponentially distributed random variable with mean $\frac{1}{a_0(m_i)}$, with $a_0(m_i) \equiv \sum_{t \in T} a_t(m_i)$ (called **total propensity**). In particular, associating an integer index to each transition exploiting a bijective function $index : T \to \{1, \ldots, n_t\}$ s.t. $index(t) = k,\ t \in T$, than the index of the firing transition t is the integer random variable with point probability $\frac{a_t(m_i)}{a_0(m_i)}$. To advance the system from m_i at time ν, the SSA generates two random numbers r_1 and r_2 uniformly sampled in the unit interval, and then takes the time of the next reaction to be $\nu + \hat{\Delta(\nu)}$ where

$$\hat{\Delta(\nu)} = \frac{1}{a_0(m_i)} ln(\frac{1}{r_1}), \tag{3.3}$$

and the index for the next transition to be the smallest integer j satisfying

$$\sum_{k=1}^{j} a_{index^{-1}(k)} > r_2 a_0(m_i). \tag{3.4}$$

The system state is then updated according to the marking change vector associated with the transition $t = index^{-1}(j)$, and this process gets repeated until some final time or condition is reached.

3.2 τ-leaping approximation

Although the SSA is an exact stochastic simulation method, it has some practical issues due to computational expensiveness due to the tracking of every

event, which makes this method impractical for many very complex problems (e.g., a huge number of entities in the model). For this reason a new strategy, called τ-leaping approximation, has been proposed in [58] and it starts from the observation that, instead of considering an infinitesimal and negative exponential distributed time interval and then make a simulation proceeding one infinitesimal interval after the other, it is possible to consider a larger and dynamic time step, τ, during which multiple transitions can fire leading to a substantial change of the state. In this way the simulation proceeds faster than before as long as small approximations are made. Indeed, we have to identify the length of this time interval τ such that each transition can fire a number of times i) without losing too much information about the evolution of the system, and ii) without having an excessive discretization that leads to something computationally similar to SSA. In this contest, we can define a new object, denoted as $K_t(\tau, m_\nu, \nu)$, for counting the number of times, given the marking m_ν at time ν, that the transition t will fire in the time interval $[\nu, \nu + \tau)$. Indeed, if τ is small enough that no propensity function change its value "appreciably", a requirement that is called **the leap condition**, then $K_t(\tau, m_\nu, \nu)$ can be approximated by a Poisson random variable $\mathcal{P}(a_t(m_\nu), \tau)$ with mean and variance $a_t(m_i)\tau$. Therefore we can defined the **τ-leaping approximation** the procedure to choose a value of τ satisfying the leap condition, and then update the marking to time $\nu + \tau$ in according to the Poisson approximation, i.e.

$$m_{\nu+\tau} = m_\nu + \sum_{t \in T} v_t \mathcal{P}(a_t(m_\nu), \tau). \tag{3.5}$$

A critical aspect in τ-leaping approach is the definition of an efficient τ, that is compatible with the leap condition. A first solution proposed by Gillespie was to choose τ such that the expected change in each propensity function $a_t(m_\nu)$ during the leap is limited by the total propensity function and an error control parameter ϵ, i.e.,

$$|a_t(m_{\nu+\tau}) - a_t(m_\nu)| < a_0(m_\nu)\epsilon, \qquad \forall t \in T. \tag{3.6}$$

Successively, Gillespie and Petzold [59] showed that the largest value of τ that satisfies Eq. 3.6 can be estimated by the following equation

$$\tau = min_{t \in T} \left\{ \frac{a_0(m_\nu)\epsilon}{|\mu_t(m_\nu)|}, \frac{(a_0(m_\nu)\epsilon)^2}{\sigma_t^2(m_\nu)} \right\}, \tag{3.7}$$

where

$$
\begin{aligned}
f_{t,t'}(m_\nu) &= \sum_{p \in P} \frac{\partial a_t(m_\nu)}{\partial m_\nu(p)} v_{pt'}, \\
\mu_t(m_\nu) &= \sum_{t' \in T} f_{t,t'}(m_\nu) a_{t'}(m_\nu), \\
\sigma_t^2(m_\nu) &= \sum_{t' \in T} f_{t,t'}^2(m_\nu) a_{t'}(m_\nu),
\end{aligned}
\tag{3.8}
$$

considering $t,\ t' \in T$, $m_\nu(p)$ the marking of place p at time ν, and $v_{pt'}$ the state change of the place p with the firing of t'.

Because the Poisson random variable is unbounded, it is possible that the Poisson approximation might fire a transition so many times that the number of tokens in one of its input places becomes negative. In this case, a simple manner to deal with the negativity is to decrease the τ value until no places have a negative number of tokens. Indeed, this approach is computational expensive so in [23] a modified Poisson τ-leaping procedure was introduced to better resolve the negativity problem. The idea is based on identifying the *critical* transitions, which are characterized by a positive propensity function that is currently within n_c firings of exhausting the tokens in one of the its input places. Therefore, a second control parameter n_c, a positive integer that is usually set somewhere between 2 and 20, is introduced in the algorithm [23]. Successively the identification of the critical transitions, the modified algorithm simulates the critical ones using an adapted version of the SSA, and the remaining transitions exploiting the Poisson τ-leaping method. Since no more than one firing of a critical transition can occur during a leap, the probability of producing a negative number of tokens is reduced to nearly zero.

3.3 Deterministic approximation

We can now recall that the deterministic approximation (also called *fluid approximation* or *mean field analysis* [16]) which can be derived by the Kurtz's theorem [83]. Observe that we have expressed it in a form that is directly related to the definition of SPN. First of all, let us define a principal property which has to characterize the CTMC derived from the SPN, because the fluid approximation can be applied.

Definition 7. *Let $X_\eta(v)$ be a parametric family of Markov chains with $\eta \in \mathbb{N}$, and state spaces $S_\eta \subset \mathbb{Z}^k$, such a family is called density dependent iff there*

exists a continuous function $f(y,l),\ y \in \mathbb{R}^k, l \in \Delta(y)$ *such that the non-diagonal entries of the infinitesimal generator corresponding to* $X_\eta(v)$ *can be written in the form:*

$$q_{k,k+l} = \eta f\left(\frac{k}{\eta}, l\right) \tag{3.9}$$

and the initial state of the chain is ηx_0, $x_0 \in \mathbb{Z}^k$, *with probability one.*

Let $X(v)$ be a deterministic process denoting the solution of the ODE system

$$\frac{dX(v)}{dv} = \sum_{l \in \Delta(X(v))} f(X(v), l) \tag{3.10}$$

with initial condition $X(0) = x_0$, meaning that the initial value of the deterministic process $X(v)$ is equal to the initial state of the parametric family of density dependent Markov chains $X_\eta(v)$ divided by η.

In [83] Kurtz has showed that under relatively mild conditions on function f the following relation holds between the function $X(v)$ and a trajectory of the CTMC $X_\eta(v)$:

$$\forall \delta > 0 : \lim_{\eta \to \infty} P\left\{\sup_{u \leq v}\left|\frac{1}{\eta} X_\eta(u) - X(u)\right| > \delta\right\} = 0. \tag{3.11}$$

Informally this means that if we consider a sequence of CTMCs with increasing initial state and if we assume that the infinitesimal generators (i.e., the velocity to reach a specific state from a different state) corresponding to this sequence satisfy the conditions expressed by Eq. 3.9, then, as η becomes large, the behavior of the CTMC converges to the solution of the ODEs given in Eq. 3.10. In this situation, the probability of finding differences between a trajectory of the CTMC and the solution of the ODEs in a finite time horizon $(0, v)$ that are larger than a predefined arbitrarily small threshold, is zero.

In [9, 10] the authors showed the applicability of the fluid approximation considering the SPN and the Symbolic Net formalism, respectively. Thus the model behavior is then represented by one equation for each individual place in the SPN case or less considering the symbolic formalism, so that the computational complexity grows linearly with the number of places, while the size of the underlying CTMC grows exponentially. In this case the ODEs system obtained from the Kurtz's theorem, Eq. 3.10, becomes the equations system expressed by the Eq. 2.4.

Chapter 4

Flux Balance Analysis

High-throughput techniques in molecular cell biology (such as RNA seq and proteomics, based on single cell technology), allow to extract information on the gene repertoire activation and on occurring biochemical reactions. Indeed, this information could be part or have an influence on the PNs design (e.g., in the production of energy or of an enzyme, facilitating the formation of conditions, such as kinases or phosphorilases, able to change the overall phosphorilation dynamics and therefore the fluidity of the cytoplasm). For instance, metabolic networks can include intra-cellular reactions occurring in a cancer stem cell, or could be expanded to account for environmental metabolic processes, i.e., all those occurring outside the cells in the tumor microenvironment. This is particularly important to investigate the energetic details of cancer dynamics across different compartments (for example in the pancreas we could consider intra-cellular Langerhans islets, connective tissues, and blood compartments). Having a list of reactions and of the enzymes catalysing them, a system of differential equations can answer interesting biological questions. However, building such a system requires knowledge of rate coefficients for every reaction, which are difficult to measure, especially for enzyme catalysed reactions in vivo (the *in vitro* condition can rarely approximate the exact local *in vivo* condition). Therefore, techniques to analyse large metabolic models must get by with just stoichiometry and other constraints

on reaction rates, such as constant bounds.

Constraint-based approaches are mathematical modeling approaches based on the definition and manipulation of stoichiometric matrices, commonly used with optimization techniques, such as the use of linear and mixed-integer programming to maximize an objective function under specific constraints (+/- infinity are allowed).
In particular, the **Flux Balance Analysis** (FBA) [137] computes the distribution of reaction fluxes in a metabolic system at the equilibrium and finding the feasible fluxes under given constraints and an objective function to maximize/minimize. Several techniques, that further explore the solution space, have addressed the issue of choosing among indistinguishable optimal solutions, like the decomposition of the flux distribution into Elementary Flux Modes, or Flux Variability Analysis. Given that FBA only identifies the metabolic flux distribution under specific constraints, a dynamic extension to FBA has been formulated, namely Dynamic Flux Balance Analysis (DFBA), to obtain further information on the metabolite concentrations or on the dynamic characteristics of the metabolic fluxes. Exploiting this approach it is possible to explicitly model constraints on the fluxes given certain metabolic concentrations.
Finally, in this chapter we recall the basis and the mathematical details regarding both the FBA and DFBA techniques, and their solution techniques.

4.1 Flux Balance Analysis formalism

The first works [112, 166] regarding the FBA dates back to the early 1980s, they showed the possibility to construct flux balance equations using a metabolic map and to exploit linear programming for deriving the fluxes in a pathway. The FBA models start considering a system characterized by a large number of reactions and metabolites, each reaction associated with an inferior and superior concentration boundaries. Thus, the dynamic mass balances (i.e. the changes of metabolites concentration over the time) for each metabolite in the system can be expressed by the following set of ODEs (in matrix notation)

$$\frac{dx}{dt} = S \cdot v(x,t), \tag{4.1}$$

where $x \in \mathbb{R}^n$ denotes the concentration vector of all the n metabolites, $v \in \mathbb{R}^m$ is the flux vector describing the activity of all the m internal and exchange fluxes (which may depend by x and t), and $S \in \mathbb{Z}^{m \times n}$ is the stochiometric matrix.

The stoichiometry (the participant molecular species and their ratios) of chemical reactions is fixed, since this is determined by conservation of matter and of charge. These sets of reactions can be considered as flow networks, see [111, 97] among others for recent comprehensive and general introductions. Although there are few different ways in which these can be represented, a meaningful approach is to consider the relationship between reactions and metabolites as a bipartite graph where both reactions and metabolites are nodes, with reaction nodes only having edges to metabolite nodes, and vice versa. The edges are directed to represent either consumption or production of a metabolite, and numbered with the stoichiometry of the considered chemical reaction. Rates are associated with each reaction, and the reaction intrinsic rate multiplied by the reaction stoichiometry defines the rate at which a particular compound is produced. From this position, we can start introducing assumptions that will allow to make the analysis easier. The first and most common of these is the steady state assumption, which supposes that there is no net surplus or deficit of any metabolite, so that the total rate of production of a metabolite is equal to the total rate of consumption (for example we disregard circadian or other oscillations).

In some sense this assumption is equivalent to Kirchoff's current law in electric circuits, and means that the rates of the reactions producing the metabolite are tied to the rates of the reactions consuming it: the flux through each metabolite in the network must be constant, i.e. the input flux must equal the output flux.

Hence, under steady state condition, the time derivative in Eq. 4.1 can be relaxed to zero, obtaining a set of linear homogeneous equations

$$S \cdot v = 0, \tag{4.2}$$

one equation per metabolite, from which it is possible to calculate flux values. In this contest, v becomes a constant vector, independent by the time t and the metabolites concentrations x.

Since the number of metabolites usually is smaller than the number of reactions, the system resulting from the Eq. 4.2 is undetermined, i.e., there exists

different flux vectors satisfying it. Therefore, defining some constraints v_i^{min} and v_i^{max} for each flux $\{v_i\}_{i=1}^m$, and an objective function $f : \mathbb{R}^m \to \mathbb{R}$ to maximize/minimize we are able to shrink the feasible space defined by the Eq. 4.2, neglecting all the fluxes whose are not physiologically relevant. An objective function plays a role as a surrogate for the most plausible physiological state among the states of the system, and it might be defined, for instance, as the maximization of the biomass, or the minimization/maximization of the ATP (in case of reaction models).

FBA methods can be divided into two groups: 1) *biased methods* make the assumption that evolution has evolved to optimise for certain properties of the reaction system, such as maximising biomass production, and simulate this maximisation to find rates (i.e., they necessitate the definition of an objective function); 2) *unbiased methods* make no such assumptions, finding into the solution space a subset of statistically analysable functional states without requiring the definition of an objective function. See [160, 161] for in depth classification of these methods and list of useful software for the various types of analyses. Therefore, FBA's biased methods add one extra assumption to the model previously outlined: that evolution has already optimised the organism to grow as fast as possible. To use this assumption we add another placeholder reaction to the existing uptake and excretion nodes, a sink for biomass. This reaction simulates the sequestration of materials that is required for growth, and is generally a reaction with a very large number of input reactants, and unusually with fractional stoichiometries, since we can add the ratios of materials required for growth by simply analysing the constituents of a whole cell.

Once we have a biomass equation, we have a direction for optimisation. We find the assignment of fluxes that will achieve the highest biomass production. This is normally achieved by linear programming, which is a fast, specialised optimisation method that is applicable to this type of problems. Mathematically the FBA is translated as a Linear Programming Problem (LPP) as

follow:

$$max\ f(v) \text{ or } min\ f(v),$$

subject to

$$\begin{aligned} & S \cdot v = 0, \\ & v_i^{min} \leq v_i \leq v_i^{max}, \\ & v_i \in \mathbb{R}, \qquad \forall i = 1, \ldots,\ m, \end{aligned} \tag{4.3}$$

where v_i^{min} and v_i^{max} are either zero or negative and positive infinity, respectively, based on the direction of the flux. Usually considering internal fluxes v_i^{min} is set to 0 and v_i^{max} to a constant or positive infinity either. Differently, for the exchange fluxes depending on the existence of a source or a sink v_i^{min} is set to negative infinity or zero, and v_i^{max} is set to zero or positive infinity, respectively. If both a source and a sink are present for the metabolite then the exchange flux is defined bidirectional with v_i^{min} and v_i^{max} set to negative and positive infinity. Finally, the objective function is defined as a function $f : E \longrightarrow \mathbb{R}$ where E represents the flux vector satisfying (i) the mass balance equation, and (ii) the constraints (both defined in Eq.s 4.3).
Since LPPs are characterized by different issues to cope with, for instance the principal on is the existence of alternate optima, several techniques are proposed to guide the FBA through the identification of the best fluxes configurations. In this contest, two well known techniques in literature are the Flux Variability Analysis (FVA)[95] and the Reaction Essentiality Analysis (REA) [29], which can be exploited to identify the reactions whose play a central role in system and whose are most variable. In details, the FVA is exploited to determine robustness of metabolic models in various simulation conditions. It consists in identifying which fluxes are necessary to reach specific optima by maximizing and minimizing each flux in the network while maintaining the biomass flux at some fraction of the optimal value reached from the initial FBA (e.g., supporting 90% of maximal possible biomass production rate). Let us note that by using the FVA it is not possible to find the best optimum flux values, but rather it is possible to define a problem with the smallest set of variables, which are the reactions that contribute more to the variability of the objective function. Differently, REA consists to identify the essential reaction, which are reactions whose omission from the metabolic system is lethal, i.e., the maximum of the biomass flux goes to zero or lower than a specific threshold. So deleting one reaction per time and solving the corresponding

LPP, it is possible to find a list of all the essential reactions and to determine which reactions and metabolic pathways are active or dormant in a particular condition.

4.2 Dynamic Flux Balance

In order to combine extra-cellular dynamics and intra-cellular steady states, and therefore model metabolism under dynamic conditions, a step-wise FBA approach, commonly referred to as Dynamic Flux Balance Analysis DFBA, was proposed in [97, 96, 178]. This represents a first attempt to tackle limitations from both the mechanistic-based and constraint-based models, merging capabilities and exploiting knowledge of already well-known processes in order to decrease costs, improve efficiency and perform more descriptive phenotype predictions.
Roughly, the idea is that FBA simulation based on the steady-state constraint is run at each time step, and the uptake rates of given nutrients for the next simulation are reduced according to how much nutrient has been consumed in the previous FBA simulation. More specifically, the assumption is that the cell has a limited availability of extra-cellular nutrients (e.g., glucose), encoded as a constraint (maximum uptake rate) in the associated exchange reaction. Then, after an FBA simulation is run, the amount of nutrient which has been actually taken up by the organism is used to reduce the nutrient availability (maximum uptake rate) for the FBA simulation at the next time step. As a result, DFBA allows investigating genome-scale networks under transient conditions. It can be thought as a compromise between fully dynamic models, which cannot be simulated at large scale, and steady-state models, which do not involve kinetics [111, 97].
In literature we can find different formulations of the dynamic extension to FBA, in which stoichiometric information is joined with kinetic data. For instance in [96] the authors introduced two methods to formulate the DFBA, called Dynamic Optimization Approach (DOA) and Static Optimization Approach (SOA). The former involves optimization over the entire time period to obtain time profiles of fluxes and metabolite levels. In this case, the dynamic optimization problem is transformed to a non-linear programming problem, which is solved once. The latter method divides firstly the time period into several time intervals, then an optimization problem is solved at the beginning

of each time interval, followed by integration over the interval. The optimization problem was solved using LPP repeatedly during the course of the batch to obtain the flux distribution at a particular time instant.
Moreover, in [102] the authors propose an integrated DFBA strategy which is based on user-defined rules and parameters to represent an integrated stoichiometric reconstruction of biological processes such as signaling, metabolic and regulatory process. These are characterized by reactions with mixed time scales, for instance signaling reactions are generally much faster than regulatory reactions. Furthermore, it is possible to simulate quantitative and dynamic predictions of species concentrations by incorporating kinetic parameters. In details, the framework proposed involves performing an optimization, over multiple discretized time steps, to approximate the dynamics of a system given strictly stoichiometric constraints. These constraints are exploited to distinguish the fast and slow reactions, which can be not included in the network at specific time points. Similarly, in [37] the FBA approach is combined with regulatory Boolean logic, and ODEs to create an integrated model of E.coli to describe in detail carbohydrate uptake control and behavior of diauxic growth.

Part II

Theoretical results

Chapter 5

New PN formalisms for modeling complex systems

In this chapter we introduce the two new PN formalisms, that we developed to model and study complex systems. In particular we show how the well-know SPN and SSN formalisms can be extended to become a convenient tool to efficiently describe epidemiological and biological systems.

5.1 Extended Stochastic Petri Net

The Extended Stochastic Petri Net (ESPN) extends the SPN definition by splitting the set of transition T in two subsets T_{ma} and T_g. The former subset contains all transitions which fire with a rate expressed in MA law. The latter one includes all transitions whose random firing times have rates that are defined as general real functions. Hence, we will refer to the transitions belonging to T_{ma} as standard transitions and as general transitions those in T_g. This allows to easily model reactions which do not follow the MA law and to encode reactions whose dynamics are characterized through different and more complex functions (for instance Michaelis Menten and Hill kinetics). An example is showed in Fig. 5.1A), in which the ESPN version of the SIR model is depicted. We used a black boxes to highlight the transitions belonging to T_g,

i.e., the *Infection* transition, and the white boxes for the transitions belonging to T_{ma}, i.e., the *Recovery* transition.

Definition 8 (Extended Stochastic Petri Net)**.** *An Extended Stochastic Petri Net (ESPN) system is a tuple* $(P, T, I, O, m_0, \lambda, \Lambda)$*, where:*

- P, I, O, m_0 *are defined as in Definition 2.*
- $T = T_{ma} \cup T_g$ *is a finite, non empty set of transitions, with* $T_{ma} \cap T_g = \emptyset$*.* $T_{ma} = \{t_i^*\}_{1 \leq i \leq n_{T_{ma}}}$ *is the set of the* $n_{T_{ma}}$ *transitions whose speeds follow the MA law.* $T_g = \{\bar{t}_i\}_{1 \leq i \leq n_{T_g}}$ *is the set of the* n_{T_g} *transitions whose speeds are defined as continuous functions.*
- $\lambda : T_{ma} \to \mathbb{R}$ *gives the firing intensity of the transitions.*
- $\Lambda = \{f_1, \ldots, f_h\}$ *is the firing intensity set grouping the functions characterizing the general transitions in* T_g*, with* $h \leq n_{T_g}$*,* $h \in \mathbb{N}$*. In particular a function* f *depend only on the marking of the input places of the respective transition* t *(*$|{}^\bullet\mathbf{t}|$*) and on time* $\nu \in \mathbb{R}^+$*, i.e.,*

$$f : \mathbb{R}^{|{}^\bullet\mathbf{t}|} \times \mathbb{R}^+ \longrightarrow \mathbb{R}.$$

  For instance the function f_1 *might represent a Michaelis Menten kinetic and* f_2 *an Hill kinetic.*

We now show how the ODEs system describing the underlying deterministic process can be automatically derived from an ESPN model.

Similarly to what was introduced in Section 2.2, let $x_i(\nu) \in \mathbb{R}^+$ be the continuous approximation of the number of tokens in place p_i so that the vector $x(\nu) \in \mathbb{R}^{n_P}$ is the marking of the ESPN at time ν.
Let define $x(\nu)_{|{}^\bullet\mathbf{t}}$ as the subset of the marking $x(\nu)$ concerning just the input places to transition t. Thus, given a transition $t \in T = T_{ma} \cup T_g$ at the time ν, it will move tokens in state $x_i(\nu)$ with speed $F(t, \nu)$ defined as follows:

$$F(t, \nu, \hat{x}) := \begin{cases} \phi(t, \hat{x}), & t \in T_{ma}, \\ f_t(\nu, \hat{x}), & t \in T_g, \end{cases} \qquad \hat{x} = x(\nu)_{|{}^\bullet\mathbf{t}},\ f_t \in \Lambda$$

where $\phi(t_i, \hat{x}) = \lambda(t_i) \prod_k \hat{x}_k^{I(p_k, t_i)}$.

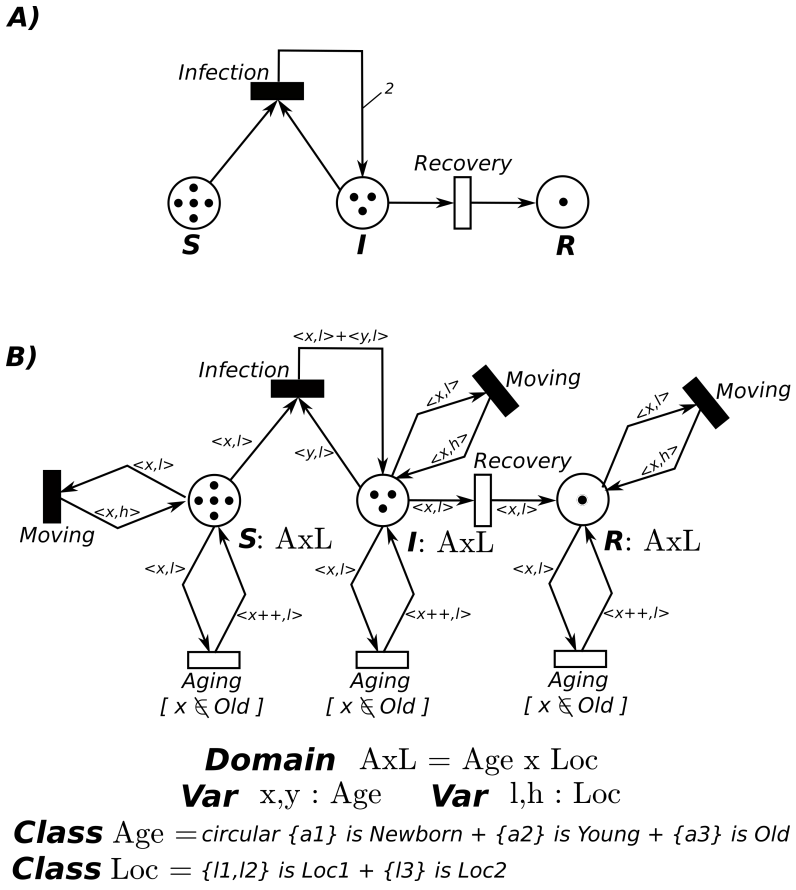


Figure 5.1: **A)** shows the ESPN of the SIR model, while **B)** the ESSN in which the age and locations of the population are taken in account.

The instantaneous changes of tokens $x_i(\nu)$ in the i-th place at time ν, is modeled by the following ODE:

$$\begin{aligned} dx_i(\nu) &= \sum_{j=1}^{n_T} F(t_j, \nu, \hat{x}) L[p_i, t_j] d\nu \\ &= \sum_{j=1}^{n_{T_{ma}}} \phi(t_j^*, \hat{x}) L[p_i, t_j^*] d\nu + \sum_{j=1}^{n_{T_g}} f_{\bar{t}_j}(\nu, \hat{x}) L[p_i, \bar{t}_j] d\nu \\ &\qquad \forall i \in \{1, \ldots, n_P\}, \end{aligned} \tag{5.1}$$

where L is the incidence matrix [142, 32] defined as $L[p_i, t_j] = I[p_i, t_j] - O[p_i, t_j]$. Therefore, if place p_i is an input (output) place of transition t_j, then transition t_j is removing (adding) tokens from (to) place p_i according to the current speed of the transition and the multiplicity given by function I (O).

Observe that the first sum of Eq. 5.1 corresponds exactly to Eq. 2.4, while the second sum accounts for the new timing specifications allowed by the ESPN formalism.

5.2 Extended Stochastic Symmetric Net

Similarly to Section 5.1, here we propose a new formalism, called Extended Stochastic Symmetric Net (ESSN), which extends the SSN exploiting the same ideas discussed in the previous section. In details, the set of transitions T is split in two subsets T_{ma} and T_g, basing on the syntax of the transitions rates. Therefore the Definition 8 is modified to consider all the features connected to the PN colours. An example is showed in Fig. 5.1B), in which the ESSN version of the SIR model presented in Section 2.4 is depicted.

Definition 9 (Extended Stochastic Symmetric Net)**.** *An ESSN is a ten-tuple:*

$$\mathcal{N}_{ESSN} = \langle P, T, \mathcal{C}, I, O, cd, \Theta, \omega, \Omega, m_0 \rangle$$

where

- $P, \mathcal{C}, I, O, cd, \Theta, m_0$ *are defined as in SSN (see definition 3).*
- T *is the set of transitions and is defined as* $T = T_{ma} \cup T_g$, *with* $T_{ma} \cap T_g = \emptyset$. *Where* $T_{ma} = \{t_i^*\}_{1 \leq i \leq n_{T_{ma}}}$ *is the set of the* $n_{T_{ma}}$ *transitions whose speeds follow the MA law, and* $T_g = \{\bar{t}_i\}_{1 \leq i \leq n_{T_g}}$ *is the set of the* n_{T_g} *transitions whose speeds are defined as continuous functions.*
- $\omega(t, c)$ *is the rate of transition* $t \in T_{ma}$ *assuming the firing of the instance* $\langle t, c \rangle$.
- $\Omega = \{f_{\langle t,c \rangle}\}_{t \in T \wedge c \in cd(t)}$ *is set grouping all the transition speeds* $\forall t \in T$. *Let us recall from definition 8 that these functions can depend only on the marking of the input places of transition* t *at time* ν. *In detail, with* $t \in T_{ma}$ *then* $f_{\langle t,c \rangle} = \varphi(\cdot, t, c)$, *where* φ *is defined in Eq. 2.6.*

Similarly to what discussed in Section 2.3, let $x_{p,c}(\nu) \in \mathbb{R}^+$ be the continuous approximation of the number of tokens in place p and color c so that the vector $x(\nu) \in \mathbb{R}^{n_P}$ is the marking of the ESPN at time ν.
Let define $\hat{x}(\nu) = x(\nu)_{|\bullet \mathbf{t}}$ as the subset of the marking $x(\nu)$ concerning only the input places to the transition t. Given $\langle t, c \rangle$ at the time ν, with transition

$t \in T = T_{ma} \cup T_g$, the firing of $\langle t, c\rangle$ will move tokens in state $x_{\langle p,c\rangle}(\nu)$ with speed $F(\hat{x}(\nu), t, c, \nu)$ defined as follows:

$$F(\hat{x}(\nu), t, c, \nu) := \begin{cases} \varphi(\hat{x}(\nu), t, c), & t \in T_{ma}, \\ f_{\langle t,c\rangle}(\hat{x}(\nu), \nu), & t \in T_g, \end{cases} \qquad f_{\langle t,c\rangle} \in \Omega(t, c) \tag{5.2}$$

where $\varphi(\hat{x}(\nu), t, c)$ is defined as in Eq. 2.8. Observe that $\varphi(\hat{x}(\nu), t, c)$ and $f_{\langle t,c\rangle}(\hat{x}(\nu), \nu)$ can depend only on the marking of the input places of transition t at time ν.

Finally the ODE characterizing the p and color tuple $c \in cd(p)$ is defined as:

$$\begin{aligned} \frac{dx_{p,c}(\nu)}{d\nu} &= \sum_{\langle \mathbf{t}', \mathbf{c}'\rangle \in E(\mathbf{t}', x(\nu))} F(\hat{x}(\nu), \mathbf{t}', \mathbf{c}', \nu)(L[p, \mathbf{t}'](\mathbf{c}')[c]) \\ &= \sum_{\substack{\langle \mathbf{t}', \mathbf{c}'\rangle \in E(\mathbf{t}', x(\nu)) \\ \wedge \mathbf{t}' \in T_{ma}}} \varphi(\hat{x}(\nu), \mathbf{t}', \mathbf{c}')(L[p, \mathbf{t}'](\mathbf{c}')[c]) \\ &+ \sum_{\substack{\langle \mathbf{t}', \mathbf{c}'\rangle \in E(\mathbf{t}', x(\nu)) \\ \wedge \mathbf{t}' \in T_g}} f_{\langle \mathbf{t}', \mathbf{c}'\rangle}(\hat{x}(\nu), \nu)(L[p, \mathbf{t}'](\mathbf{c}')[c]) \end{aligned} \tag{5.3}$$

where $\hat{x}(\nu) = x(\nu)_{|\bullet \mathbf{t}'}$.

5.3 Application example: Lotka-Volterra Model

In this section we show how the ESPN and ESSN formalisms can be efficiently used to analyse the Lotka-Volterra model, also known as the *predator-prey model* [93].

The Lotka-Volterra model is defined by a pair of ODEs, which describes the dynamics of biological systems characterized by two species that may interact, one as a predator and the other as prey. From the literature it is possible to find several definitions of this model [39, 35, 125], and in [72] a first classification of these models is reported depending on the *functional response*, i.e., the change in the rate of prey consumption by a predator when the prey density varied. In details, the functional response can be classified in three major types, namely Holling type I, II, III, whose definitions can be summarize as follows.

- **Holling type I**: the number of prey consumed shows a linear relationship with the prey density, the green line in Fig. 5.2.
- **Holling type II**: the gradient of the number of prey consumed decreases monotonically with increasing prey density, eventually saturating at a constant value of prey consumption, the orange line in Fig. 5.2. This follows from the assumption that the consumer is limited by its capacity to process food. A real example of this phenomena was published in [39] considering wolves and caribou. It was shown that the proportion of caribou killed per wolf decreases as caribou density increases, given that wolves are more easily satiated and the total number of caribou kills reaches a plateau. Mathematically, this model is equivalent to the model of enzime kinetics developed in 1913 by L. Michaelis and M. Menten [101].
- **Holling type III**: the gradient of the number of prey consumed first increases and then decreases with increasing prey density, the blue line in Fig. 5.2. This sigmoidal behaviour has been attributed to the existence of '*learning behaviour*' in the predator population, for example predators learning more specialised techniques for hunting or prey handling.

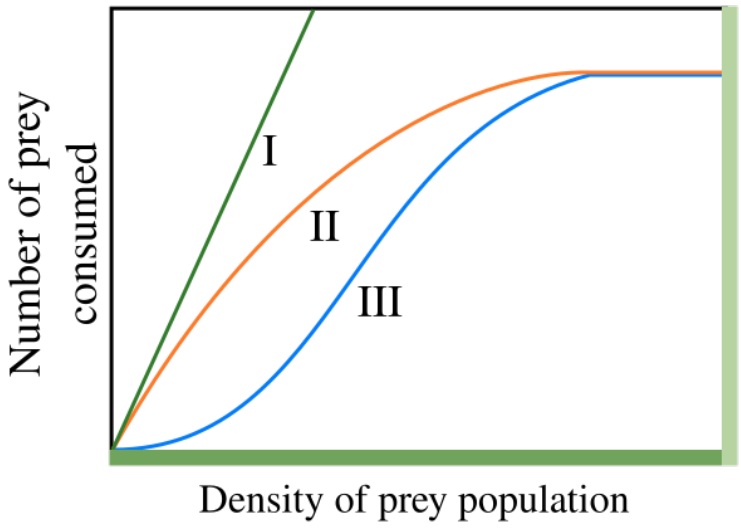


Figure 5.2: Holling types I, II, III functional responses.

Independently by the functional response exploited, a general version of the prey-predator model is defined by the following ODEs system.

$$\begin{aligned} \frac{dx_{Prey}}{d\nu} &= f(x_{Prey})x_{Prey} - g(x_{Prey}, x_{Predator})x_{Predator}, \\ \frac{dx_{Predator}}{d\nu} &= \epsilon g(x_{Prey}, x_{Predator})x_{Predator} - \gamma x_{Predator}, \end{aligned} \tag{5.4}$$

where:

1. x_{Prey} is the number of preys (e.g., caribou, rabbits, etc);
2. $x_{Predator}$ is the number of predators (e.g., wolves, foxes, etc);
3. ν represents time;
4. $f(x_{Prey})$ is the individual prey growth rate in the absence of predators;
5. $g(x_{Prey}, x_{Predator})$ is the functional response of the model;
6. ϵ is the efficiency of the predator in converting consumed prey into predator offspring;
7. γ is the predator mortality rate.

The simplest and more known prey-predator model [93] exploits a functional response of type I, in which a predator might interact with all the prey, thus the product of the two populations is the obvious outcome. This model can be easily represented using the SPN formalism, see Fig. 5.3a, from which the following ODEs system can be derived:

$$\begin{aligned} \frac{d\hat{x}_{Prey}}{d\nu} &= \alpha\hat{x}_{Prey} - \beta\hat{x}_{Prey}\hat{x}_{Predator}, \\ \frac{d\hat{x}_{Predator}}{d\nu} &= \delta\hat{x}_{Prey}\hat{x}_{Predator} - \gamma\hat{x}_{Predator}, \end{aligned} \tag{5.5}$$

where:

1. x_{Prey} is the average number of tokens in the *Prey* place, representing the preys;
2. $x_{Predator}$ is the average number of tokens in the *Predator* place, representing the predators;

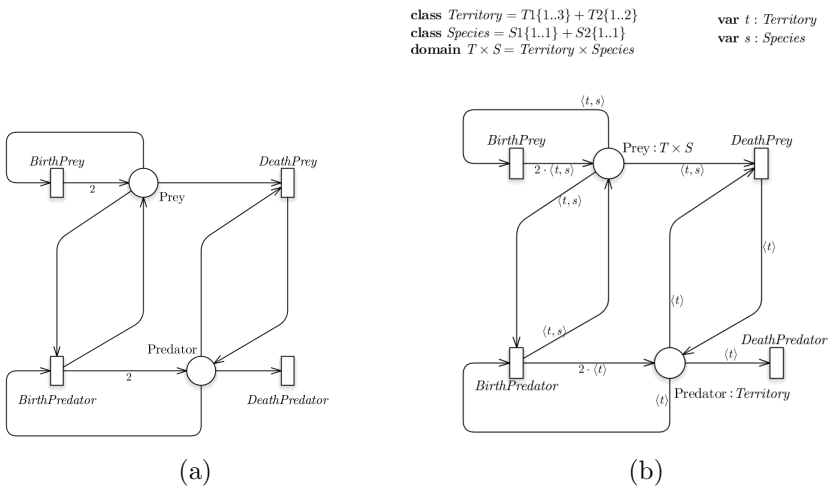


Figure 5.3: The Lotka-Volterra model represented exploiting the SPN formalism in (a) and the SSN formalism in (b), considering different territories and prey species.

3. α, β, γ, θ are positive real parameters describing the interaction of the two species and defining the rate of the *BirthPrey*, *DeathPrey*, *BirthPredator*, *DeathPrey* transitions, respectively.

Let use note that Eq.s 5.5 can be obtained from Eq.s 5.4 defining $g(x_{Prey}, x_{Predator}) = \beta x_{Prey}$, $f(x_{Prey}) = \alpha$, and $\delta = \beta\epsilon$.

In this model we are assuming that the prey have an unlimited food supply and it is able to reproduce exponentially (*BirthPrey* transition), unless subject to predation (*DeathPrey* transition). Differently, the food supply of the predator population depends entirely on the size of the prey population. Thus, the predator birth depends linearly on the number of prey at a specific time point (*BirthPredator* transition), while the predators death does not depend on it (*DeathPredator* transition). We can denote that all the transitions have a velocity rate defined according to MA law. Finally, the periodic dynamics of the system are showed in Fig. 5.4.

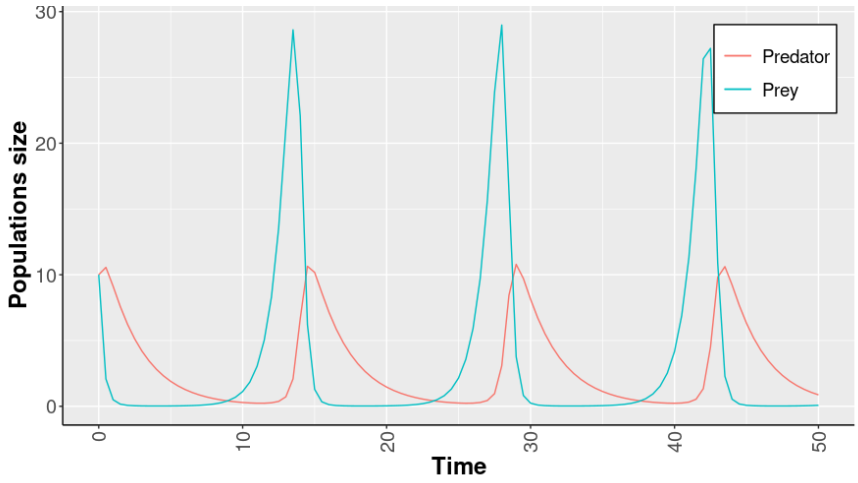


Figure 5.4: The Lotka-Volterra dynamics fixing: $\alpha = 1.1$, $\beta = 0.4$, $\gamma = 0.1$, $\theta = 0.4$, and the initial number of both the species equals to 10.

A similar example is depicted in Fig. 5.3b, in which the prey-predator model is represented by exploiting the SSN formalism in which different territories and prey species are modeled by using the color classes *Territory* and *Species* respectively. The *Territory* color class is defined by two sub-classes $T1$ and $T2$, representing for instance two different zones, characterized respectively by three and two colors (i.e., in the same zone we can have different positions). While *Species* models two different prey species defined by the two sub-classes $S1$ and $S2$, both characterized by one color. Since we are modeling that only the prey might belong to separate species, thus the color domain of the *Prey* place is defined by both the classes, while the color domain of the *Predator* place is characterized only by the *Territory* color class. Let us note that to have an easier comparison with the previous model and since we are not interested (in this case) to the model dynamics, we are not considering movements among the territories.
In this case we have to deal with 15 equations instead of two, modeling for all the five territories the interaction among two prey species and one of predators. Assuming that the interactions happen only if predator and prey are in the same territory, Eq.s 5.5 become

$$\begin{aligned}\frac{d\hat{x}_{Prey_{t,s}}}{d\nu} &= \alpha(t,s)\hat{x}_{Prey_{t,s}} - \beta(t,s)\hat{x}_{Prey_{t,s}}\hat{x}_{Predator_t},\\ \frac{d\hat{x}_{Predator_t}}{d\nu} &= \delta(t,s)\hat{x}_{Prey_{t,s}}\hat{x}_{Predator_t} - \gamma(t)\hat{x}_{Predator_t},\end{aligned} \tag{5.6}$$

where s, t represent the prey specie and the territory, respectively. Thus, $\hat{x}_{Prey_{t,s}}$ is the number of preys in the territory t and belonging to the s specie. For completeness all the parameters depend on the $t,\ s$ colors.

However, most ecological interest in functional responses has to involve types II and III. For instance, if it is considered that a single predator can feed only until the stomach is full, a saturation function is needed to indicate the intake of food, which is modeled using the Holling type II term. A simple example of this term is expressed by Eq. 5.7, where a is the attack rate at which the consumer encounters food items per unit of food density, and h is the average handling time spent on processing a food item. Indeed, more complex examples are given in [35, 125].

$$g_{II}(x_{Prey}, x_{Predator}) = \frac{a\ x_{Prey}}{1 + a\ h\ x_{Prey}}. \tag{5.7}$$

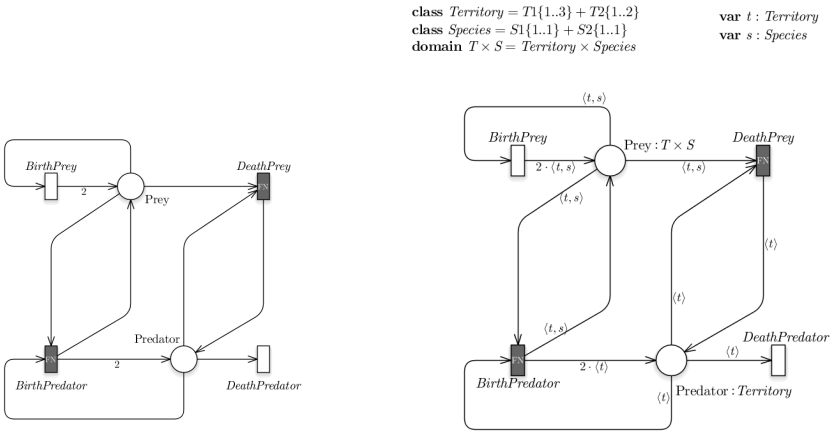
Similarly, type III functional responses can be characterized by the Eq. 5.7 if the attack constant rate a is defined in function of the number of preys [74], for instance a general form is given by a hyperbolic function of x_{Prey}:

$$a = \frac{d + b\ x_{Prey}}{1 + c\ x_{Prey}} \tag{5.8}$$

in which $b,\ c,\ d$ are constants. Thus, we can easily derive a general equation of type III as follows:

$$g_{III}(x_{Prey}, x_{Predator}) = \frac{d\ x_{Prey} + b\ x_{Prey}^2}{1 + c\ x_{Prey} + d\ h\ x_{Prey} + b\ h\ x_{Prey}^2}. \tag{5.9}$$

Finally, considering the functional response types described in Eq.s 5.7 and 5.9, in terms of SPN they should define the rate of the *DeathPrey* and *BirthPredator* transitions. Indeed this is not easy using the SPN (or SSN) formalism, while it can be achieved easily by exploiting the ESPN (or ESSN) formalism. Indeed in the extended formalisms it is possible to integrate in the model more complex functional response than the type I by defining the



(a) (b)

Figure 5.5: The Lotka-Volterra model represented exploiting the ESPN formalism in (a) and the ESSN formalism in (b).

DeathPrey (and *BirthPredator*) transition as a general transition. Let us recall from Section 5.1 that a general transition $t \in T_g$ is defined by a function $f_t(\hat{x}(\nu), \nu)$, where $\hat{x}(\nu)$ represents the vector of the average number of tokens for all the transition input places at time ν. Hence, the general transition velocities of Fig. 5.5a should be defined as follows

$$f_{DeathPrey}(\hat{x}(\nu), \nu) = g(x_{Prey}, x_{Predator})\ x_{Predator} \tag{5.10}$$
$$f_{BirthPredator}(\hat{x}(\nu), \nu) = \epsilon g(x_{Prey}, x_{Predator})\ x_{Predator}$$

with $g()$ equals to $g_{II}()$ from Eq. 5.7 or $g_{III}()$ from Eq. 5.9 in order to use a functional response of type II or III, respectively. Let us note that in the ESSN case the Eq.s 5.10 can be easily rewritten considering the color instances given by the the prey species and the territories. Both the ESPN and ESSN models are showed in Fig.s 5.5a and 5.5b, in which we used black boxes to highlight the general transitions.

Chapter 6

Extended solution techniques

In this chapter we describe three new techniques developed to make more efficiently the analysis of complex ESPN and ESSN models. In particular, Sec. 6.1 focuses on exploiting the model symmetries to reduce the solution computational cost. In this contest, we developed an algorithm to improve the previously introduced procedure in Section 2.4 to automatically derive the reduced ODEs system from the (E)SSN model exploiting the systems symmetries, by avoiding the complete net unfolding (from which the same measures derived by the original ODEs can be computed). In Section 6.2 we describe how ESPN formalism can be exploited as a "*metaformalims*" to create hybrid model in which some sub-components were detailed as a mechanism model and other as a constraint model. This means that some components are modeled with a high level of precision exploiting an ODEs system (mechanism-based model), and other by a low level of precision given by the FBA (constraint-based models). Finally, in Sec. 6.3 we describe an adaptation of the τ-*leaping algorithm* [58], to be used on general models (i.e., not only for chemical reactions models characterized by MA reactions).

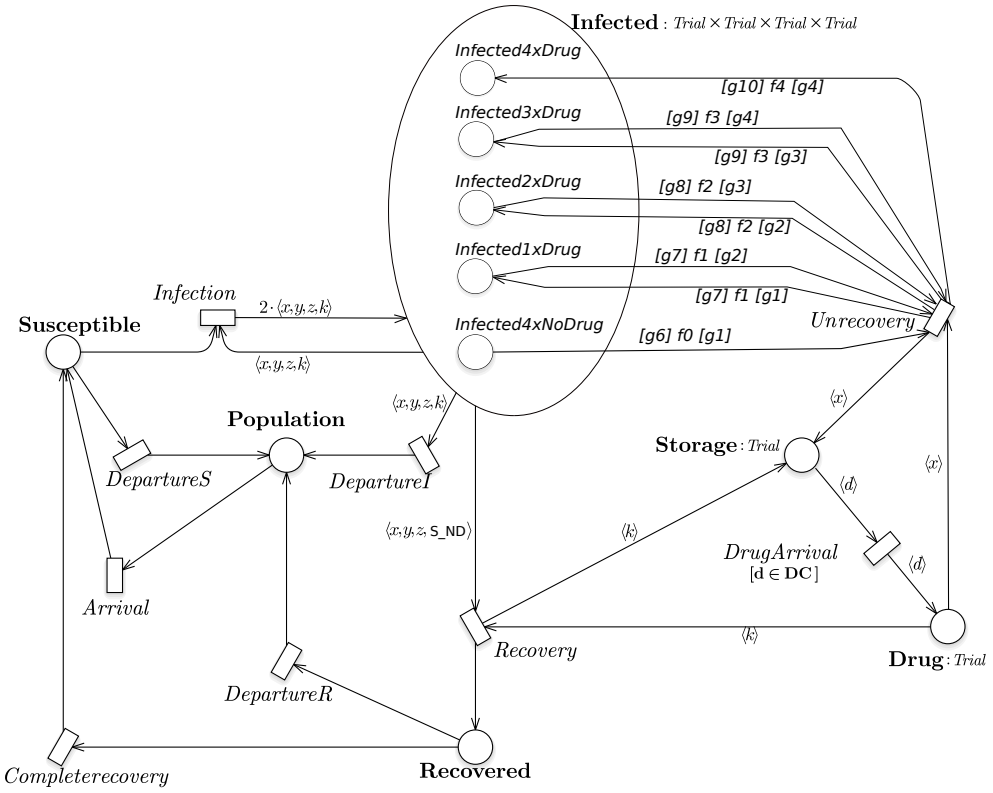
6.1 Symbolic formalism without complete unfolding

In Section 2.3 we explained how to derive from an SSN model the corresponding ODEs system, from which the system behaviour can be generated. In particular we highlighted that the unfolding step is typically the bottleneck of this task. Moreover in Section 2.4 we showed that this step is also needed when a reduced ODEs system is generated exploiting the symmetries. Since this can substantially affect the efficacy of such approach, during my PhD we developed a new method in which the reduced ODEs system is directly obtained from a partially unfolded SSN model through an algebraic manipulation of its color notation.

This section is then organized as follows: in Sec. 6.1.1 is briefly introduced an example which will be used through the algorithm explanation. Successively, we explain the two steps characterizing the new method to derive the reduced ODE system starting from an SSN partially unfolded without completely unfolding the model: 1) in Sec. 6.1.2 the partial unfolding procedure, and 2) in Sec. 6.1.3 the specific functions required to automatically generate a reduced set of ODE that represent the whole behavior of the system. In Sec. 6.1.4 the benefits of the method and its implementation are discussed in details. Finally, in Sec. 6.1.5 we discuss which constrains have to be defined on the functions characterizing the general transitions to extend this approach for the ESSN formalism.

6.1.1 Case study

In Fig. 6.1 the SSN model used throughout this section is depicted: it represents a Susceptible-Infected-Recovered-Susceptible (SIRS) model where members of a population (place *Population*) may join (transition *Arrival*) a community of susceptible individuals (place *Susceptible*), which can be infected by meeting (transition *Infection*) infected community members (place *Infected*). Place *Infected* is represented as a big ellipse containing five places: this will be used later to illustrate the partial unfolding procedure. For the moment let us just consider the big place as a whole, disregarding the enclosed places and the annotations on the arcs incident on them described in the table 6.1.



Color class definition: Trial = ND ∪ DC, |ND| = 1, |DC| = 5

Figure 6.1: A SIRS model with antibiotic resistance.

Infected individuals can be treated with a drug (e.g., an antibiotic is prescribed, possibly after an antibiogram analysis); the treatment may not be effective for various reasons, e.g., the antibiotic did not reach the infected tissue with enough concentration, or the patient did not accurately follow the prescription. Bacteria may develop resistance to an antibiotic as a result of an unsuccessful treatment, and this is modeled by recording in the token colors of place *Infected* the types of drugs (up to four) that did not lead to a full recovery. Transition *Unrecovery* represents the occurrence of unsuccessful treatment leading to a higher degree of resistance of the involved bacterium, up to a maximum level which cannot be treated with any drug. In the low and intermediate resistance levels a successful treatment may occur, represented by transition *Recovery*, leading to a complete recovery (passing through place *Recovered* and then back to the *Susceptible* state). Place *Drug* represents the available drugs that can be used for treatment. Periodically, a new supply of drugs is brought to the pharmacy from a *Storage* (transition *DrugArrival*).

Trans	guard	rate
Arrival	*true*	6e−8
CompleteRecovery	*true*	0.025
DepartureR	*true*	3.9e−8
DepartureS	*true*	3.9e−8
Infection	$g_1 \vee g_2 \vee g_3 \vee g_4 \vee g_5$	4.89e−7
DepartureI	$g_1 \vee g_2 \vee g_3 \vee g_4 \vee g_5$	3.9e−8
DrugArrival	$[d \in \mathrm{DC}]$	1e−3
Unrecovery	$g_1 \vee g_2 \vee g_3 \vee g_4$	2e−5
Recovery	$g'_1 \vee g'_2 \vee g'_3 \vee g'_4$	1.98e−3
$g_6 = d_1 \in \mathrm{ND} \wedge d_2 \in \mathrm{ND} \wedge d_3 \in \mathrm{ND} \wedge d_4 \in \mathrm{ND}$ $f_0 = \langle S_{\mathrm{ND}}, S_{\mathrm{ND}}, S_{\mathrm{ND}}, S_{\mathrm{ND}} \rangle$ $g_1 = x \in \mathrm{DC} \wedge y \in \mathrm{ND} \wedge z \in \mathrm{ND} \wedge k \in \mathrm{ND}$ $g_7 = d_1 \in \mathrm{DC} \wedge d_2 \in \mathrm{ND} \wedge d_3 \in \mathrm{ND} \wedge d_4 \in \mathrm{ND}$ $f_1 = \langle x, S_{\mathrm{ND}}, S_{\mathrm{ND}}, S_{\mathrm{ND}} \rangle$ $g_2 = x \in \mathrm{DC} \wedge y \in \mathrm{DC} \wedge z \in \mathrm{ND} \wedge k \in \mathrm{ND} \wedge x \neq y$ $g_8 = d_1 \in \mathrm{DC} \wedge d_2 \in \mathrm{DC} \wedge d_3 \in \mathrm{ND} \wedge d_4 \in \mathrm{ND}$ $f_2 = \langle x, y, S_{\mathrm{ND}}, S_{\mathrm{ND}} \rangle$ $g_3 = x \in \mathrm{DC} \wedge y \in \mathrm{DC} \wedge z \in \mathrm{DC} \wedge k \in \mathrm{ND} \wedge$ $\wedge x \neq y \wedge x \neq z \wedge y \neq z$ $g_9 = d_1 \in \mathrm{DC} \wedge d_2 \in \mathrm{DC} \wedge d_3 \in \mathrm{DC} \wedge d_4 \in \mathrm{ND}$ $f_3 = \langle x, y, z, S_{\mathrm{ND}} \rangle$ $g_4 = x \in \mathrm{DC} \wedge y \in \mathrm{DC} \wedge z \in \mathrm{DC} \wedge k \in \mathrm{DC} \wedge$ $\wedge x \neq y \wedge x \neq z \wedge x \neq k \wedge y \neq z \wedge y \neq k \wedge z \neq k$ $g_{10} = d_1 \in \mathrm{DC} \wedge d_2 \in \mathrm{DC} \wedge d_3 \in \mathrm{DC} \wedge d_4 \in \mathrm{DC}$ $f_4 = \langle x, y, z, k \rangle$ $g_5 = x \in \mathrm{ND} \wedge y \in \mathrm{ND} \wedge z \in \mathrm{ND} \wedge k \in \mathrm{ND}$		
$g'_1 = x \in \mathrm{ND} \wedge y \in \mathrm{ND} \wedge z \in \mathrm{ND} \wedge k \in \mathrm{DC}$ $g'_2 = x \in \mathrm{DC} \wedge y \in \mathrm{ND} \wedge z \in \mathrm{ND} \wedge k \in \mathrm{DC} \wedge x \neq y$ $g'_3 = x \in \mathrm{DC} \wedge y \in \mathrm{DC} \wedge z \in \mathrm{ND} \wedge k \in \mathrm{DC} \wedge$ $\wedge x \neq y \wedge x \neq k \wedge y \neq k$ $g'_4 = x \in \mathrm{DC} \wedge y \in \mathrm{DC} \wedge z \in \mathrm{DC} \wedge k \in \mathrm{DC} \wedge$ $\wedge x \neq y \wedge x \neq z \wedge y \neq z \wedge x \neq k \wedge y \neq k \wedge z \neq k$		

Table 6.1: Parameters and transition definition of the SSN model depicted in Fig.6.1.

Thus SSN model is characterized by a single basic color class, $Trial$, composed of two *static subclasses*: ND (= $\{nd\}$), and DC, of cardinality $n \geq 4$ (in the experiments $n = 5$). Thus, places *Drug* and *Storage* have color domain $Trial$, and actually hold only tokens with colour in subclass DC. Instead place *Infected* has color domain $Trial \times Trial \times Trial \times Trial$, and thus contains 4-tuples of elements of class $Trial$ representing the achieved antibiotic resistance of the bacterium of an infected individual. The other places have a "neutral" color domain (i.e., they contain "black" tokens) composed, by convention, of the neutral color class $E = \{e\}$. The initial marking we are considering has 890000 black tokens in place *Susceptible*, 50 tokens

with color $\langle nd, nd, nd, nd\rangle$ in place *Infected*, 100 tokens of color $\langle drug_i\rangle$ for all $drug_i \in$ DC, 9950 black tokens in place *Population*. The arc expressions are rather simple: syntactically they appear as tuples (1-tuple $\langle x\rangle$ and 4-tuples $\langle x, y, z, k\rangle$, or $\langle x, S_{\mathrm{ND}}, S_{\mathrm{ND}}, S_{\mathrm{ND}}\rangle$) whose elements are projections (taking the form of variables as x, y, z, k, d) or constant functions (e.g., S_{ND}).

Let us consider the arcs connecting transition *Unrecovery* to the places enclosed by *Infected*, which result from a partial unfolding procedure that will be explained later: in the original net there are only one input and one output arc, annotated by $I[Infected, Unrecovery] = f_0[g_1] + f_1[g_2] + f_2[g_3] + f_3[g_4]$ and $O[Infected, Unrecovery] = f_1[g_1] + f_2[g_2] + f_3[g_3] + f_4[g_4]$, respectively, where the definition of f_i and g_i are given in Fig. 6.1. A term like $f_i[g_j]$ evaluates to $f_i(c)$ if $g_i(c) = true$, otherwise it results in $\emptyset$.

The transition color domains are $Trial$ (*DrugArrival*, with parameter d); $Trial \times Trial \times Trial \times Trial$ (*Unrecovery*, *Infection*, *Recovery*, *DepartureI*); E (*Arrival*, *DepartureS*, *DepartureR*, *Completerecovery*).

Each transition has an associated *guard*, defining its valid instances, and a *rate* parameter ($w(trans.instance)$) that concurs to the definition of the transition firing speed (also called *transition intensity*): the transition guards and rates for this model are listed on the table 6.1. Transition intensities (characterizing the stochastic delay between transition enabling and firing) are derived from the transition rate and the marking according to the *mass action law* (MA), when $t \in T_{ma}$, or to a continuous function if $t \in T_g$. For simplicity in this section we shall consider the SSN formalism, thus only transitions following the MA law.[1] Therefore, when the MA semantics is used, the transition intensity is given by the product of $w(t, c)$ (the base rate) and $\prod_{p \in {}^\bullet t} \prod_{c' \in I[p,t](c)} m(p)[c]^{I[p,t](c)[c']}$ (the product of all the transition input places, each powered to the cardinality of the transition input arc, respectively).

The table 6.2 refers to places $Infected$3x$Drug$ and *Susceptible*, and contains the expressions of relations $\mathcal{A}$ and $\mathcal{R}$ (see Section 2.4.1) for all transitions they are connected to. These symbolic expressions are the basis for the construction of the ODE associated with place $Infected$3x$Drug$, which contains positive terms corresponding to $\mathcal{A}$, and negative terms corresponding to $\mathcal{R}$,

[1] In Section 6.1.5 we will discuss how this can be extended for ESSN, i.e., which constraints must be added for the continuous function defining the rate of the general transitions.

representing the flow of tokens into and out of the place, respectively. Analogously for place *Susceptible*.

$\mathcal{A}(Infected3xDrug, Infection) =$ $1\langle d_1, d_2, d_3, d_4\rangle[d_1 \neq d_2 \wedge d_1 \neq d_3 \wedge d_2 \neq d_3 \wedge$ $d_1 \in \mathrm{DC} \wedge d_2 \in \mathrm{DC} \wedge d_3 \in \mathrm{DC} \wedge d_4 \in \mathrm{ND}]$
$\mathcal{R}(Infected3xDrug, Recovery) =$ $1\langle d_1, d_2, d_3, S - d_1 \cap S - d_2 \cap S - d_3 \cap S_{\mathrm{DC}}\rangle$ $[d_1 \neq d_2 \wedge d_1 \neq d_3 \wedge d_2 \neq d_3 \wedge d_1 \in \mathrm{DC} \wedge$ $d_2 \in \mathrm{DC} \wedge d_3 \in \mathrm{DC} \wedge d_4 \in \mathrm{ND}]$
$\mathcal{A}(Infected3xDrug, Unrecovery) =$ $1\langle d_1, d_2, d_3, S_{\mathrm{ND}}\rangle[d_1 \neq d_2 \wedge d_1 \neq d_3 \wedge d_2 \neq d_3 \wedge$ $d_1 \in \mathrm{DC} \wedge d_2 \in \mathrm{DC} \wedge d_3 \in \mathrm{DC} \wedge d_4 \in \mathrm{ND}]$
$\mathcal{R}(Infected3xDrug, Unrecovery) =$ $1\langle d_1, d_2, d_3, S - d_1 \cap S - d_2 \cap S - d_3 \cap S_{\mathrm{DC}}\rangle$ $[d_1 \neq d_2 \wedge d_1 \neq d_3 \wedge d_2 \neq d_3 \wedge d_1 \in \mathrm{DC} \wedge$ $d_2 \in \mathrm{DC} \wedge d_3 \in \mathrm{DC} \wedge d_4 \in \mathrm{ND}]$
$\mathcal{R}(Infected3xDrug, DepartureI) =$ $1\langle d_1, d_2, d_3, d_4\rangle[d_1 \neq d_2 \wedge d_1 \neq d_3 \wedge d_2 \neq d_3 \wedge$ $d_1 \in \mathrm{DC} \wedge d_2 \in \mathrm{DC} \wedge d_3 \in \mathrm{DC} \wedge d_4 \in \mathrm{ND}]$
$\mathcal{R}(Susceptible, Infection) =$ $1\langle S_{\mathrm{ND}}, S_{\mathrm{ND}}, S_{\mathrm{ND}}, S_{\mathrm{ND}}\rangle + 1\langle S_{\mathrm{DC}}, S_{\mathrm{ND}}, S_{\mathrm{ND}}, S_{\mathrm{ND}}\rangle$ $+1[d_1 \neq d_2]\langle S_{\mathrm{DC}}, S_{\mathrm{DC}}, S_{\mathrm{ND}}, S_{\mathrm{ND}}\rangle$ $+1[d_1 \neq d_2 \wedge d_1 \neq d_3 \wedge d_2 \neq d_3]\langle S_{\mathrm{DC}}, S_{\mathrm{DC}}, S_{\mathrm{DC}}, S_{\mathrm{ND}}\rangle$ $+1[d_1 \neq d_2 \wedge d_1 \neq d_3 \wedge d_1 \neq d_4 \wedge d_2 \neq d_3 \wedge d_2 \neq d_4 \wedge d_3 \neq d_4]$ $\langle S_{\mathrm{DC}}, S_{\mathrm{DC}}, S_{\mathrm{DC}}, S_{\mathrm{DC}}\rangle$
$\mathcal{A}(Susceptible, Arrival) = 1\langle S_E\rangle$
$\mathcal{A}(Susceptible, CompleteRecovery) = 1\langle S_E\rangle$
$\mathcal{R}(Susceptible, DepartureS) = 1\langle S_E\rangle$

Table 6.2: The expressions of relations $\mathcal{A}$ and $\mathcal{R}$.

Place $Infected3xDrug$ comes from the partial unfolding of $Infected$: this replica holds four-tuples of colors whose first three are of type DC and all different, and the fourth one is of type ND (the singleton $\{nd\}$). This is clearly indicated by the guards of all corresponding $\mathcal{A}$ and $\mathcal{R}$ terms. The symbolic expression of $\mathcal{A}(Infected3xDrug, Infection)$ is a template of those instances of $Infection$ putting tokens of color $\langle d_1, d_2, d_3, d_4\rangle$ into the place[2]: this simple

[2]Recall that $cd(Infected) = Trial \times Trial \times Trial \times Trial$, hence symbol d_i denotes the value of the i^{th} element in the four-tuple associated with each token on the place.

template indicates that (only) a transition instance $x = d_1$, $y = d_2$, $z = d_3$, $k = d_4$ puts exactly *one* (according to the weight of the unique term of $\mathcal{A}$) such a token into the place. Observe that despite there is input arc from $Infected$ to $Infection$ (which after the partial unfolding is replicated for each of the five unfolded places, including $Infected3xDrug$), $\mathcal{R}(Infected, Infection) = \emptyset$.

The expression of $\mathcal{R}(Infected3xDrug, Recovery)$ is a bit more complex: it represents those instances of $Recovery$ withdrawing tokens of color $\langle d_1, d_2, d_3, d_4 \rangle$ from $Infected3xDrug$. The expression's unique term means that an instance $x = d_1$, $y = d_2$, $z = d_3$, and with k bound to *any* color (of type DC) other than d_1, d_2, d_3, removes exactly *one* such token from the place. According to the term cardinality, there are $|\mathrm{DC}| - 3$ such instances.

The symbolic expression of $\mathcal{R}(Susceptible, Infection)$ is built of five terms. As a whole, it represents which instances of $Infection$ may remove a *black token* from place $Susceptible$: they are 1) those with $x \in$ ND, $y \in$ ND, $w \in$ ND, $k \in$ ND, which corresponds to $x = y = z = k = nd$, 2) those with $\forall x \in$ DC, $y \in$ ND, $w \in$ ND, $k \in$ ND (there are $|\mathrm{DC}|$ such instances), 3) those with $\forall x \in$ DC, $\forall y \in$ DC, $x \neq y$, $w \in$ ND, $k \in$ ND (there are $|\mathrm{DC}|(|\mathrm{DC}| - 1)$ such instances), and so forth. In this case *filters* appear in the expression, allowing specific color instances to be selected from the (parametric) set represented by the following tuple (e.g., those satisfying $x \neq y$ in the third term): as said, a variable symbol d_i in a filter explicitly refers the i^{th} element of a given type (in our example, coinciding with the i^{th} position) in the tuple.
Let us show how the ODE looks like for two places in the SIRS model, place $Susceptible$ and place $Infected3xDrug$, for which the $\mathcal{A}()$ ad $\mathcal{R}()$ have been presented earlier in this section. We recall that the function $\varphi(\cdot)$ represents the transition instance intensity defined following the MA law (Eq. 2.6 in Section 2.3).

$$\begin{aligned}\frac{dx[Susceptible, e]}{d\nu} &= \varphi(x, Arrival, e) \cdot 1 - \varphi(x, Departure, e) \cdot 1 \\ &+ \varphi(x, CompleteRecovery, e) \cdot 1 \\ &- \sum_{\langle d_1,d_2,d_3,d_4\rangle \in D_0} \varphi(x, Infection, \langle d_1, d_2, d_3, d_4\rangle) \cdot 1 \\ &- \sum_{\langle d_1,d_2,d_3,d_4\rangle \in D_1} \varphi(x, Infection, \langle d_1, d_2, d_3, d_4\rangle) \cdot 1 \\ &- \sum_{\langle d_1,d_2,d_3,d_4\rangle \in D_2} \varphi(x, Infection, \langle d_1, d_2, d_3, d_4\rangle) \cdot 1 \\ &- \sum_{\langle d_1,d_2,d_3,d_4\rangle \in D_3} \varphi(x, Infection, \langle d_1, d_2, d_3, d_4\rangle) \cdot 1 \\ &- \sum_{\langle d_1,d_2,d_3,d_4\rangle \in D_4} \varphi(x, Infection, \langle d_1, d_2, d_3, d_4\rangle) \cdot 1\end{aligned} \tag{6.1}$$

where D_i and g_i are domains and filters respectively, which are defined as follows:

$$\begin{aligned}D_0 &= \mathrm{ND} \times \mathrm{ND} \times \mathrm{ND} \times \mathrm{ND} \\ D_1 &= \mathrm{DC} \times \mathrm{ND} \times \mathrm{ND} \times \mathrm{ND} \\ g_2 &= d_1 \neq d2 \\ D_2 &= \mathrm{DC} \times \mathrm{DC} \times \mathrm{ND} \times \mathrm{ND} \\ g_3 &= d_1 \neq d_2 \wedge d_1 \neq d_3 \wedge d_2 \neq d_3 \\ D_3 &= \mathrm{DC} \times \mathrm{DC} \times \mathrm{DC} \times \mathrm{ND} \\ g_4 &= d_1 \neq d_2 \wedge d_1 \neq d_3 \wedge d_1 \neq d_4 \wedge d_2 \neq d_3 \wedge d_2 \neq d_4 \wedge d_3 \neq d_4 \\ D_4 &= \mathrm{DC} \times \mathrm{DC} \times \mathrm{DC} \times \mathrm{DC}\end{aligned}$$

and a filter prefixing a domain restricts it to the tuples satisfying the predicate.

The first term derives from $\mathcal{A}(Susceptible, Arrival)$, and stands for a summation with only one element: in fact the neutral class E is a singleton $\{e\}$, and the neutral transition *Arrival* has only one instance with intensity $\varphi(x, Arrival, e) = \omega(Arrival)\ x[Population]$ and the coefficient 1 is that appearing in $\mathcal{A}(Susceptible, Arrival)$. Similar arguments justify the second and third terms (the third term is subtracted because it derives from $\mathcal{R}(Susceptible, Departure)$). The following five terms instead refer to several instances of *Infection* that withdraw tokens from different unfolded places

replacing the original place Infected. The first term for transition *Infection* comprises only one instance (since $|D_0| = 1$) and the intensity of such instance is

$$\varphi(x, Infection, \langle d_1, d_2, d_3, d_4 \rangle) = \omega(Infection) x[Susceptible] x[Infected4\text{x}NoDrug]$$

(observe that the tokens in place $Infected4\text{x}NoDrug$ have all color $\langle nd, nd, nd, nd \rangle$). The second term for transition *Infection* comprises |DC| instances (since $|D_1| = |\text{DC}|$) and the intensity of such instances is

$$\varphi(x, Infection, \langle d_1, d_2, d_3, d_4 \rangle) = \omega(Infection) x[Susceptible] x[Infected1\text{x}Drug].$$

Assuming $n = |\text{DC}| \geq 4$, the next three terms comprise respectively $n \cdot (n-1), n \cdot (n-1) \cdot (n-2)$ and $n \cdot (n-1) \cdot (n-2) \cdot (n-3)$ instances and their intensities depend on the (average) marking of any color 10 in place $Infected2\text{x}Drug$, $Infected3\text{x}Drug$ and $Infected4\text{x}Drug$, respectively.
The ODE for place $Infected3\text{x}Drug$ is derived similarly:

$$\begin{aligned} \frac{dx[Infected3\text{x}Drug, d_{D_3}]}{d\nu} &= +\omega_1 \cdot x[Susceptible] x[Infected3\text{x}Drug, d_{D_3}] \cdot 1 \\ &\quad - \omega_2 \cdot x[Infected3\text{x}Drug, d_{D_3}] \cdot 1 \\ &\quad + \omega_3 \cdot x[Infected2\text{x}Drug, d_{D_3}] x[Drug] \cdot 1 \\ &\quad - \sum_{\langle d_1, d_2, d_3, d_4 \rangle \in [g_3']D_3} \omega_3 \cdot x[Infected3\text{x}Drug, d_{D_3}] x[Drug] \cdot 1 \\ &\quad - \sum_{\langle d_1, d_2, d_3, d_4 \rangle \in [g_3']D_3} \omega_4 \cdot x[Infected3\text{x}Drug, d_{D_3}] x[Drug] \cdot 1 \end{aligned} \tag{6.2}$$

where:

$$\omega_1 = \omega(Infection),$$
$$\omega_2 = \omega(DepartureI),$$
$$\omega_3 = \omega(Unrecovery),$$
$$\omega_4 = \omega(Recovery);$$

and $d_{D_3} = \langle a, b, c, nd \rangle$ represents an arbitrary fixed tuple in $[g_3]D_3$ (where a, b, c are arbitrary distinct colors in DC representing the drugs that failed to fight the infection while the last element is $nd \in \text{ND}$) and $g_3' = g_3 \wedge d_1 = a \wedge d_2 = b \wedge d_3 = c$. The last two summations range over $|\text{DC}| - 3$ instances: those with $d_1 = a \wedge d_2 = b \wedge d_3 = c$ and $d_4 \in \text{DC}$ and different from a, b, c.

6.1.2 First step: partial unfolding

The first step of this new method to compute the set of $\widehat{ODE}$s is a preliminary, partial unfolding of some portions of the original SSN, since it is necessary that all instances of any transition have the same (base) rate and the (average) number of tokens of each color c in place p at any time ν is the same. In details, transitions are partially unfolded **only if** their rates are color dependent: assuming that the rate of t, with an associated guard $g(t)$, is expressed as a set of k pairs $\{\langle g_i, w_i\rangle\}$ (with $w_i \in \mathbb{R}^+$, $j \neq i \Rightarrow \neg(g_i \wedge g_j)$, and $\bigvee_i g_i \equiv g(t)$), after the unfolding there will be k copies of t, denoted $t_{[g_i]}$, with same input and output arcs as t, with guard g_i, and rate w_i. Instead, places are partially unfolded in two steps: the first step is based on the static partition of color classes[3], while the second step involves those places that have the same static subclass repeated in their color domain, after the first step.

Let $\widetilde{C_i} = \{C_{i,h}, h : 1 \ldots |\widetilde{C_i}|\}$ be the set of static subclasses of color class C_i. The number of places resulting from the complete unfolding of a place p with color domain $cd(p) = C_{i_1}^{n_1} \times \ldots \times C_{i_k}^{n_k}$ is given by the product $\prod_{j=1}^{k}(\sum_{h=1}^{|\widetilde{C_{i_j}}|} |C_{i_j,h}|)^{n_j}$, see Section 2.4. Where the notation $C_{i_j}^{n_j}$ means that class C_{i_j} occurs n_j times in the Cartesian product. Differently, considering the partial unfolding of the same place p, the resulting number of places from the first step is given by the product $\prod_{j=1}^{k} |\widetilde{C_{i_j}}|^{n_j}$. Moreover, place unfolding exploits the possibility of prefixing the functions on the arcs incident on an unfolded place instance with *filters* ensuring that the tokens yielded by evaluating the functions match the color pattern associated with that instance. In such manner each partially unfolded place is labelled with $p_{[g]}$, where g is a conjunction of membership clauses $c_{i_j,q} \in C_{i_j,h}$[4], $\forall j : 1 \ldots k$, $q : 1 \ldots n_j$, associating a static subclass with each element of the Cartesian product $cd(p)$. Each place $p_{[g]}$ is connected to the unfolded transitions of the same transitions as the original place p, with the corresponding arc functions prefixed by filter $[g]$. The second place partial unfolding step involves all those places p that have n (> 1) repetitions of a color class C_i in $cd(p)$. This unfolding can be done iteratively, considering one class C_i and one static subclass $C_{i,h} \in \widetilde{C_i}$ at a time. Let $p_{[g]}$ be a place obtained after the first unfolding step, and assume that filter $[g]$ includes m (> 1) predicates in the form $c_{i,q} \in C_{i,h}$, then the place

[3] In the case a class is not partitioned, we consider it as having only one static subclass.

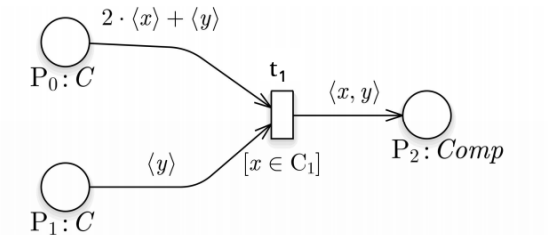
[4] The variable $c_{i_j,q}$ refers the q^{th} occurrence of C_{i_j}.

is unfolded into as many places as the number np of partitions of a set of cardinality m into at most $|C_{i,h}|$ parts. For each such a partition $\rho_j, j : 1 \dots np$ (which denotes a partition of the set of symbols $\{c_{i,q}\}$), the corresponding unfolded place p_{ρ_j} is associated with a predicate g'_j, which is the conjunction of a number of (in)equalities: g'_j contains equalities for the variables $c_{i,q}$ that are in the same subset of partition ρ_j, and an inequality for each pair $c_{i,q1}, c_{i,q2}$ belonging to different subsets in partition ρ_j. Each unfolded place, labelled with $p_{[g \wedge g'_j]}$, is connected to the same transitions as $p_{[g]}$, with the corresponding arc functions prefixed by filter $[g \wedge g']$ (in some cases this may result in a null arc). Observe that, as long as we consider unordered color classes and the static subclasses have cardinality greater than or equal to the maximum number of repetitions of the corresponding class in any place color domain, the partial unfolding of an SSN is independent on static subclass cardinalities. The number of copies of a partially unfolded place may be reduced by taking into account the form of the arc functions appearing on the arcs connected to the place.

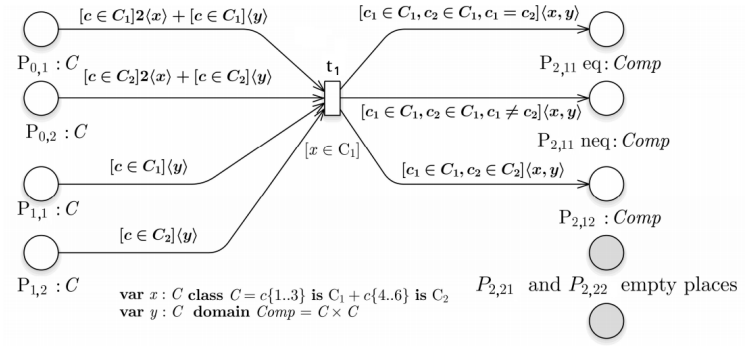
Let us introduce the simple SSN model depicted in Fig. 6.2a, which comprises a single transition t_1 with two input places, ${}^\bullet t_1 = \{P_0, P_1\}$, and one output place, $t_1^\bullet = \{P_2\}$. Input places have color domain C, where $C = C_1 \cup C_2$ is a basic color class partitioned in two subclasses. Note that in the color domain of place P_2 class C appears twice. Due to the transition guard, the first element of the pairs in P_2 must necessarily belong to C_1. Thus, the set $var(t_1)$ includes two type C variables x, y, moreover t_1 which has the guard $[x \in C_1]$, hence $cd(t_1) = C \times C$. The input arc functions, both of arity $C \times C \to Bag[C]$, are:

$$I[P_0, t_1] := 2\langle x \rangle + \langle y \rangle \qquad\qquad I[P_1, t_1] := \langle y \rangle.$$

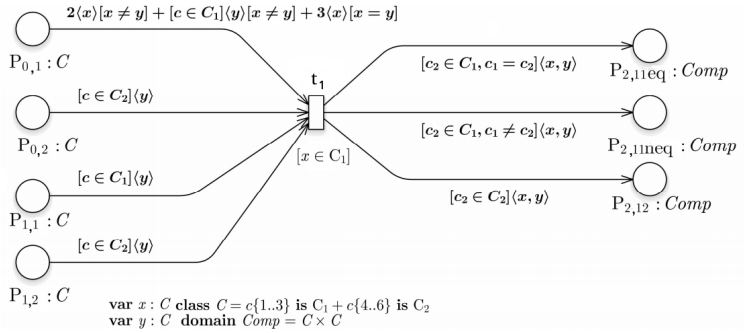
The partial unfolding, depicted in Fig. 6.2b, provides the duplication of both input places, to take into account the partition of C. We denote $P_{0,i}$ and $P_{1,i}$, $i = 1,\ 2$ the two instances of P_0 and P_1. The possible instances of P_2 instead are: $P_{2,11}$, $P_{2,12}$ with double index due to the repetition of C in $cd(P_2)$, and first index equal to one due to the transition guard. Place $P_{2,11}$ has repetition of class C in its color domain and when the two elements belong to the same static subclass the case in which the two elements are equal or different must



(a)



(b)



(c)

Figure 6.2: In (a) a simple example of an SSN model is shown, while (b) shows the same model after partial unfolding, and (c) after partial unfolding and simplification.

be separated: $P_{2,11eq}$ and $P_{2,11neq}$. Input arc functions are accordingly prefixed by filters $[c \in C_i]$, $i : 1, 2$. Similarly, in Fig. 6.2c it is shown a simplified version of the same partially unfolded model, where the empty places are omitted and the arc functions are rewritten by taking in consideration the only guard. For instance, let us focus on $P_{0,1}$, the instance of P_0 that holds tokens of color C_1. In fact the function $I[P_{0,1}, t_1]$ needs to be rewritten as (guard $[x \in C_1]$ is implicit):

$$3\langle x\rangle[x = y] + (2\langle x\rangle + \langle y\rangle)[x \neq y \wedge y \in C_1] + 2\langle x\rangle[y \in C_2]. \tag{6.3}$$

This expression points out that an instance $\langle c, c\rangle$ of t withdraws three tokens of color c, whereas an instance $\langle c, c'\rangle$, with $c \neq c'$, withdraws two tokens of color c and, if $c' \in C_1$, one of color c'. This affects the transition instance intensity that depends on the maximum number of tokens of a certain color withdrawn from a place. The input functions of the other places do not require any further manipulation: their *simplified* forms, implicitly taking the guard into account, are:

$$I[P_{0,2}, t_1] := \langle y\rangle[y \in C_2] \qquad I[P_{1,1}, t_1] := \langle y\rangle[y \in C_1]$$
$$I[P_{1,2}, t_1] := \langle y\rangle[y \in C_2]$$

In the more complex example model introduced in Section 6.1.1, the partial unfolding is limited to places *Drug*, *Storage* and *Infected*. Assuming that the color subclass ND has cardinality 1 and DC has cardinality n, the complete unfolding of these three places produce $(n + 1)$ copies of the first two places and $(n+1)^4$ copies of the third one. Instead, the first partial unfolding step would produce 2 copies of the *Drug* and *Storage*, and 2^4 copies of the *Infected*, the latter should be further unfolded to take into account the case where elements belonging to the same static subclass are equal or different. Finally some unfolded copies can be eliminated if they remain isolated when considering transition guards. In this example the number of non isolated copies is relatively small: *Drug* and *Storage* are connected only to transitions moving tokens in subclass DC; considering place *Infected* there are only five copies because ND has cardinality 1 (hence all elements in that class are surely equal), and the elements in DC are surely different due to the guards of any transition connected to the place. Moreover the guards introduce also restrictions on the way elements belonging to the two static subclasses are

interleaved in the tuples. The five copies of *Infected* correspond to the number of acquired antibiotic resistance, which in this model spans from 0 to 4. The arcs connecting transitions *Infection* and *Recovery* to place *Infected* are replicated for each of the five unfolded copies: the arc expressions are prefixed with the filter appropriate for the specific replica of *Infected*. Observe that increasing the size of *DrugChoice* does not change the structure of the partially unfolded net (while increasing the size of ND would change it). The model in Fig. 6.1 shows the five instances of place *Infected* and the connections between the unfolded places and transition *Unrecovery* after the unfolding: the filters g_6 to g_{10} appearing in the arc expressions show the static subclasses of each element of the tuples withdrawn from or added to the place and the relation (equality or inequality) among tuple elements belonging to the same static subclass.

6.1.3 Second step: symbolic ODE generation

In this section, we described the second step of new method to directly derives the associated set of $\widehat{ODE}$ starting from a partially unfolded SSN (the first step described in the previous section), by avoiding its complete unfolding. This method can be roughly summarize in three main steps: 1) computation of the firing intensity of macro terms, with a refinement into sub-terms characterized by uniform intensity, 2) calculation of the structural relations $\mathcal{A}$ and $\mathcal{R}$, by operating in a purely symbolic way, and 3) computation of the cardinality of the sets of transition instances represented by the refined terms (producing the same state changes with same rate).
Thus, in the rest of this section we described in details the preliminary rewriting of input functions, which will be used in the refinement of the symbolic expression (representing the ODE for the place) to take into account transition firing *intensity*. Successively, we defined the formulae for the derivation in symbolic form of the terms to be included in each $\widehat{ODE}$, and the procedure to express the cardinality (enabling degree) of transition instances appearing in the $\widehat{ODE}$.

Preliminary rewriting of input arc functions

A subtle point in the derivation of $\widehat{ODE}$ is the possibility that transition instances, that (after the partial unfolding) are assumed to have the same

base rate, have different firing intensities, according to the adopted marking dependency. The reason is that an input arc function may take different shapes when evaluated on different instances of a transition. A rewriting of input arc functions may be required, so that function shapes implicitly match the intensity-based logical partitioning of the set of transition instances.
Specifically, functions $I[p,t] : cd(t) \to Bag[cd(p)]$ have to be expressed as sum $\sum_j \lambda_j F_j$, $F_j = [g_j]T_j[g'_j]$, where: a) $j_1 \neq j_2 \Rightarrow (F_{j_1} \cap F_{j_2}) = \emptyset$; b) $g'_{j_1} \neq g'_{j_2} \Rightarrow (g'_{j_1} \wedge g'_{j_2}) =$ false (terms must be pairwise disjoint and the associated guards are either equal or mutually exclusive).
Arc functions[5] $I[p,t]$ can thus be partitioned into sub-sums $\sum_1 + \ldots \sum_m$ of (disjoint) terms having the same guard, i.e., $\sum_h = \left(\sum_{j_h} \lambda_{j_h}[g_{j_h}]T_{j_h}\right)[g'_h]$. For instance, if we consider the rewritten arc function showed in Eq. 6.3 regarding the SSN model depicted in Fig. 6.2c, then it is straightforward to observe the correspondence to the notation with the partition given by the guards:

$$
\begin{aligned}
&I[P_{0,1}, t_1] = \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (6.4)\\
&\quad \lambda_1 T_1[g'_1] \quad +(\lambda_{2,1}T_{2,1} + \lambda_{2,2}T_{2,2})[g'_2] \quad +\lambda_3 T_3[g'_3] =\\
&\quad 3\langle x\rangle[x = y] \quad +(2\langle x\rangle + \langle y\rangle)[x \neq y \wedge y \in C_1] \quad +2\langle x\rangle[y \in C_2].
\end{aligned}
$$

Guards $\{g'_h\}$, with the additional complementary one $g'_0 = \neg(\bigvee_h g'_h) \wedge g(t)$, define a *parametric partition* of $cd(t)$: all t's instances matching a given g'_h withdraw the same multiset of colors (up to a symmetry-preserving, color permutation) from p. The predicate g'_0 represents instances of t (if there are any) not withdrawing tokens from p.

Let $p \in {}^\bullet t$ such that $G_{t,p} = \{g'_0, \ldots, g'_m\}$ denote the intensity-based partition of t instances with respect to p. Let us define the map $\mu[t,p] : G_{p,t} \to \mathbb{N}$ as:

$$
\begin{cases}
\mu[t,p][g'_0] = 0\\
\mu[t,p][g'_h] = \eta_h \quad h \neq 0
\end{cases}
$$

where $\eta_h = \sum \lambda_{j_h}$ when the MA law is considered, but in general it depends on the law used to compute the transition intensities. For instance, $\mu[t_1, P_{0,1}][g'_1] = 3$ and $\mu[t_1, P_{0,1}][g'_2] = 2 + 1 = 3$.
If $|{}^\bullet t| > 1$, the partitions related to all t's input places $p \in {}^\bullet t$ must be combined in order to get a final one. This is done by calculating the *Cartesian product* $G_t = \otimes_{p \in {}^\bullet t} G_{t,p}$, which results in a set of tuples $\langle g_{p_1}, \ldots, g_{p_n}\rangle$, each interpreted

[5] over which the incident transition's guard $g(t)$ implicitly spans

as a conjunctive form (resulting $false$ elements are erased). By construction, G_t (as $G_{t,p}$) represents a partition of $cd(t)$. The map $\mu[t] : G_t \to \otimes_{p \in {}^\bullet t}\mathbb{N}$ associates intensity-equivalent classes of t's instances with corresponding tuples of coefficients. It is defined as follows: let ${}^\bullet t = \{p_1, \ldots, p_n\}$, $g_{p_i} \in G_{t,p_i}$

$$\mu[t](g_{p_1} \wedge \ldots \wedge g_{p_n}) = \langle \mu[t, P_1](g_{p_1}), \ldots, \mu[t, p_n](g_{p_n}) \rangle.$$

In the special case ${}^\bullet t = \{p\}$ then $\mu[t] = \mu[t, p]$. Guards mapping to the same tuple of coefficients may be proficiently replaced by a single equivalent OR term. The set G_t of guards will be used to refine the ODE symbolic expression on the basis of the intensity of t's instances. In the particular (but not rare) case $G_t = \{g(t)\}$ (meaning that all color instances of t have the same enabling degree), no refinement is needed. Observe that if in $\mu[t]$ there is an element mapping to a tuple with all zeroes, it would mean there are some instances of t not having any input place: this should be considered as a modeling error.

Let us refer again to the partial unfolded SSN model depicted in Fig. 6.2c. The map $\mu[t_1]$ applied to G_{t_1} is reported below in a matrix form (Eq. 6.5), considering the MA law defining the transition instances intensity (the membership clause $x \in C_1$ is implicit). Note that, since we are considering that transition t_1 is defined by the MA law, some of its instances are brought together. In particular the case $y \in C_1$ includes the sub-cases in which $x = y$ or $x \neq y$, because when the product of the number of tokens of the transition input places is computed there is no difference between $x^3_{P_{0,1}} \cdot x_{P_{1,1}}$ (i.e., $y \in C_1,\ x = y$) and $x^2_{P_{0,1}} \cdot x^1_{P_{0,1}} \cdot x_{P_{1,1}}$ (i.e., $y \in C_1,\ x \neq y$). Therefore, the 1^{st} row of the matrix in Eq. 6.5 means that an instance of t_1 of type $\langle c, c' \rangle$, $c, c' \in C_1$ withdraws three tokens of the *same* color from $P_{0,1}$, and one from $P_{1,1}$. These values are obtained from the λ values in Eq. 6.4 by applying the first guard ($[y \in C_1]$) to the $I[\cdot, t_1]$ for each input places of t_1. While the 2^{nd} row means that an instance of t_1 of type $\langle c, c' \rangle$, $c \in C_1,\ c' \in C_2$ withdraws two tokens from $P_{0,1}$, and one from $P_{0,2}$ and from $P_{1,2}$.

$$\mu[t]G_{t_1} = \begin{array}{c} \\ [y \in C_1] \\ [y \in C_2] \end{array} \begin{array}{c} \begin{array}{cccc} P_{0,1} & P_{0,2} & P_{1,1} & P_{1,2} \end{array} \\ \begin{pmatrix} 3 & 0 & 1 & 0 \\ 2 & 1 & 0 & 1 \end{pmatrix} \end{array} \tag{6.5}$$

Symbolic representation of ODE

Successively to the rewriting of the input arc to obtain a parametric partition of transition instances based on firing intensity, in this paragraph we analyse the symbolic expressions $\mathcal{A}(p,t)$ and $\mathcal{R}(p,t)$, which mainly define the $\widehat{ODE}$ corresponding to place p (Eq. 2.14). Indeed, these terms are defined as a sum of expressions which group several transition instances that add/withdraw the same number of tokens, if the number of tokens are different then the rewriting of the input arc explained above may induce a further refinement of these terms into sub-sums of disjoint terms (which now will add/withdraw the same number of tokens). After this observation, to derive the system of $\widehat{ODE}$ in terms of $\mathcal{R}$ and $\mathcal{A}$, we need to compute the cardinality of the tuples appearing in summations and the multiplicity of (generic) token colors required by the input arc functions, necessary to derive the enabling degree of transition instances used to compute the rate (intensity) (which will be explained in the next paragraph). Thus, in this paragraph we show that the functions $\mathcal{R}$ and $\mathcal{A}$ can be always written as sum of constant-size functions, whose cardinality can be easily computed.

As said, $\mathcal{R}$, $\mathcal{A}$ are formally represented as weighted sums of *tuples* $\sum_i \lambda_i F_i$, $\lambda_i \in \mathbb{N}$, $F_i = [g_i]T_i[g_i']$, $\forall c \in cd(p), \forall c' \in cd(t)\ F_i(c)[c'] \leq 1$. A term of $\mathcal{R}$ and $\mathcal{A}$ can be seen as a *parametric* set of t instances, each one withdrawing/putting λ_i tokens of color c from/to p. Hence we need to compute the *cardinality* of these parametric sets, which may depend on $c \in cd(p)$. So now we define a **constant-size function** which can be associated with a syntactical characterization necessary for computing tuple cardinality.

Definition 10 (Constant-size function)**.** *A function $F[g] : D \to Bag[D']$ is constant-size if and only if $\exists k \in \mathbb{N} : \forall c \in D, g(c) \Rightarrow |F(c)| = k$.*

The above definition includes the particular case $g \equiv true$. The *cardinality* $|F[g]|$ of a constant-size function is equal to $|F(c)|$, for any c s.t. $g(c)$ is true. A function tuple $t_1[g] \in \mathcal{L}$ is constant-size if and only if, for each t_1's component (class function) f, $f[g]$ is constant-size.

The following property defines a syntactical condition for a (guarded) class function $f[g]$ being constant size:

Property 1. *$f[g]$ is* constant-size *if: f either belongs to the basic-set BS of class functions or it takes one of these forms*

$$a) \quad \bigcap_{j\in Q,\,|Q|<|C|} S - v_j \qquad b) \quad S_{C_k} \bigcap_{j\in J,|J|<|C_k|} S - v_j$$

where in b) for each v_j: $g \Rightarrow v_j \in C_k$.

The cardinalities of terms of type a) and b) are $|C| - |Q|$ and $|C_k| - |J|$, respectively. The cardinalities of functions in BS can be easily inferred. For instance, if we consider a function defined as $f = S - v_1 \cap S - v_2[v_1 \neq v_2]$, where v_1, v_2 are two variables of type C, then f is constant size with cardinality $|C| - 2$.

We finally state a syntactical condition on a filter $[g']$ ensuring that $[g']T[g] \in \mathcal{L}$ is constant-size.

Property 2. *$[g']T[g] \in \mathcal{L}$ is* constant-size *if $t_1[g]$ is constant size and*

1. *g' is a conjunctive form composed only of (in)equations $c_i = (\neq)c_j$, $i < j$,*
2. *for each (in)equation $c_i = (\neq)c_j$ the corresponding class-C functions f_i, f_j in t_1 are such that $f_j \equiv f_i$ (i.e., tuple components referred to by any (in)equation in the filter must be equal).*

Let us constructively prove Property 2 by sketching the general algorithm for computing tuple cardinality. We can express $[g]T[g'] : D \rightarrow Bag[D']$ as $(\bigotimes_{C\in D'}[g_C]T_C)[g']$, where $[g_C]T_C : D \rightarrow Bag[C^e]$, and e the number of repetitions of C in D'. In other words we consider separately the subtuples of t_1 involving each class C and the terms in g involving those components. Note that it may be $g_C = true$ for some colour class C. The function $[g]T[g']$ is constant-size iff every $[g_C]T_C[g']$ is constant-size and in this case $\left|[g]T[g']\right| = \prod_C \left|[g_C]T_C[g']\right|$.

Let us focus on $[g_C]T_C[g']$. Let $J(g_C) = \{j\}$, s.t. c_j occurs in g_C, in other words $J(g_C)$ identifies the set of variables c_j of type C appearing in g_C. We can partition g_C (a conjunctive form) into $\{g_1, \ldots, g_n\}$, such that for each g_i, g_j, $i \neq j$, $J(g_i) \cap J(g_j) = \emptyset$ (in this way we separate independent subsets g_i of terms in g_C)). The terms in g_i can be partitioned in equalities and inequalities: let us introduce the notation $g_i = g_{i,eq} \wedge g_{i,neq}$ to separate the two parts of g_i. Note that $g_{i,eq}$ or $g_{i,neq}$ may be simply $true$. Without loss of generality, we assume that equalities in $g_{i,eq}$ take all the form $c_j = c_x$ (for an arbitrarily

fixed c_j), and $g_{i,eq} \not\equiv true \wedge g_{i,neq} \not\equiv true \Rightarrow J(g_{i,eq}) \cap J(g_{i,neq}) = \{j\}$, in other words if g_i contains both equalities and inequalities there is just one variable c_j occurring both in $g_{i,eq}$ and in $g_{i,neq}$. Under the initial hypothesis, all elements in subtuple T_C corresponding to the index set $J(g_i)$ are *equal*. Let λ (> 1) be their cardinality, and let us denote $card_i$ the cardinality of the T_C's subtuple corresponding to $J(g_i)$ after being filtered through g_i. If $g_{i,neq} \equiv true$ (g_i just contains equalities) then $card_i$ is simply λ. Otherwise $g_{i,neq}$ can be seen as a system of inequalities among $n = |J(g_{i,neq})|$ integer variables on the domain $\{1, \ldots, \lambda\}$. Let G be the connected graph of order n representing such a system: the number of system's solutions ($= card(i)$, for the particular form of g_i) is the *chromatic polynomial* value $P(G, \lambda)$, corresponding to the number of distinct λ-colourings of G. Finally, the cardinality of $[g_C]T_C[g']$ is obtained by multiplying $\prod_i card(i)$ by the cardinality of T_C components not corresponding to any index in $J(g_i)$.

Considering the SSN example depicted in Fig. 6.2c, $\mathcal{R}$ expression for the place $P_{0,1}$ can be obtained as follows:

$$\mathcal{R}(P_{0,1}, T) = 2 < c_1, S_{C2} > [c_1 \in C1] + 3 < c_1, c_1 > [c_1 \in C1],$$

where the expression $1 < S - c_1 \cap S_{C1}, c_1 > [c_1 \in C1] + 2 < c_1, S - c_1 \cap S_{C1} > [c_1 \in C1]$ is not considered since with MA law we not take in account the case in which y and x are different and in C_1.
Thus, the instances of t_1 that withdraw tokens from this place are given by $\mathcal{R}(P_{0,1}, T)$ above, according to its expression such instances are partitioned into two disjoint sets.

- Any instance $< t_1, c, c' >$ of transition t_1 belonging to $\langle c_1, S_{C_2} \rangle [c_1 \in C_1]$ withdraws 2 tokens $c \in C_1$ from $P_{0,1}$, 1 token $c' \in C_2$ from $P_{0,2}$, 1 token $c' \in C_2$ from $P_{1,2}$. There are $|C_2|$ instances of t_1 of this kind.

- Any instance $< t_1, c, c >$ of transition t_1 belonging to $3\langle c_1, c_1 \rangle [c_1 \in C_1]$ withdraws 3 tokens $c \in C_1$ from $P_{0,1}$ and 1 token of the same color from $P_{1,1}$.There is only one transition instance of this kind.

Let us now consider the case study introduced in Section 6.1.1. The term

of $\mathcal{R}(Infected3xDrug, Unrecovery)$:

$$\langle d_1, d_2, d_3, S - d_1 \cap S - d_2 \cap S - d_3 \cap S_{\mathrm{DC}} \rangle$$
$$[d_1 \neq d_2 \wedge d_1 \neq d_3 \wedge d_2 \neq d_3 \wedge d_1 \in \mathrm{DC} \wedge d_2 \in \mathrm{DC} \wedge$$
$$d_3 \in \mathrm{DC} \wedge d_4 \in \mathrm{ND}]$$

has a guard ensuring that the intersection in the tuple is constant-size: its cardinality (and that of the tuple) is $|\mathrm{DC}| - 3$. Similarly, the term of $\mathcal{R}(Susceptible, Infection)$:

$$[d_1 \neq d_2 \wedge d_1 \neq d_3 \wedge d_2 \neq d_3] \langle S_{\mathrm{DC}}, S_{\mathrm{DC}}, S_{\mathrm{DC}}, S_{\mathrm{ND}} \rangle$$

has a filter, that has to be taken into account: letting $n = |\mathrm{DC}|$, $m = |\mathrm{ND}|$, the term's cardinality is expressed by the formula $n(n-1)(n-2)m$. The product $n(n-1)(n-2)$ is the number of triplets with all different elements (chosen from a set of n).

Finally, introducing the following property it possible to guarantee that terms $F_i = [g_i]T_i[g_i']$ appearing in $\mathcal{R}$ or $\mathcal{A}$ are *constant-size*.

Property 3. *Any expression $e \in \mathcal{L}$ can be rewritten as a weighted sum of constant-size tuples $[g_i']T_i[g_i]$.*

Thus, according to the transpose semantics, one such term $[g_i]T_i[g_i']$ of $\mathcal{R}$ or $\mathcal{A}$ represents a *set* of $n_i = |F_i|$ colour instances of t, each withdrawing/adding *exactly* λ_i (the term's coefficient in the weighted sum) tokens from/to place p (these instances satisfy the predicate g_i').

If, in addition, all t's colour instances matching $[g_i]T_i[g_i']$ had the same intensity (denoted by $\varphi(x(\nu), t)$), we could directly express the $\widehat{ODE}$ relating place p as:

$$\frac{dx[p,c]}{d\nu} = \sum_{t:p\in t^\bullet, F_i \, in \, \mathcal{A}(p,t)} \lambda_i n_i \varphi(x(\nu), t) - \sum_{t:p\in {}^\bullet t, F_j \, in \, \mathcal{R}(p,t)} \lambda_j n_j \varphi(x(\nu), t) \tag{6.6}$$

Each term in the $\widehat{ODE}$ is a product of four factors: the cardinality of the expression identifying a set of (n_i) homogeneous transition instances, the number (λ_i) of tokens withdrawn/added by any transition instance in the set,

the base rate w of transition instances in the set, and the marking-dependent factor (the two last factors are combined in φ). The latter depends on the number of coloured tokens at time ν required by the arc functions labelling the input arcs of any transition instance in the set.
Some terms $[g_i]T_i[g'_i]$ of $\mathcal{A}$ or $\mathcal{R}$, however, may have to be *preliminarily* split into equivalent sums of tuples representing classes of transition instances characterized by the same intensity (enabling degree).

Computation of the enabling degree

Let us consider the $\widehat{ODE}$ for place p expressed in Eq. 2.14. We saw above that the contribution due to a transition t connected to p is expressed by $\mathcal{R}(p,t)$ or $\mathcal{A}(p,t)$, whose weighted terms $\lambda_i F_i$ represent parametric sets of $n_i = |F_i|$ instances of t, that withdraw/add exactly λ_i tokens from/to p. We may have to split these terms into subterms denoting instances with the same intensity, and we may have to derive the formal expression of the marking-dependent factor. Both tasks are straightforwardly carried out by using the set G_t of guards (representing the intensity-based symbolic partition of t's instances) and the associated map $\mu[t]$, computed during the rewriting of the input arcs of the SSN (Paragraph *Preliminary rewriting of input arc functions*).
If $|G_t| > 1$ (i.e., the intensity-based partition of t's instances is not trivial), G_t guards are used as *filters* to possibly split the parametric sets $F_i = [g_i]T_i[g'_i]$, with $n_i > 1$, into subsets with the same intensity, formally:

$$\lambda_i F_i \mapsto \lambda_i(\textstyle\sum_{g\in G_t}[g \wedge g_i]T_i[g'_i]).$$

This rewriting is coherent, resulting in an equivalent expression, because the domain of any $g \in G_t$ is $cd(t)$, and $\bigvee_{g\in G_t} g \equiv g(t)$ (G_t is a partition of $cd(t)$).

Let F'_i be a subterm of F_i obtained by applying a filter $g \in G_t$ to F_i ($F'_i = F_i \leftrightarrow G_t = \{g(t)\}$). The associated marking-dependent factor to be used in the $\widehat{ODE}$ is directly obtained from the tuple of coefficients $\mu[t](g) = \langle \eta_{p_1}, \ldots, \eta_{p_n} \rangle$, which takes the MA law form:

$$\prod_{p_k \in {}^\bullet t} x[p_k]^{\eta_{p_k}}.$$

Let us return on the simple model introduced at the begin of this section, Fig. 6.2. The $\mathcal{R}$ terms for the partially unfolded input places P_0, P_1 are (there

are no $\mathcal{A}$ terms for these places):

$$\begin{aligned}
\mathcal{R}(P_{0,1}, t_1) =& 2\langle c_1, S_{C2}\rangle[c_1 \in C1] + 3\langle c_1, c_1\rangle[c_1 \in C1] \\
\mathcal{R}(P_{0,2}, t_1) =& 1\langle S_{C1}, c_1\rangle[c_1 \in C2] \\
\mathcal{R}(P_{1,1}, t_1) =& 1\langle S_{C1}, c_1\rangle[c_1 \in C1] \\
\mathcal{R}(P_{1,2}, t_1) =& 1\langle S_{C1}, c_1\rangle[c_1 \in C2]
\end{aligned}$$

Let us consider the $\widehat{ODE}$ from the point of view of $P_{0,1}$ (the restriction of P_0 for $c \in C_1$). The instances of t_1 that withdraw tokens from this place are given by $\mathcal{R}(P_{0,1}, t_1)$ above. According to its form, such instances are partitioned into two pair-wise disjoint sets, so applying the filters (guards that are used as filters) in $G_{t_1} = \{[y \in C_1],\ [y \in C_2]\}$ to $\mathcal{R}(P_{0,1}, T)$ (i.e., by using one guard per time it is possible to omit part of the instances) the $\widehat{ODE}$ for $P_{0,1}$ is (w_t is the transition's base rate):

$$\begin{aligned}
\frac{dx[P_{0,1}]}{d\nu} = -w_t \cdot (& 2 \cdot |C_1| \cdot x[P_{0,1}]^3 \cdot x[P_{1,1}]^1 + \\
& 1 \cdot |C_1| \cdot x[P_{0,1}]^3 \cdot x[P_{1,1}]^1 + \\
& 2 \cdot |C_2| \cdot x[P_{0,1}]^2 \cdot x[P_{0,2}]^1 \cdot x[P_{1,2}]^1).
\end{aligned}$$

Finally, we obtained the $\widehat{ODE}$ after 1) a partial unfolding of the net which ensures the uniform rate (w_t) for all instances $\langle t, c\rangle,\ c \in cd(t)$ for any transition t; 2) the rewriting of the input arc functions to obtain a parametric partition of transition instances based on firing intensity; 3) we showed the the functions $\mathcal{A}$ and $\mathcal{R}$ are sums of constant size terms F_i (so we can calculate their cardinality); 4) we computed the enabling degree for the instances grouped in a term of $\mathcal{A}$ and $\mathcal{R}$, which are unformized by applying the rewriting of the input arc functions; 5) for each term F_i, representing a parametric set of transition instances with uniform intensity $\phi()$, only one term is generated in the $\widehat{ODE}$ and multiplied by $|F_i|$.

6.1.4 Application of the method to the case study

The new method for the automatic derivation of a set of symbolic ODE ($\widehat{ODE}$) from an SSN presented in this section has been implemented in a new module of *SNexpression* (tool targeted at the structural analysis of SSNs) [54], `www.di.unito.it/~depierro/SNex`. Thus, in this section the results on the SIRS

example presented in Section 6.1.1 are showed: the number of equations and overall size of the $\widehat{ODE}$ system is compared with the corresponding sizes of the ODE system obtained from the complete unfolding of the SSN model. The method based on the complete unfolding prevents the computation of indices when the size of one class grows, while the $\widehat{ODE}$ system size does not change (only the coefficients in the $\widehat{ODE}$ are updated).

Figure 6.3 plots the average marking of four places of the SIRS model: the results have been computed both with the set of ODE obtained from the completely unfolded model (up to a certain size), and with the set of $\widehat{ODE}$ obtained with the *SNexpression* tool: the plotted lines show a perfect matching, indeed the relative difference is very small (always below 10^{-7}, the precision set for the numerical solution of the system of differential equations). Table 6.3 shows the size of the system of ODE and of $\widehat{ODE}$ for different sizes ($n = 5, \ldots, 10$) of static subclass DC showing the limit of the solution based on the complete unfolding.
The files describing both nets (described according to the *SNexpression* syntax) and the $\widehat{ODE}$ system automatically obtained for each model are available for download in the *SNexpression* web page[6].

Complete Unfolding						
n	5	6	7	8	9	10
♯ ODE	1311	2418	4115	6582	10023	14666
♯ Trans.	831	2076	4409	8334	14451	23456
R size (MB)	0.350	0.718	1.6	2.8	4.6	7.4
Sol. time (s)	30	67.45	131.98	785.94	1770.35	n.c.
Mem. peak (GB)	0.96	1.18	1.6	2.9	6.6	13.4
Partial Unfolding ($\widehat{ODE}$)						
n	> 4					
♯ $\widehat{ODE}$	12					
♯ terms	74					
R size	6KB					
Sol. time (s)	0.5					

Table 6.3: Comparison of the size of the set of ODE and the set of $\widehat{ODE}$ varying $n = |\mathrm{DC}|$.

[6] `http://www.di.unito.it/~depierro/SNexpression/#Documentation%20and%20sample%20files`

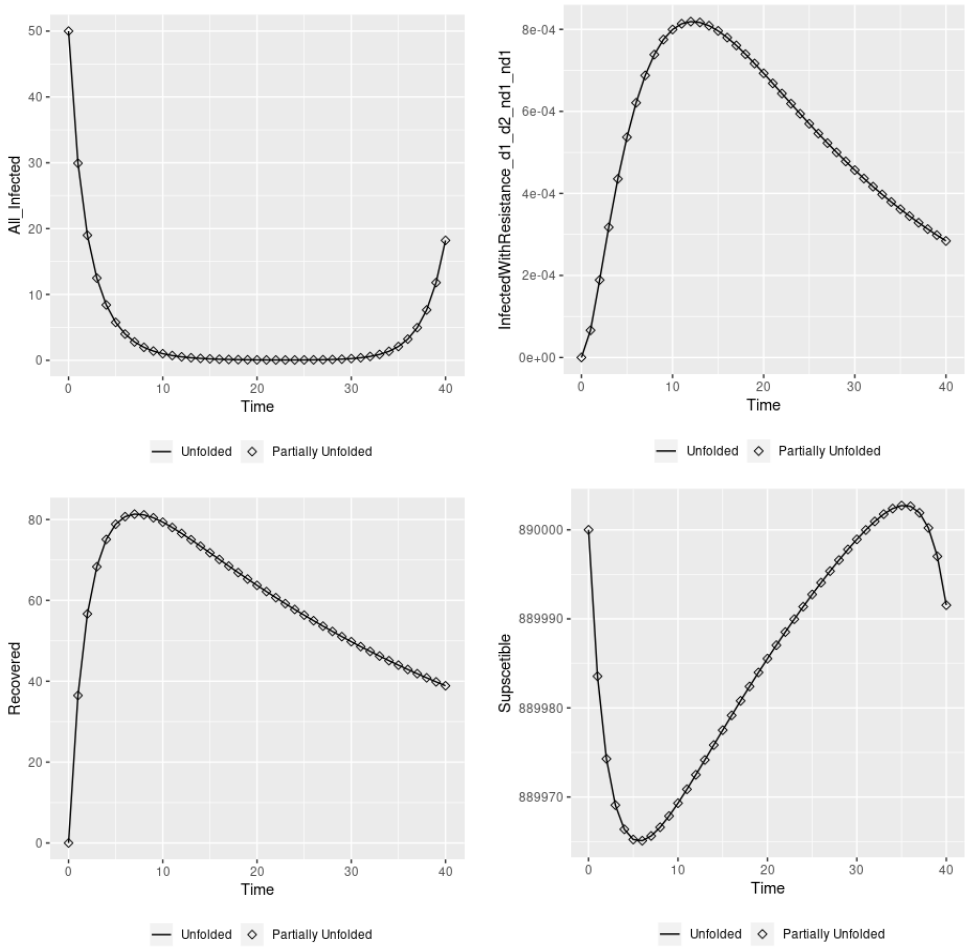


Figure 6.3: Some measures obtained from the completely unfolded model and the compact one: in each plot the two curves match.

6.1.5 Extension to the ESSN

In this section we discuss how this method can be extended to the ESSN formalism. Let us recall from Section 5.2 that in an ESSN model we have to deal with two types of transitions: standard ($t \in T_{ma}$) and general ($t \in T_g$). Since in the previous sections we showed the applicability of the algorithm when all the transitions in the model are characterized by the MA law, now we describe how this can be extended for general transitions by adding constraints on the function $f_{\langle t,c \rangle}$ associated with the transition.

Let $f_{\langle t,c \rangle} \in \Omega$ (see Definition 9) be the continuous function associated with

a color instance of the general transition $t \in T_g$, and depending only on the marking of the input places of transition t at time ν. First, we can observe that the definition of $f_{\langle t,c\rangle}$ must preserve the equivalence among markings belonging to the same symbolic marking (SM) $\widehat{\boldsymbol{m}}$. As described in Section 2.4, given two equivalent markings $\boldsymbol{m}_1,\ \boldsymbol{m}_2 \in \widehat{\boldsymbol{m}}$, there exists a correspondence between the transition instances enabled in $\boldsymbol{m}_1$ and those enabled in $\boldsymbol{m}_2$. Indeed, if s is the permutation that allows one to obtain marking $\boldsymbol{m}_2$ from $\boldsymbol{m}_1$, i.e. $\boldsymbol{m}_2 = s.\boldsymbol{m}_1$, for each transition instance $\langle t, c\rangle$ enabled in $\boldsymbol{m}_1$ there exists a transition instance $\langle t, s.c\rangle$ enabled in $\boldsymbol{m}_2$ with the same rate. Moreover markings $\boldsymbol{m}_1'$ reached by firing $\langle t, c\rangle$ in $\boldsymbol{m}_1$, and $\boldsymbol{m}_2'$ reached by firing $\langle t, s.c\rangle$ in $\boldsymbol{m}_2$ are equivalent (i.e. they belong to the same SM $\widehat{\boldsymbol{m}}'$: indeed $\boldsymbol{m}_2' = s.\boldsymbol{m}_1'$). Given that this holds for each transition instance with same rate, then it is straightforward to define the syntax of the general transition intensity $f_{\langle t,c\rangle},\ t \in T_g$, such that

$$f_{\langle t,c\rangle}(\hat{\boldsymbol{m}}_1, \nu) = f_{\langle t,s.c\rangle}(\hat{\boldsymbol{m}}_2, \nu) \tag{6.7}$$

where $\hat{\boldsymbol{m}} = \boldsymbol{m}_{|\bullet \mathbf{t}}$ as the subset of the marking $\boldsymbol{m}$ at time ν concerning only the input places to the transition t. In terms of the Definition 4, considering the mapping $\widetilde{c} : C \to \mathbb{N}$ from a color to the index of its corresponding color class, and the mapping $\widetilde{s} : C \to \mathbb{N}$ from a color to the index of its corresponding static subclass, then given two color instances $c = \langle c_1, \ldots, c_n\rangle$ and $c' = \langle c_1', \ldots, c_n'\rangle$, of the general transition $t \in T_g$ (i.e., belonging to $cd(t)$) the following implication is verified for any marking $\hat{\boldsymbol{m}}$ and time ν

$$\forall c,\ c' \in cd(t) :\ \widetilde{c}(c) = \widetilde{c}(c') \wedge \widetilde{s}(c) = \widetilde{s}(c') \to f_{\langle t,c\rangle}(\hat{\boldsymbol{m}}, \nu) = f_{\langle t,c'\rangle}(\hat{\boldsymbol{m}}, \nu).$$

Indeed, if this condition is true then the marking equivalence should hold making possible the $\widehat{ODE}$ generation.

Let us consider the ESSN model introduced in Section 5.3 regarding the prey-predator model in Fig. 5.5b. The general transition of the prey death from Eq. 5.10 applied to the ESSN model can be written as follows:

$$f_{\langle DeathPrey,\bar{t},\bar{s}\rangle}(\hat{m}(\nu), \nu) = \frac{a\ m_{Prey_{\bar{t},\bar{s}}}}{1 + a\ h\ m_{Prey_{\bar{t},\bar{s}}}}\ m_{Predator_{\bar{t}}} \tag{6.8}$$

where $\hat{m}(\nu) = [m_{Predator_{\bar{t}}}, m_{Prey_{\bar{t},\bar{s}}}], \forall \bar{t} \in cd(Predator), \langle \bar{t}, \bar{s}\rangle \in cd(Prey)$, with $m_P \in \hat{m}(\nu)$ is the number of tokens at time ν in place P, for all the color

variables $\bar{t}$, $\bar{s}$ identifying the position and the prey species respectively. While the constant parameter a is the attack rate at which the consumer encounters food items per unit of food density, and h is the average handling time spent on processing a food item. Let us recall that the prey-predator model depicted in Fig. 5.5b is characterized by two color classes: 1) $Territory$ representing the different positions of an animal, which is defined by two subclasses $T1$ and $T2$ composed respectively by three and two colors, and 2) $Species$ modeling two different species defined by the two subclasses $S1$ and $S2$, both composed by one color. In particular, the color domain of the $Prey$ place is defined by both the classes, while the color domain of the $Predator$ place is characterized only by the $Territory$ color class. Thus, assuming that $\forall \bar{t}, \bar{s}$ belonging to specific subclasses (e.g., $T1$ and $S1$ respectively) the initial marking of the instance places $m_{Predator_{\bar{t}}}$ and $m_{Prey_{\bar{t},\bar{s}}}$ are not different with respect to the $\bar{t}$, $\bar{s}$ colors, then it is straightforward to conclude that the function in Eq. 6.8 associated with the $DeathPrey$ transition does not assume different values varying $\bar{t} \in T1$(or $T2$) and $\bar{s} \in S1$(or $S2$). Instead, if the attack rate (a) or the marking of one place would have assumed different values with respect to $\bar{t} \in T1$, then we could not exploit the symmetries in $T1$ and the condition in Eq. 6.7 would not hold anymore.
Finally, to apply the method, which depends on the intensity law defining each transition (in the previous sections we considered only standard transitions which are defined by the MA law), we need to define the same intensity (i.e., equal enabling degree) to all the transition instances characterized by the same function. Indeed, in this case the preliminary rewriting of input arc functions is not necessary and the terms in the $\widehat{ODE}$s regarding the general transitions can be written directly.

6.2 Hybrid Model

In this section we explained how ESPN can be successfully exploited to combine signaling, metabolic, and regulatory networks in a unique model. In particular we focused on integrating the Flux Balance Analysis (FBA), used for large scale analysis of metabolic network (see Chapter 4), with the more detailed model based on Ordinary Differential Equations (ODEs). Thus, this section is organised as follows: in Section 6.2.1 we introduced the mathematical formalisation of this cross-talk between different models by exploiting

the ESPN formalism. In this contest we focused on two aspects: 1) how to define the fluxes constraints, and 2) how to synchronize the solution of the two modeling approaches. In Section 6.2.2 we showed how to exploit ESPN formalism to combine the FBA model with the ODEs system characterizing the dynamics of the main metabolic pathways altered in **Pancreatic Ductal Adenocarcinoma** cells. This case study is principally a proof of concept of our approach and it was published in [115].

6.2.1 Hybrid model implementation

In Sec. 5.1 we defined the generic ODE of a place p (Eq. 5.1) as composition of two sums corresponding to state change due to standard transitions and general transitions. In this section we further refine this equation splitting the set of general transitions into two disjoint subsets T_g^{FBA} and $T_g^{\neg FBA}$, so that i) $T_g^{FBA} \cup T_g^{\neg FBA} = T_g$, and ii) $t \in T_g^{FBA}$ iff its rate is estimated by solving an FBA model. Thus, the instantaneous changes of tokens $x_i(\nu)$ in the i^{th} place at time ν expressed in the Eq. 5.1 can be rewritten as follows:

$$\begin{aligned} dx_i(\nu) &= \sum_{j=1}^{n_{T_{ma}}} \phi(t_j^*, \hat{x}) S(p_i, t_j^*) d\nu + \sum_{j=1}^{n_{T_g}} f_{\bar{t}_j}(\nu, \hat{x}) S(p_i, \bar{t}_j) d\nu \\ &= \sum_{j=1}^{n_{T_{ma}}} \phi(t_j^*, \hat{x}) S(p_i, t_j^*) d\nu + \sum_{j=1}^{n_{T_g^{\neg FBA}}} f_{\bar{t}_j}(\nu, \hat{x}) S(p_i, \bar{t}_j) \\ &\qquad + \sum_{j=1}^{n_{T_G^{FBA}}} f_{\bar{t}_j}(\nu, \hat{x}) S(p_i, \bar{t}_j) d\nu \\ &\forall i \in \{1, \ldots, n_P\}, \end{aligned} \tag{6.9}$$

where the notation n_{set} indicates the cardinality (number of transitions) of set, $\phi(*)$ is the standard transitions intensity following the MA law, and $f_{\bar{t}}(\nu, \hat{x})$ is the function associated with the general transition $\bar{t} \in T_g^{\neg FBA}$, which depends on its input places marking $\hat{x}$ at time ν. In this way the $f_{\bar{t}}(\nu, \hat{x})$ associated with a general transition $\bar{t} \in T_g^{FBA}$ must be defined as follows:

$$f_{\bar{t}}(\nu, \hat{x}) = v_{\bar{t}} \text{ s.t. } \exists i \in \{1, \ldots, m\} : v_{\bar{t}} = v_i^k \ \wedge \ \bar{t} \in T_g^{FBA} \tag{6.10}$$

with v^k conditioned to the k^{th} linear programming problem (LPP)

$$max\ f^k_{obj}(v^k) \text{ or } min\ f^k_{obj}(v^k),$$

subject to

$$S^k \cdot v^k = 0,$$
$$f_{lower}(v^k_i, \hat{x}) \leq v^k_i \leq f_{upper}(v^k_i, \hat{x}),$$
$$v^k_i \in \mathbb{R}, \qquad \forall i = 1, \ldots, m^k,$$

where $v^k \in \mathbb{R}^m$ is the fluxes vector, m^k the number of reactions in the k^{th} network considered by the FBA, f^k_{obj} the objective function, S^k the stoichiometric matrix. Let us highlight that the $k(\geq 1)$ value implies the possibility to consider multiple independent LPPs. Indeed, the k^{th} LPP is in common to all the $t \in T^{FBA}_g$ such that the respective flux belongs to v^k (i.e., the fluxes vector corresponding to the k^{th} LPP). The functions f_{lower} and f_{upper} represent the lower and upper bounds in which each v^k_i flux varies, we will discuss later how to define them. Let us now consider a simple ESPN example showed in Fig. 6.4, in which we used a blue-shaded rectangle to highlight the sub part of the net modeled with the FBA. Hereafter for simplicity we will omit the k notation since we consider only one LPP (i.e., $k = 1$). Indeed,

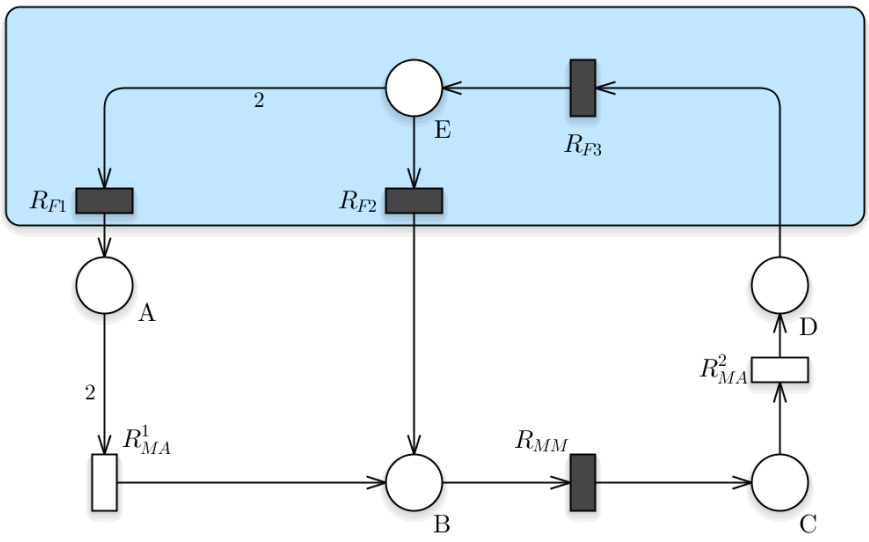


Figure 6.4: A simple example of ESPN model including different types of generic transitions. Rates of reactions in the blue-shaded rectangle are estimated by linear programming problems.

in this example the set of standard transitions T_{ma} contains $R^{[1]}_{MA}$ and $R^{[2]}_{MA}$

transitions; while the set of generic transitions T_g contains R_{MM}, R_{F1}, R_{F2} and R_{F3} transitions (i.e. $\{R_{F1},\ R_{F2},\ R_{F3}\} = T_g^{FBA}$ and $\{R_{MM}\} = T_g^{\neg FBA}$). According to this, the speeds of $R_{MA}^{[1]}$ and $R_{MA}^{[2]}$ follow the MA law and they are thus defined as $\phi(R_{MA}^{[1]}, x_A) = \lambda(R_{MA}^{[1]})x_A^2$ and $\phi(R_{MA}^{[2]}, x_C) = \lambda(R_{MA}^{[2]})x_C$ respectively. Conversely, R_{MM} represents a Michaelis-Menten reaction so that its rate is $f_{R_{MM}}(\nu, x_B) = \dfrac{V_{max}x_B(\nu)}{K_M + x_B(\nu)}$ where $V_{max},\ K_M$ are the maximum rate and the Michaelis constant characterizing such reaction. Finally the velocity of transitions $R_{F1},\ R_{F2},\ R_{F3}$ (transitions that define the set T_g^{FBA}) are estimated by FBA on the metabolic network (which is composed by only the places in the blue-shaded rectangle. Specifically, they are obtained by solving a LPP, such as:

$$\max(V_{R_{F1}}) \qquad s.t. \qquad \overbrace{\begin{bmatrix} -2 & -1 & +1 \\ 0 & 0 & -1 \end{bmatrix}}^{stoichiometric\ matrix} \begin{bmatrix} V_{R_{F1}} \\ V_{R_{F2}} \\ V_{R_{F3}} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \tag{6.11}$$

$$\begin{aligned} f_{lower}(R_{F1}, x_E) <& V_{R_{F1}} < f_{upper}(R_{F1}, x_E), \\ f_{lower}(R_{F2}, x_E) <& V_{R_{F2}} < f_{upper}(R_{F2}, x_E), \\ f_{lower}(R_{F3}, x_D) <& V_{R_{F3}} < f_{upper}(R_{F3}, x_D), \end{aligned}$$

where $f_{lower/upper}(\cdot)$ are the constraints, which depend on the respective transition and its input places, and $V_{R_{F1}}, V_{R_{F2}}, V_{R_{F3}}$ are the fluxes of the transitions R_{F1}, R_{F2}, R_{F3}, respectively. Therefore, following the notations introduced above, we define $f_{R_{F1}}(\nu, x_E) = V_{R_{F1}}$, $f_{R_{F2}}(\nu, x_E) = V_{R_{F2}}$, and $f_{R_{F3}}(\nu, x_D) = V_{R_{F3}}$.
As a consequence, the ODE system defined by equations such 6.11, characterizing the model quantities in Fig. 6.4 is:

$$\frac{dx_A(\nu)}{d\nu} = \overbrace{-2\ \phi(R^{[1]}_{MA}, x_A)}^{\text{transition } R^{[1]}_{MA}} \overbrace{+1\ f_{R_{F1}}(\nu, x_E)}^{\text{transition } R_{F1}}$$

$$\frac{dx_B(\nu)}{d\nu} = \overbrace{+1\ \phi(R^{[1]}_{MA}, x_A)}^{\text{transition } R^{[1]}_{MA}} \overbrace{+1\ f_{R_{F2}}(\nu, x_E)}^{\text{transition } R_{F2}} \overbrace{-1\ f_{R_{MM}}(\nu, x_B)}^{\text{transition } R_{MM}}$$

$$\frac{dx_C(\nu)}{d\nu} = \overbrace{+1\ f_{R_{MM}}(\nu, x_B)}^{\text{transition } R_{MM}} \overbrace{-1\ \phi(R^{[2]}_{MA}, x_C)}^{\text{transition } R^{[2]}_{MA}}$$

$$\frac{dx_D(\nu)}{d\nu} = \overbrace{-1\ f_{R_{F3}}(\nu, x_D)}^{\text{transition } R_{F3}} \overbrace{+1\ \phi(R^{[2]}_{MA}, x_C)}^{\text{transition } R^{[2]}_{MA}} \quad .$$

Notice that, since place E is studied under the hypothesis of Flux Balance, we decided to omit the corresponding differential equation because its variation is always equal to zero.

Constraints definition. The functions f_{lower} and f_{upper} represent the lower and upper bounds in which each v_i flux varies, which may depend on the marking of the transition input places. Indeed, as reported in [37], there are several types of metabolic flux constraints which can be used to define the f_{lower}/f_{upper} functions:

1. **irreversibility constraints**, where the lower bound of the reaction is set to zero for reactions which can only proceed in the forward direction [36];

2. **environmental constraints**, where the maximum flux through an exchange reaction is limited by the amount of substrate in the culture medium [158];

3. **transport constraints**, which are represented as a maximum substrate uptake[7];

4. **regulatory constraints**, where the flux through an enzyme is restricted by the expression of the corresponding protein(s) [164];

[7]Typically the rate of uptake of nutrients is dictated by availability (a nutrient that is not present cannot be absorbed), concentration and diffusion constants (higher concentrations of quickly-diffusing metabolites are absorbed more quickly).

5. **ODE matching constraints**, where the fluxes are specified by the ODE model.

The first three type of constraints are derived from the literature (and it is the classical manner to define the constraints), so the functions can be expressed as $f_{lower/upper}(v_i, \hat{x}) = k_i^{lower/upper}$, where $k_i^{lower/upper} \geq 0$ are constants. Indeed, when the flux does not depend on the ODEs values than the $\hat{x}$ dependency can be omitted. For instance, considering the simple example showed above, we can define $f_{lower}(V_{R_{F2}}) = 0$ and $f_{upper}(V_{R_{F2}}) = k_2$.
The regulatory constraints can be obtained by encoding in the function associated with the general transition a Boolean regulatory model, i.e., a set of Boolean logic equations which involve restricting expression of a transcription unit (sequence of nucleotides in DNA that codes for a single RNA molecule) to the value 1 if the transcription unit is transcribed and 0 if it is not. Similarly, the presence of an enzyme or regulatory protein, or the presence of certain conditions inside or outside of the cell, may be expressed as 1 if the enzyme, protein, or a certain condition is present and 0 if it is not. In this thesis, we decided to not consider this type of constraints, but to focus on the ones which depend on the ODEs characterizing the part of the ESPN which is not considered in the FBA. In this case, as shown in [37], the upper and lower bounds of the FBA fluxes depend by the entities modeled from the ODE system, and they are settled equal to the corresponding rate calculated by the ODE model. For instance, we can define $f_{lower}(V_{R_{F3}}, x_D) = -x_D(\nu)$ and $f_{upper}(V_{R_{F3}}, x_D) = +x_D(\nu)$. In such manner the concentrations from the ODEs could be used as flux constraints because they represent the availability of the metabolites in the environment and therefore represent the uptake limit. Usually, the upper constraint is set only if a growth limit has to be defined, otherwise it should be infinity.

Synchronization. Since we are implementing a hybrid model technique which combines two different algorithms for solving differential and algebraic systems, we have to define rules to synchronize these techniques. In particular, we define the ODEs system as the model leading the simulation, while the LPP characterizing the FBA is exploited to calculate the value of some components of the ODEs system. Thus, according to this, we have to decide how many times and under which hypothesis the LPP has to be solved, and in this manner the respective components in the ODEs system updated.

In [37] the authors proposed a new type of DFBA, called integrated FBA, in which they integrated the FBA metabolic network with a Boolean transcriptional regulatory network and an ODE model. In particular, their simulations are characterized by a series of consecutive numerical integration of the ODE model and solution of the LPP, whose interaction depends on some common metabolites and variables. In this contest they suggest to choose the length of each time step to be large enough that the FBA assumption (the concentrations of internal metabolites are time-invariant) holds, and yet small enough for the ODE model to calculate the system dynamics without accumulating numerical error.
Starting from this consideration, we defined a more general way to call the LPP during the ODEs solution without stopping it. In details, we decided to solve the LPP only if the differences between all the input places of the general transitions modeling the FBA at time t_1 and time t_2, with $0 \leq t_1 < t_2 < t_{final}$ (t_{final} is the final time selected for the ODE solution) is greater than $\epsilon > 0$. Mathematically this can be expressed as follows (starting from the Eq.s 6.9):

$$\begin{aligned} dx_i(\nu) = &\sum_{j=1}^{n_{T_{ma}}} \phi(t_j^*, \hat{x}) S(p_i, t_j^*) d\nu + \sum_{j=1}^{n_{T_g} - n_{T_g^{FBA}}} f_{\bar{t}_j}(\nu, \hat{x}) S(p_i, \bar{t}_j) \\ &+ \sum_{j=1}^{n_{T_G^{FBA}}} \left[(1 - \mathbf{1}_{\{cond\}}) f_{\bar{t}_j}(\nu, \hat{x}^{last}) + \mathbf{1}_{\{cond\}} f_{\bar{t}_j}(\nu, \hat{x}) \right] S(p_i, \bar{t}_j) d\nu \\ &\forall i \in \{1, \ldots, n_P\}, \end{aligned} \tag{6.12}$$

where $\mathbf{1}_{\{cond\}}$ is the indicator function which is 1 if the condition expressed in *cond* is true, otherwise is 0. Such condition can be defined as follows:

$$cond = \begin{cases} TRUE & \text{if} \quad \bigvee_{x_p:\ x_p \in \hat{x}_{FBA}} |x_p^{last}(\nu') - x_p(\nu)| > \epsilon, \\ FALSE & \text{otherwise,} \end{cases} \tag{6.13}$$

where $\hat{x}_{FBA} = \{x_p : p \in {}^\bullet \mathbf{t} \ \wedge \ t \in T_g^{FBA}\}$ is the set of the markings of all the input places of the general transitions modeling the FBA, and x_p^{last} represents the marking of $x_p \in \hat{x}_{FBA}$ the last time ν' at which the FBA was calculated, with $\nu' < \nu$. Therefore, the LPP is solved only if there exists a difference (defined by ϵ) between the only variables that could change the results of the FBA, since the constraints of the flux associated with a specific general

transition $t \in T_g^{FBA}$ might depend on its input places marking. Otherwise, there would not be any important differences in the fluxes obtained from solving the FBA characterized by the same constraints (the only parameters of the FBA model which vary and depend on the ODE model). In the next section we will show an application in which we varied the value of ϵ.

6.2.2 Application Example

In this section we introduce a case study inspired by Pancreatic Ductal Adenocarcinoma (PDA) model discussed in [132] as a proof of concept of our proposed approach, which provides a biologically and mathematically grounded decision making setting for the integration of regulatory, signalling, and metabolic networks and greatly increases model interpretability and reusability.

The kinetic model is the result of a set of experimental findings obtained by Son et al. [146] and represents the main metabolic pathways altered in PDA cells. It models the pathways where the glucose and glutamine are catabolysed in order to produce energy (Adenosine TriPhosphate - ATP). ATP is the main source of energy for healthy cells and a dysregulation in its production can help cancer cells to survive and proliferate. Fig. 6.5 reports the actual ESPN model of PDA cells metabolism. It consists of 46 metabolites (that are the products of glucose and glutamine breakdown and correspond to the places of the net) and 67 chemical reactions (the transitions of the net) subdivided in two MA reactions and 65 general reactions[8]. We exploited the $\mathcal{G}reat\mathcal{M}od$ framework, which will be described in details in Chapter 7, to automatically derive the ODEs system proposed in [132] from the corresponding ESPN model (drawn using GreatSPN GUI). This clearly shows how the expressivity of the ESPN formalism is rich enough to deal with biological networks in which complex biological dynamics coexist. In details, the time required to convert the model in Fig. 6.5 into an *R file* constituted by 47 differential equations (one per metabolites plus one for the cell number) is just 0.001 *sec.* The Backward Differentiation Formula (BDF) method [18] (which is able to cope with stiffness problems), and the R package *Rglpk* (`https://CRAN.R-project.org/package=Rglpk`) are utilized to solve the ODE system and the LPP, respectively.

[8] A detailed list of all the dynamic associated with the reactions is reported in the Supplementary material of [132].

To illustrate an example of the cross-talk between PN and FBA we introduce the simple ESPN model in Fig. 6.5 (bottom-left). This model exemplifies the Oxidative Phosphorylation, a pathway where the majority of ATP is produced within the cell. In details this net consists of four transitions and seven places, and it was not explicitly considered in the kinetic model of Roi et al. [132]. This pathway uses the free energy released during the oxidation of $FADH_2$ to drive the synthesis of several ATP molecules starting from ADP and Pi molecules. In the model the transition speeds are computed by FBA assuming ATP production to be maximized. Indeed, in presence of oxigen, this pathway produces the main source of ATP (i.e., energy) in human cells.

Observe that the whole ESPN model is obtained by the superposition between places having the same names in the two models depicted in Fig. 6.5. Hence, the new places P, $FADH_2$ and FAD, are added to the original model and the speeds of the three new transitions *ATP_Synthase*, $ATP\_Synthase_2$ and *SDH_reaction* are defined solving FBA submodel. In particular, their rates were calculated solving a LPP in which the production of ATP and $mFUM$ is maximized. The time required to generate the new ODEs system is similar to the previous one, while the computational time to solve the ODEs system increases with smaller ϵ values (i.e., how many times the FBA submodel is solved considering a difference of *epsilon* between the the last and the current markings of the input places of $t \in T_g^{FBA}$) as reported in Table 6.4.

Model	**Mean time** (sec)
Original model (only ODE)	20.2
Original model + FBA ($\epsilon = 100$)	26.8
Original model + FBA ($\epsilon = 10$)	26.9
Original model + FBA ($\epsilon = 1$)	93.82
Original model + FBA ($\epsilon = 0.1$)	11608

Table 6.4: Mean solution times considering firstly the original model composed by only the ODEs system, and successively the extended model with the FBA considering different ϵ values.

Therefore, by integrating the original model with this simple net it is possible to investigate the evolution of ATP over a period of five days. In Fig. 6.6 are plotted and compared the time evolutions of ATP, ADP, ASP and OAA derived by the original model (i.e. without FBA submodel) and by the extended one (i.e. with FBA submodel) in which four ϵ values are

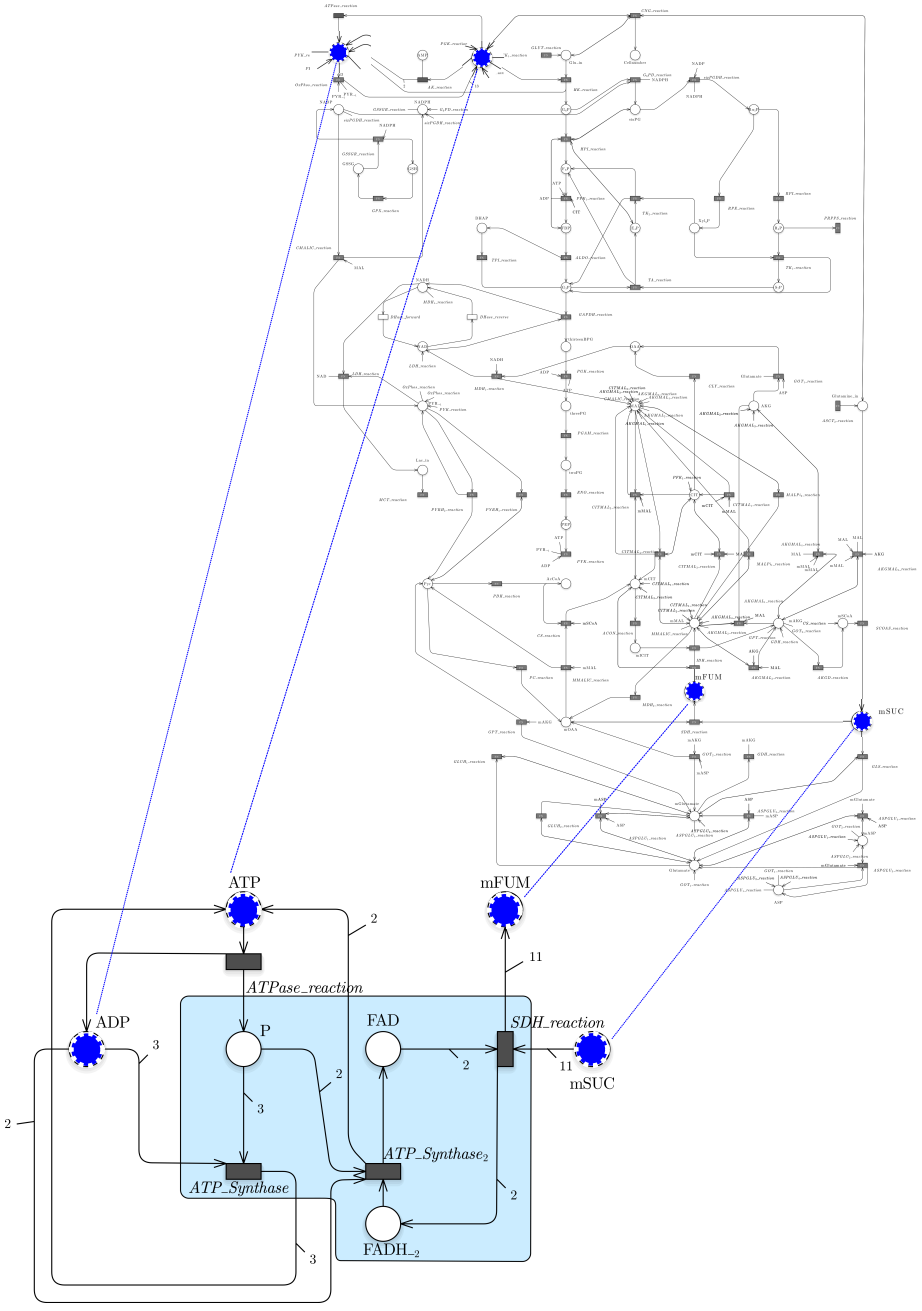


Figure 6.5: ESPN submodel describing the Oxidative Phosphorylation pathway (bottom left). At top right, the ESPN model of the PDA is reported, places shared with the this model are highlighted in dark blue. Reactions within the light blue-shaded box are computed using FBA.

considered. In details, the analysis varies the ϵ value from 0.1, representing that the FBA submodel is solved after small variations in the markings of the input places $\forall t \in T_g^{FBA}$, to 100, i.e., the FBA submodel is solved only to initialize the solution of the ODE system. In details the black continuous lines are related to the original model while the colored dashed lines to the extended model. As expected by introducing the Oxidative Phosphorylation submodel, the plots in Fig. 6.6 report an increase in ATP concentration and a decrease of ADP respectively. Moreover the other two plots in Fig. 6.6 highlights as the integration of this submodel has a global impact on the whole system behaviours. Indeed, even quantities not directly connected with FBA submodel, as *ASP* and *OAA*, are substantially affected by this extension.

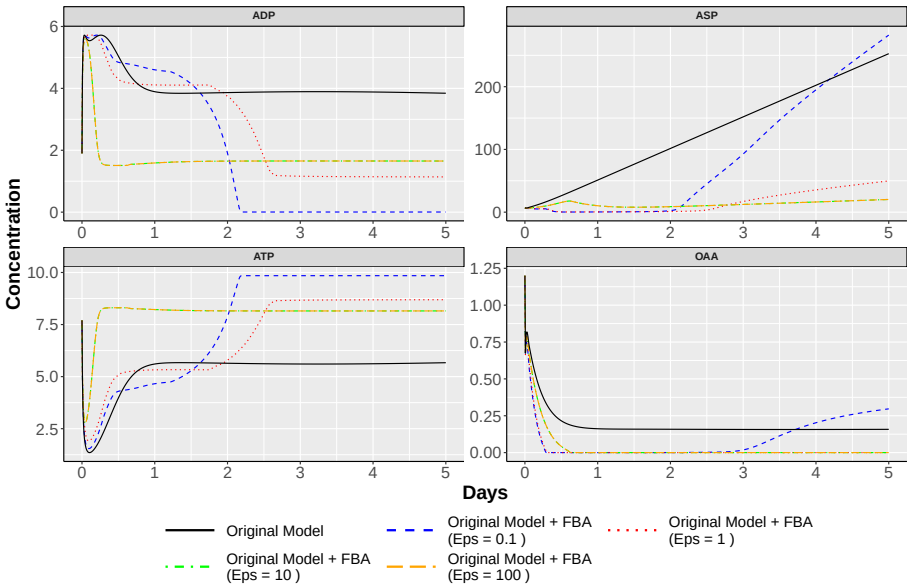


Figure 6.6: Dynamics of ATP, ADP, ASP and OAA considering both the original (solid black line) and the extended model considering different ϵ values (dashed colored lines) of the Oxidative Phosphorylation.

6.2.3 Discussion

According to this simple example, the FBA can be seen as a global source and sink of the system for a specific set of metabolites (e.g. ATP, mFUM, etc.). Clearly, this is very simple use of FBA, and in general different ways of using FBA results within an ESPN model are possible depending on the available

data. For instance, one possibility arises in the extreme case when no kinetics are available. In this situation it is possible to assume that the speeds of all transitions of the net are defined using FBA model. Therefore, in this case the ESPN formalism becomes an efficient tool to describe more easily an FBA model through a parametric and graphical formalism.
Another possibility is to identify sub-cycles in the net where kinetics information is missing and where the FBA approach can be conveniently applied. In this situation the ESPN formalism allows modeler to describe in an appealing manner the inter-playing between the system deeply detailed components with respect to those abstracted through FBA model.
Finally, a third possibility comes from the observation that a net could be an open-system as the ESPN model in Fig. 6.5 where glucose and glutamine are continuously imported into the system and lactate (the product of glucose breakdown) is exported out of it as soon as it is produced. In this last situation the interactions between the modeled system and its environment can be abstracted through an FBA submodel which allows the modeler to close the system under study.

6.3 Stochastic Simulation

In this section we described how to extend the $\tau-$leaping approach introduced in Sec. 3.2 to consider different kinetics from the Mass Action (MA) law. In details, this approach was defined under the assumption that all the reactions follow the MA law. Assumption that is exploited to compute the partial derivatives of the propensity functions, as shown in Eq. 3.8, making the algorithm dependent on the transition velocity syntax (i.e., the MA law). In general, we proposed a derivatives approximation to consider any continuous functions as kinetic (i.e., transition intensity). In such manner the $\tau-$leaping approach can be applied for the stochastic simulations considering ESPN and ESSN formalisms, in which the general transitions are defined by continuous functions. Let us note that the SSA does not need to be extended since the algorithm is independent with respect to the transition velocity syntax.
Indeed, if all the transitions within the model follow the MA law (or in general if the transitions velocity is defined with a polynomial structure), then the propensity function of the transition instance $\langle t, c\rangle$ in marking m is given

by the Eq. 2.6, that is:

$$a_t(m) = \omega(t,c) \prod_{\langle p,c' \rangle |\ p \in {}^\bullet \mathbf{t}\ \wedge\ c' \in cd(p)} m[p][c']^{I[p,t](c)[c']}, \tag{6.14}$$

where $t \in T_{ma}$, and $m[p][c']$ denotes the marking of place p for color c'. Let us note that in the case of a ESPN the propensity function is reported in Eq. 2.5, defined by a similar equation, but without the color dependencies.
Thus, the partial derivatives of Eq. 6.14 with respect to the number of tokens in a place p and color c', denoted as $\dfrac{\partial a_t(m)}{\partial m[p][c']}$, are straightforward to compute:

$$\frac{\partial a_t(m)}{\partial m[p][c']} = \frac{I[p,t](c)[c']}{m[p][c']} a_t(m), \qquad \forall p \in {}^\bullet \mathbf{t}\ \wedge\ c' \in cd(p) \tag{6.15}$$

where the number of tokens in $m[p][c']$ is always different from zero, otherwise the transition is not enabled. Let us note that the Eq. 6.15 derives from the following relation, which could be easily generalized for any formulation of the MA law:

$$\begin{aligned} a(x) &= x_1^n\ x_2^m\ c, \\ \frac{\partial a(x)}{\partial x_1} &= n\ x_1^{n-1}\ x_2^m\ c \\ &= \frac{n}{x_1} a(x), \end{aligned}$$

$\forall n,\ m > 0$, a constant $c \in \mathbb{R}$ and $x_1 \neq 0$.
Naturally, when this assumption is not true anymore, such as considering general transitions characterized by firing intensities defined as continuous functions $\{f_1, \ldots, f_h\} = \Lambda$, it is no longer possible to know in advance the propensity functions and so their partial derivatives. To overcome this problem a method of approximation of derivatives, called **Richardson's Extrapolation** [127], was exploited.
This procedure is a general method for generating high-accuracy results using low-order formulas, for instance it is possible to turn a second-order approximation of the first derivative into a fourth order approximation exploiting two second-order approximations. Where the n^{th} order approximation is a polynomial of degree n, which is obtained by truncating the Taylor series to this degree.
Therefore, we decided to exploit the following fourth-order approximation obtained from two second-order approximations:

$$f'(x) = \frac{f(x-2h) - 8f(x-h) + 8f(x+h) - f(x+2h)}{12h} + O(h^4), \tag{6.16}$$

to calculate a sixth-order approximation of the derivative:

$$f'(x) = \frac{16\ S(h) - S(2h)}{15} + O(h^6), \tag{6.17}$$

where $S(h)$ derives from Eq. 6.16 and it is defined as follows:

$$S(h) = \frac{f(x-2h) - 8f(x-h) + 8f(x+h) - f(x+2h)}{12h}. \tag{6.18}$$

Finally, we can approximate the partial derivatives of the propensity function with respect to the number of tokens in a specific colored place in Eq.s 3.8 by using the Eq. 6.17 in which the function f is defined as the propensity function of a transition $\langle t, c \rangle$ in marking m:

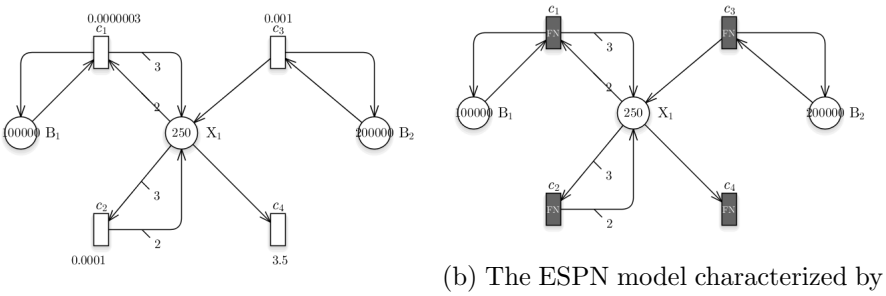
$$S(h) = \frac{a_t(m - 2h_{p,c'}) - 8a_t(m - h_{p,c'}) + 8a_t(m + h_{p,c'}) - a_t(m + 2h_{p,c'})}{12h_{p,c'}}, \tag{6.19}$$

where $h_{p,c'}$ is a small increment only on the number of tokens in the place p with color c'.

6.3.1 Application of the method to the case study

In this section we applied the $\tau-$leaping extension introduced to a reaction model famous for its bistable steady-state distribution, called the *Schlögel model* [138]. In this contest, we implemented both the SSA and $\tau-$leaping approaches in a $C++$ algorithm embedded in the *model generation* module of the modeling framework, $\mathcal{G}reat\mathcal{M}od$, which will be described in details in Chapter 7.

Therefore, we used the ESPN formalism to model the Schlögel reactions (see Fig. 6.7). In particular we defined two ESPN models, the former (Fig. 6.7a) characterized by standard transitions, which follow by definition the MA law, and the latter (Fig. 6.7b) by general transitions, which are associated with functions defined as the MA law. Therefore, we compared the results obtained from the extended $\tau-$leaping approach by varying the h value representing the increment in the derivatives approximation (from the Richardson Extrapolation) and the ϵ value (the error control parameter from the $\tau-$leaping approach). The histogram distance [21, 20] is introduced to measure the simulation error between the trajectories obtained from the two ESPN models.



(a) The ESPN model characterized by all standard transitions.

(b) The ESPN model characterized by all general transitions defined as MA law.

Figure 6.7: The Schlögel reactions model.

Schlögel model.

The Schlögel model is a quite famous example of a simple reaction network which exhibits bi-stability, i.e. the solution strongly depends on the initial conditions and parameters, converging to one of the two stable states. The model is characterized by two reactions:

$$B1 + 2X1 \underset{c_2}{\overset{c_1}{\rightleftharpoons}} 3X1,$$
$$B2 \underset{c_4}{\overset{c_3}{\rightleftharpoons}} X1,$$

where $B1$ and $B2$ denote buffered species whose respective molecular populations are assumed to remain constant over the time interval of interest, while $X1$ is the only time-varying specie. The corresponding ODEs system is

$$\begin{aligned}
\frac{dB1}{d\nu} &= -c1/2 * B1 * X1^2 + c2/6 * X1^3 \\
\frac{dX1}{d\nu} &= +c1/2 * B1 * X1^2 - c2/6 * X1^3 + c3 * B2 - c4 * X1 \\
\frac{dB2}{d\nu} &= -c3 * B2 + c4 * X1
\end{aligned} \tag{6.20}$$

where the $c1$, $c2$, $c3$ and $c4$ are constant parameters. To obtain the two stable states, the parameter values were settled as follows:

$$c1 = 3 \cdot 10^{-7}, \qquad c2 = 10^{-4}, \qquad c3 = 10^{-3}, \qquad c4 = 3.5,$$

with an initial concentrations of the species defined as $B1(0) = 1{\times}10^5$, $B2(0) = 2 \times 10^5$ and $X1(0) = 250$.

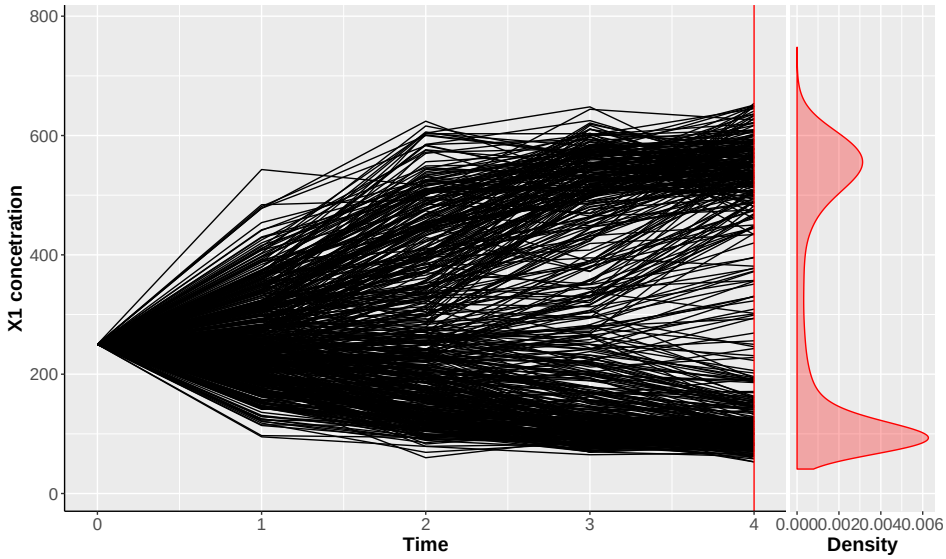


Figure 6.8: The two stable states of the Schlögel reactions model obtained exploiting the SSA method. In red the empirical probability density function at the final time, $\nu = 4$.

In Fig. 6.7a the Schlögel reactions model is represented using the ESPN formalism, where all the transitions are standard one (i.e., they follow the MA law as in Eq.s 6.20). Since our aim is to show the good approximation (introduced above) in the τ-leaping approach when continuous functions define some reactions in the model, we exploited a second ESPN model, Fig. 6.7b, in which all the transitions are treated as general transitions whose respective functions are defined as the MA law, i.e., Eq. 2.5.

Histogram Error Distance.

Let us recall some basic definitions to introduce the histogram distance for approximating the distance between the probability density functions of two random variables.
Let X be a continuous random variable, then its Cumulative Distribution Function (CDF) is defined as

$$F_X(x) = P(X \leq x) = \int_{-\infty}^{x} p_X(x)dx, \tag{6.21}$$

which can be expressed as integral of its Probability Density Function (PDF) $p_X(x)$. In many practical problems, it is not possible to obtain an analytic

distribution, thus we can consider instead the Empirical Cumulative Distribution Function (Empirical Cumulative Distribution Function (ECDF)) to approximate the CDF, and the histogram function to measure the PDF. So, let $x_1,\ x_2, \ldots,\ x_N$ be independent realizations of X, then the ECDF of X is defined as

$$F_N(x) = \frac{1}{N} \sum_{i=1}^{N} \psi(x - x_i) \tag{6.22}$$

where the function ψ is called the sign function and it is defined as follows:

$$\psi(x) = \begin{cases} 1, & x \leq 0 \\ 0, & x > 0 \end{cases} .$$

Therefore, the sum in Eq. 6.22 gives the number of points that are smaller than x. When it is divided by N, we obtain the fraction of points smaller than x, which approximates the CDF.
Supposing that all the sample values are bounded in the interval $I = [x_{min},\ x_{max})$, with $L = x_{max} - x_{min}$, then we can divide I into K subintervals $I_i = [x_{min} + \frac{(i-1)L}{K},\ x_{max} + \frac{i\ L}{K})$. Thus, p_X can be approximated by the histogram function h_X computed from

$$h_X(I_i) = \frac{L}{KN} \sum_{j=1}^{N} \chi(x_j, I_i) \tag{6.23}$$

where $\chi(x, I_i)$ is the characteristic function, which denotes if x belongs to the interval I_i,

$$\chi(x, I_i) = \begin{cases} 1, & x \in I_i \\ 0, & \textit{otherwise} \end{cases} .$$

The sum in Eq. 6.23 gives the number of points falling into the interval I_i, and when this sum is divided by N, the fraction of the points inside that interval is obtained, which approximates the probability of a sample point lying inside that interval. We divide this by the interval length, $\frac{L}{K}$, to approximate the probability density. Thus, $h_X(I_i)$ measures the average density function of X in the interval I_i.

Let us consider two continuous random variables, X and Y, which have probability density functions p_X and p_Y, respectively. Then, the *density distance* between X and Y is defined as follows:

$$D(X,Y) = \int |p_X(s) - p_Y(s)| ds. \tag{6.24}$$

This measure can be approximated by the *histogram distance* by substituting 6.23 in 6.24, obtaining:

$$\begin{aligned} D_K(X,Y) &= \sum_{i=1}^{K} \frac{|h_X(I_i) - h_Y(I_i)|L}{K} \\ &= \sum_{i=1}^{K} |\frac{\sum_{j=1}^{N} \chi(x_j, I_i)}{N} - \frac{\sum_{j=1}^{M} \chi(y_j, I_i)}{M}|, \end{aligned} \tag{6.25}$$

where $x_1,\ x_2, \ldots,\ x_N$ and $y_1,\ y_2, \ldots,\ y_M$ are the two groups of realization of the variables X and Y of size N and M respectively.
Indeed, $D_K(X,Y)$ varies depending on the value of K (the number of bins of the histogram). When K becomes larger we obtain more detailed information about the difference, and $D_K(X,Y)$ will increase, but we must generate a large number of samples, otherwise there will not be enough data falling into each subinterval and there will be a large measurement error. In particular with K, N and M sufficiently large, the histogram distance $D_K(X,Y)$ is close to the density distance area $D(X,Y)$:

$$D_K(X,Y) \longrightarrow D(X,Y) \qquad \text{as} \qquad N,\ M,\ K \longrightarrow \infty.$$

Simulations.

We used the histogram distance error measure defined in Eq. 6.25 to compare the simulations obtained exploiting the extended version of the $\tau-$leaping approach applied to the stochastic process generated from the ESPN model with all general transitions (depicted in Fig. 6.7b) and the SSA approach considering the ESPN model with standard transitions (in Fig. 6.7a). In particular we analysed the results obtained by varying the h value from the Richardson Extrapolation[9] and the error control parameter ϵ (see Eq. 3.6) value, considering 10^5 simulation runs using the initial state and the constant parameters reported in the previous paragraph.

[9]For simplicity we considered an increment value h equal for each $h_{p,c'}$ with $\langle p, c' \rangle$ is the place instance respectively to that increment.

Therefore, we firstly compared the SSA and the $\tau-$leaping approaches considering the ESPN model characterized by all standard transitions, to have base values to evaluate the approximation introduced in the τ-leaping algorithm to take in account continuous functions. Thus, in this case the simulations approaches are the ones described in Chapter 3, and the only varying parameter is ϵ. The plot with red title box in Fig. 6.9 shows the histogram error calculated at time $\nu = 4$ as a function of ϵ. It is straightforward to see that the errors increase roughly linearly with ϵ, reproducing the results reported in [21]. Differently, the other plots in Fig. 6.9 show the distance error between the simulations obtained from the SSA applied to the ESPN model with standard transitions and the corresponding extended version of the τ-leaping approach applied to the second ESPN model, in which all the transitions are treated as general (with associated function the MA law). In such manner we are able to compare the results. Finally, Fig. 6.9 shows that with $h \leq 10^{-4}$ the error varying ϵ is similar to the referring case (the red box), otherwise we have that the error derived from the control parameter ϵ is neglected by the error derived from the derivatives approximation using the Richardson's Extrapolation.

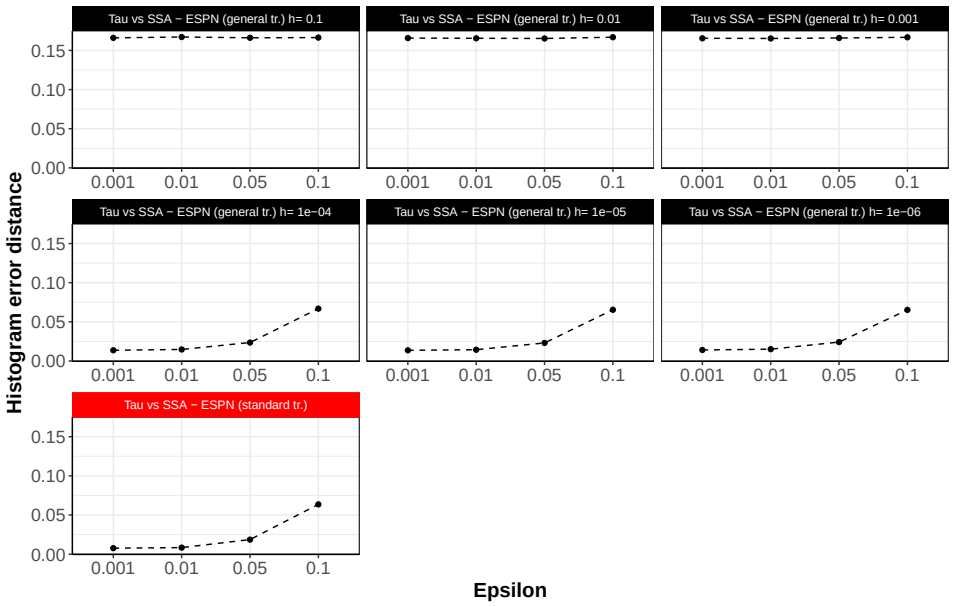


Figure 6.9: Plots of histogram distance errors with $K = 100$ corresponding to different ϵ values for the Schlögl model. Histogram distance errors are measured at the $\nu = 4$ considering 10^5 samples generated by the SSA method and the τ-leaping approach, respectively, using different h and ϵ values.

Part III

Applications and tool implementation

Chapter 7

$\mathcal{G}reat\mathcal{M}od$

In this chapter we described in detail $\mathcal{G}reat\mathcal{M}od$[1], a new framework that we developed during my PhD taking into account all the theoretical results presented in the first part of this thesis. In particular, $\mathcal{G}reat\mathcal{M}od$'s novelties and strengths can be summarized as follows: (1) the use of a graphical formalism based on Petri Nets (i.e., ESPN and ESSN) to simplify model construction and to provide an intuitive description of system behaviour; (2) the containerization (into Docker images) of all the implemented analysis techniques to improve the framework portability and to ensure the reproducibility of the derived results; (3) the implementation of an R package to provide a user-friendly interface for easily accessing to the analysis techniques[2] also for users without advanced mathematical and computational skills; (4) the specification of a well-defined schema and related infrastructure to allow users to integrate their own analysis workflows in the framework.
Thus, Section 7.1 focuses on the architecture of the framework. While, the effectiveness of $\mathcal{G}reat\mathcal{M}od$ framework is shown through the following three case studies described in the next chapter: 1) Italian Pertussis vaccination policy, 2) Multiple Sclerosis disease, and 3) COVID-19 Epidemic in Piedmont.

[1] Our framework is available to the following web page: `https://qbioturin.github.io/epimod/`

[2] including the one proposed in the first parts of this thesis.

1. **Italian Pertussis vaccination policy**. This work was published in [27] and it was exploited as a step by step guide to use the framework and to point out that $\mathcal{G}reat\mathcal{M}od$ can be easily used to develop an efficient workflow to analyse very complex systems.

2. **Multiple Sclerosis disease**. Applying $\mathcal{G}reat\mathcal{M}od$ to model the MS, we were able to identify the key parameters involving in the modulation of the effect of a specific therapy, and to perform *in silico* experiments helping to improve the understanding of this complex disease. Furthermore we show how $\mathcal{G}reat\mathcal{M}od$ can be applied to model several scenarios considering woman pregnancy [116], different treatments [120, 116, 114, 117], and space dependency [118].

3. **COVID-19 Epidemic in Piedmont**. In this work [119] $\mathcal{G}reat\mathcal{M}od$ was exploited to calibrate the model considering the surveillance Piedmont data available at the website of the Italian Ministry of Health / Civil Protection from February 24^{th} to May 2^{nd}, and successively to simulate the evolution of infected and deceased individuals under different scenarios. In particular, we showed how the control measures have proven effective in containing the epidemic, limiting the potential dangerous impact of a large proportion of undetected cases.

7.1 Framework

The architecture of this framework is composed of three main modules (see Fig. 7.1). The first module consists of a Java Graphic User Interface (GUI) based on Java Swing Class which allows to draw models using the PN formalism. This graphical editor is part of **GreatSPN** [6, 5], a software suite for modeling and analyzing complex systems using the PN formalism and its extensions. In particular, for the purposes of the framework, the **GreatSPN GUI** has been upgraded to support the ESPN and ESSN, which enable users to define a system in a compact and parametric manner, and to specify in a natural way the rate functions which may be associated with the model events, see Chapter 5.
The other two modules, consisting of an R library, called `Epimod`, and a set of docker images, implement all the framework functionalities needed for the model analysis (including the new ones proposed in this thesis). The Docker

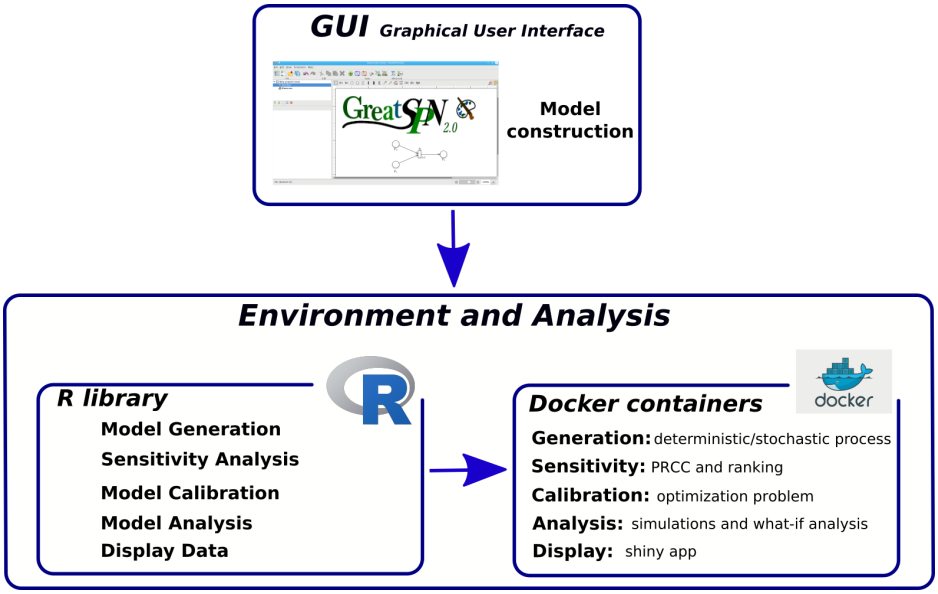


Figure 7.1: $\mathcal{G}reat\mathcal{M}od$ framework schema depicting its modules and its functionalities from a user point of view.

containerization, a *lightweight Operation System (OS)-level virtualization*, is exploited to simplify the distribution, the utilization and the maintenance of the analysis tools; the R library provides an easier user interface for which no knowledge on the docker commands is needed. Notice that all these docker images and R functions were created following the guidelines specified by Reproducible Bioinformatics Project (RBP) project to achieve a framework for developing reproducible workflow of analysis [82]. Container technology was recently proposed in the area of Bioinformatics as an efficient solution to simplify the distribution, the usage and the maintenance of bioinformatics software [38]. Indeed, exploiting containerization the users have not to deal with dependency or compilation problems; since applications and their dependencies are already packaged and installed together into the container image. Obviously, this simplifies considerably the installation and the usage of the applications encapsulated into a container image. Among the container platforms proposed in literature, Docker (`http://www.docker.com`) is getting actually the standard environment to quickly build, deploy, scale and manage containerized applications under Linux. In summary docker strengths are its high level of portability, which allows users to easily register and share containers over different hosts, and to achieve a more effective resource use and

a faster deployment compared with other similar software.

We now briefly describe all the functions implemented in the R library, `Epimod` and their associated docker images. We report in the supplementary appendix, section A.1, the instructions to install the framework.

7.1.1 The `Epimod` package

`Epimod` provides five base functions which can be combined to build a pipeline of analysis specific for the system under study:

1. `model_generation()`: derives the deterministic and stochastic processes underlying the PN model from its graphical representation;
2. `sensitivity_analysis()`: analyses the unknown parameters to identify which have a great impact on the model behaviour;
3. `model_calibration()`: finds the parameters configuration which matches better a given reference data;
4. `model_analysis()`: provides the solution and/or simulation of the underlying processes;
5. `display_data()`: offers a web application developed in Shiny, providing an interface for data visualization.

Let us now explain the details behind each introduced function.

Model generation. The generation of the stochastic and deterministic processes underlying a PN model is implemented by the R function: ***model_generation()***. In details, starting from the GreatSPN GUI is possible to draw the PN model (using whichever formalisms among the SPN, SSN, and their extensions ESPN, ESSN) and save it in a file with *PNPRO* extension, the function automatically derives from *PNPRO* file the corresponding deterministic and stochastic processes using the C/C++ program *PN2ODE* embedded in the docker image *greatspn*. The derived processes and the library used to simulate them are packaged into a binary file with *.solver* extension. In the literature many algorithms are proposed for the numerical solution of ODEs systems and for numerically generating time trajectories of a stochastic process. Obviously, each method has its strengths and weaknesses, and for

these reasons we decided to integrate more than one algorithm in our framework, which are contained in the *.solver* file and can be exploited in all the `Epimod` functions.
Considering the numerical solution of ODEs systems we implemented three explicit methods (i.e., *Runge-Kutta* 5^{th} order integration, *Dormand-Prince method*, and *Kutta-Merson method*) which can be efficiently used for systems without stiffness (i.e., the system solution is numerically stable) [18]. Instead for systems with stiffness we provided a Backward Differentiation Formula (BDF) method [18] that we implemented using the C++ LSODA library (`https://en.smath.com/view/lsoda`). For the simulation of the stochastic process, we implemented the *Stochastic Simulation Algorithm* SSA [57], the *τ-leaping algorithm* [58] (introduced in Section 3.1), and our respective extended version (introduced in Section 6.3). Finally, we proposed a further experimental module in this function, which is still not completely integrated in the framework, to automatically recognize and exploit the model symmetries (based on the theoretical results of Section 6.1).

Sensitivity analysis. The R function ***sensitivity_analysis()*** implements the sensitivity analysis starting from the *.solver* file generated by the ***model_generation*** function. Sensitivity analysis is a well-known approach exploited in computational modeling to investigate which parameters affect mostly the variability of the outcomes generated by the model. In the literature several approaches are proposed to achieve this task, such as Pearson correlation coefficient (CC) method (for linear relationships), Partial Rank Correlation Coefficient (Partial Rank Correlation Coefficient (PRCC)) method (for non-linear and monotonic relationships), and Fourier Amplitude Sensitivity Test (FAST) method (for any non-linear relationships) [98, 134]. Thus, this R function calls the R script *sensitivity.mngr.R* encapsulated into the docker image *epimod_sensitivity* to compute a sampling-based method which combines Monte Carlo Sampling (MCS) with PRCC index.

In details MCS is exploited to generate the samples of the model input variables from a random sequence of numbers with a prescribed probability distribution (generally the uniform distribution). Then the model is run N times on a fixed temporal interval: one for each generated input variable sample combination. Finally, PRCC between the generated input variables and the obtained model outputs are evaluated on the same chosen interval. In

this way the PRCC analysis and corresponding significance tests (i.e significant p-value) are utilized to identify key model parameters and to select time points which need an additional in-depth investigation. Specifically, PRCC values close to 1 (resp. -1) identify positive (resp. negative) monotone relationships between inputs and outputs; while the significance tests allow to discover those correlations that are important, despite having relatively small PRCC values.

Model calibration. The model calibration is performed by the R function ***model_calibration()***. This function executes the R script *calibration.mngr.R* embedded in the docker image *epimod_calibration* that calls the right solvers according to the passed input parameter and produces as output a textual file in which all the generated parameter values are ranked according to their ability to fit the real data (i.e., from the best data fitting to the worst one). This is obtained solving an optimization problem in which the input objective function is minimized. In Computer Science, Mathematics, and Operations Research, optimization or mathematical programming consists of minimizing (or maximizing) a function by consistently selecting the values of its variables from a set of feasible possibilities utilizing analytical or numerical methods. Formally an Optimization Problem (OP) with inequality constrains can be defined as follows:

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & \mathcal{F}_{opt}(\mathbf{x}) \\ \text{subject to} \quad & \mathcal{G}_i(\mathbf{x}) \geq b_i, \quad 1 \leq i \leq l \\ & \mathcal{L}_i(\mathbf{x}) \leq c_j, \quad 1 \leq j \leq m \end{aligned}$$

where the vector $\mathbf{x} = (y_1, \ldots, y_n)$ is the *variable vector*, the function $\mathcal{F}_{opt} : \mathbb{R}^n \to \mathbb{R}$ is the *objective function*, the functions $\mathcal{G}_i(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$ and $\mathcal{L}_i(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$ are *inequality constraint functions*, and the constants $b_1, \ldots, b_l$, $c_1, \ldots, c_m$ are the *bounds* for the constraints. A vector $\mathbf{x}^\bullet$, called *optimal*, is the solution of the OP if, among all vectors that satisfy the constraints, it is that which yields the smallest (largest) value of the optimization function: $\forall \mathbf{z}$ s.t. $\mathcal{G}_1(\mathbf{z}) \geq b_1, \ldots, \mathcal{L}_1(\mathbf{z}) \leq c_m$ we have that $\mathcal{F}_{opt}(\mathbf{z}) \geq \mathcal{F}_{opt}(\mathbf{x}^\bullet)$.

OP is termed a *linear program* if the objective and constraint functions are linear and *non-linear* otherwise. In our framework, the focus is on non-linear programs in which constraints can be non-linear as well. To solve this type of OPs, several algorithms have been proposed in the literature, an overview on these methods is reported in [85]. Among the available algorithms, the

one integrated in our framework is the Generalized Simulated Annealing for Global Optimization implemented in the R package GenSA [176], since it was designed to solve complicated nonlinear objective functions with a large number of local minima.

Model analysis. The R function ***model_analysis()*** solves the model and generates an output representing the time evolution of the model. The R script *model.mngr.R* embedded in the docker image *epimod_model* is then executed by ***model_analysis()*** function. Thus, this script simulates the underlying deterministic or stochastic process, given a fixed parameters configuration, and returns a textual file in which the system solution is provided.

Display data. To help the user in both experimentation and analysis of the model, our workflow encompasses a data visualization function. Specifically, the function ***display_data()*** offers a web application developed in Shiny providing a basic-level interface and an expert-level interface for data visualization. The basic-level interface consists of a simple visualization environment, so that the user can directly focus on analyzing and visualizing the results with just few clicks rather than spending its efforts setting up the necessary environment for the visualization. On the other hand, when the basic level interface is not enough to highlight complex behaviours of the system under study, then the expert-level interface of *display_data* can be exploited. Indeed it allows the user to implement its own visualization plots. In this case, the user has to provide a function describing how the output data derived by analysis phase must be manipulated to be plotted. Hence, this functionality makes the data visualization very flexible, since it supports all the functionalities of *ggplot2* [174] R library.

How to integrate a new function in the framework.

The customization of the framework is one of its strengths since it provides the generalizations needed to use this same framework for completely different studies. For instance, in this thesis we applied the framework to analyse a molecular biological system by modeling the MS disease (Section 8.2), and two epidemiological systems considering the Pertussis disease (Section 8.1) and the COVID-19 disease (Section 8.3); but in general $\mathcal{G}reat\mathcal{M}od$ could be exploited to model other biological systems other those described in this thesis. To

this aim we described in this subsection how new solution functionalities can be easily added in the framework. Practically, a user must firstly embed the new tool into a docker images following the tutorial reported at `http://www.reproducible-bioinformatics.org/` in the section *"How to be part of the Reproducible Bioinformatics project"*. Secondly, an R function implementing an interface for the created docker images has to be provided. To simplify the creation of such controlling function the R function *skeleton.R*, reported in the library, can be exploited as prototype. Then, any new R function and associated docker image must always be supported by an explanatory vignette, accessible online as html document, and by a set of test data accessible online as well. Finally, this new R function and associated docker image must be submitted to the `info@reproducible-bioinformatics.org` so that the RBP core team verifies the compliance of the new functionalities with the RBP guidelines. In our case, this protocol means that, once the framework has been certified by the RBP core team, every new addition or improvement must first be verified by the RBP organization before integrating it into the framework. More details on this task can be found in [82].

Chapter 8

Applications

In this chapter we show three different case studies successfully investigated using $\mathcal{G}reat\mathcal{M}od$.

8.1 Pertussis and its vaccination policy in Italy

In this section we described how the $\mathcal{G}reat\mathcal{M}od$ framework was exploited to study the Pertussis infection and its vaccination cycle in Italy. We first introduced the problem in Section 8.1.1, and then we showed in Section 8.1.2 how a model for this complex system can be created. In Section 8.1.3 we described the use of the functions and feature implemented in $\mathcal{G}reat\mathcal{M}od$ to reproduce real data coming from the observation of the spread of Pertussis in Italy during the period from 1974 to 2016. Moreover, we demonstrated that our framework can be easily exploited to support a what-if analysis on the model representing this complex system.

8.1.1 The disease and its vaccination policy

Pertussis, also known as whooping cough, is a highly contagious infectious disease caused by the bacterium Bordetella Pertussis which colonizes the ciliated cells of the respiratory mucosa. It provokes an uncontrollable coughing which often makes breathing hard and which can possibly lead to serious com-

plications including death. The first vaccine against Pertussis was developed already in the 1930s by pediatrician Leila Denmark. Despite this, Pertussis remains a challenging public health problem because many aspects of its infection, disease, and immunity are not completely understood yet.
Although the implementation of Pertussis vaccination programs in many countries has decreased substantially its diffusion and mortality, Pertussis has not been eliminated and Pertussis-related hospital admissions and fatalities are still evident, particularly in young infants [104].
Moreover, the European Centre for Disease Prevention and Control (ECDC) in its annual 2017 report [51] highlighted an increasing trend of Pertussis cases in EU, probably due to the decrease in vaccine effectiveness over time and pathogen adaptation [81, 104, 140]. In this context computational modeling can play an important role in providing insights on the drivers of Pertussis epidemiology, in investigating alternative explanations of the observed resurgence and in predicting potential effects of different vaccination strategies.
To these aims, several models were proposed in the literature since 1980s; for instance in [70, 69], an age-structured model is exploited to analyse the possible effects of adopting different vaccination strategies in Australia. Other models expressed in terms of systems of differential equations are used to explain the duration of the Pertussis natural immunity [167], or the importance of age-structured contacts [128]. Differently in [156], a set of Partial Differential Equation (PDE)s, characterized by age and time dependent variables, is proposed to study the vaccination related changes that may have occurred for the pertussis epidemic in the Netherlands from 1996 to 1997. In [94] it is shown that a stochastic process can be used to better capture Pertussis vaccination behaviour, as well as the nature and degree of protection provided by the acellular Pertussis (aP) vaccine. Similarly, in [40, 15] a stochastic process modeling Pertussis vaccination is presented for the analysis of the disease effect in different countries, respectively Massachusetts (United States) and Thailand. However, all of these works address only a subset of the specific peculiarities of the pertussis disease. In [19] the authors report the necessity of incorporating into a single model more details of the disease (e.g., the population age, the individual immunization level, ...) to better match the real observed dynamics and to predict the outcome of vaccination measures [19].

8.1.2 The model

The many aspects of the Pertussis disease and of the vaccination strategies can be conveniently represented by extending the classical Susceptible - Infectious - Recovered - Susceptible (SIRS) model. In particular, this new model considers a population in which each individual is described by their age (i.e., newborn, young, or adult), their level of immunization (i.e., resistance level), their vaccination status (i.e., how many doses were administered) and their infectious status (i.e., susceptible, infected, and recovered). The main system events are: the infection of a susceptible individual due to a contact with an infected one, the vaccination of an individual involving the administration of vaccine doses at different time points, and the recovering of an infected individual.

To keep under control the complexity of this phenomenon, the ESSN formalism is used, and in Fig. 8.1 the developed model is showed. It consists of eight places and 30 transitions, and it is organized in four modules highlighted through colored boxes.

In details, places *BirthCount*, *VacCount*, and *InfectedCount* are introduced to count the total number of births, vaccinations, and infections happened during the system simulation. Hence, these places have a neutral domain and are introduced to make easier the computation of the measures of interest (e.g. the number of infected individuals in each year).

Places S, V, Ip, Is, and R encode the possible infectious status in which a population member may be (i.e., *Susceptible*, *Vaccinated*, *Infected due to primary infection*, *Infected due to repeated infection*, and *Recovered* respectively).

It is worth noting that the *Infected* state is modeled with two places to distinguish between individuals that are experiencing a primary infection (Ip) and those experiencing a repeated infection (Is). This distinction is important because primary and repeated infections have different characteristics [167].

The number of tokens in these places denotes the number of population members that are *Susceptible*, *Infected*, *Vaccinated*, and *Recovered* at any point in time, during the evolution of the system represented by the model. Moreover, each token (i.e., individual) in these places is labelled with the age, the level of immunization, and the vaccination status to better characterise each individual in the system. This is carried out defining the following three color classes:

- The class $A = \{a_1, a_2, a_3\}$ records the age of the population. It is divided in three static subclasses: $N = \{a_1\}$ representing Newborn individuals (from $0 \sim 11$ months), $Y = \{a_2\}$ representing Young individuals (11 months $\sim$ 18 years:), and $O = \{a_3\}$ representing all the others ($18 \sim 99^+$ years).

- The class $V = \{v_0, v_1, \ldots, v_5\}$ represents how many vaccination doses were currently received. Since the Italian vaccination policy establishes three doses within the first 11 months of life followed by two additional boosters between 12 and 18 years of age, then we accordingly split this class in six static subclasses (i.e., $NV = \{v_0\}$ no vaccination, $V1 = \{v_1\}$ first vaccination, $\ldots V5 = \{v5\}$ fifth vaccination). The vaccine coverage data were extracted from [171] and [62]. Let us note that the under-reporting is not considered in this analysis.

- The class $L = \{l_0, \ldots, l_3\}$ represents the level of protection against the infection. It is divided into four static subclasses (i.e., $L_0 = \{l_0\}, \ldots, L_3 = \{l_3\}$) encoding an increasing level of resistance.

The color domain associated with these places is defined by the Cartesian product $A \times V \times L$. Moreover the transitions *GrowthS*, *GrowthIp*, *GrowthIs*, *GrowthR*, *GrowthV*, *RecRecall*, *RecoveryIp*, *LevDecreasingR* and *LevDecreasingV* are standard transitions (i.e following MA law) while all the others are general transitions (i.e. whose rates are defined as continuous functions).
Let us note that the contact matrix depending on the three age ranges (N, Y and O) was estimated from that provided by [107], in which the Italian contact rates are reported assuming the population divided into 15 age ranges. While from the Italian Ministry of Health [172] we obtained the Italian population size, annual numbers of live births and deaths from 1974 to 2016. According to this we defined i) the initial marking of the system, and ii) the birth and death rates as the average number of births and deaths, respectively, per day in each age class during the reference period. Moreover, from [62] and the surveillance data, we were able to estimate the number of infects in each age class in the 1974 for initializing the analysis. In general, all the constants, the numerical values, the generic functions associated with these transitions, and how we defined the initial marking are described in detail in the Appendix, Section A.2. The four modules corresponding to the four infectious status of an individual are now described.

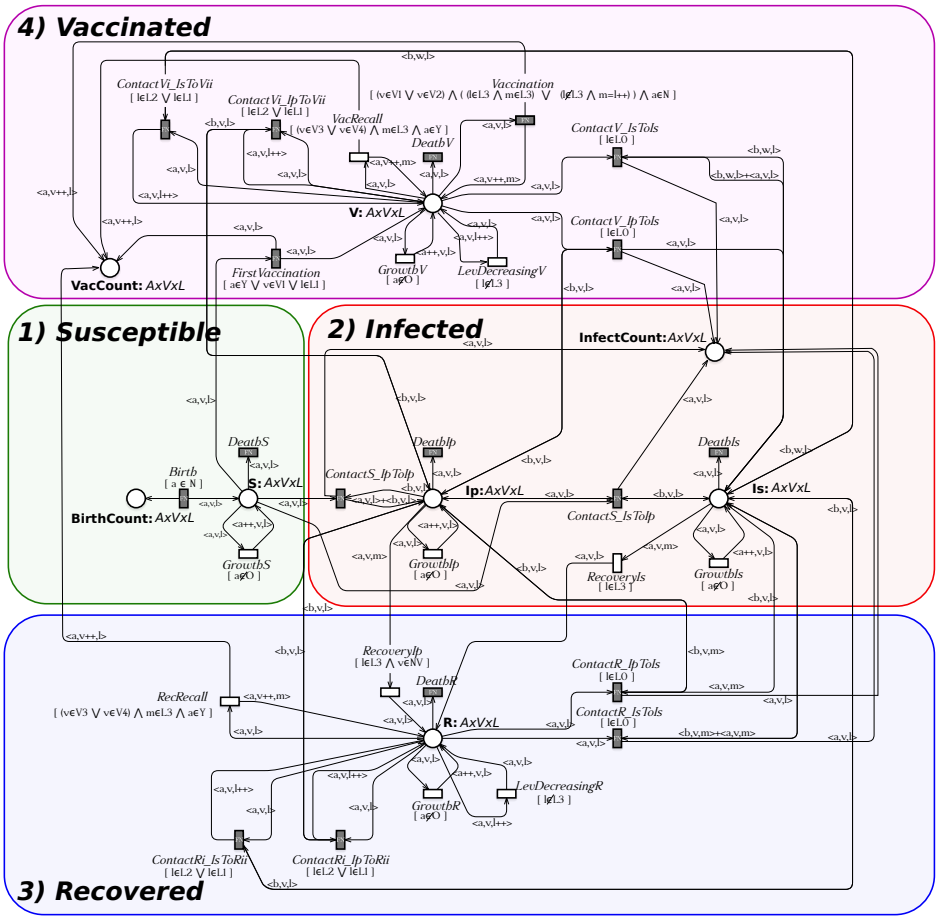


Figure 8.1: ESSN model developed for studying Pertussis epidemiology and vaccination in Italy. It is divided in four sub-models representing the possible infectious status in which a person might be: *Susceptible*, *Vaccinated*, *Infected*, and *Recovered*

1) Supsceptible module. It describes the behaviour of susceptible individuals. Transition *Birth* models the birth of a new person adding a new token in places *BirthCount* and S. Since a newborn enters into the system with the lowest level of resistance and without vaccination then the token added in place S is $\langle a_1, v_0, l_0 \rangle$. Differently, the age growth and the death of a susceptible individual are modeled by transitions *GrowthS* and *DeathS* respectively. Observe that the successor operator (i.e., $s{+}{+}$) in the arc function labeling the output arc connecting *GrowthS* to S is used to represent the increasing of the age, while the guard $[a \notin O]$ associated with *GrowthS* guarantees that this transition is disabled when the maximum level of age (i.e., O) is reached.

2) Infected module. It models the behaviour of infected individuals. In particular, two types of infections, primary and repeated infections are considered and represented by places Ip and Is, respectively. Similarly to what done in the *Supsceptible* module, the age growth of an individual with primary (resp. repeated) infection is modeled by the transition *GrowthIp* (resp. *GrowthIs*), while the individual death is represented by the transition *DeathIp* (resp. *DeathIs*).
Transition *ContactS_IpToIp* (resp. *ContactS_IsToIp*) models the infection of a susceptible member due to a contact with one individual with primary (resp. repeated) infection. Thus its firing removes one token from S and adds it into Ip. Finally, the recovery from a primary (resp. repeated) infection is modeled by transition *RecoveryIp* (resp. *RecoveryIs*), which removes one token from the place I_p (resp. I_s) and adds it to the place R. In particular, the guards associated with these transitions (i.e., *RecoveryIp* and *RecoveryIs*) guarantee that the recovered patient has the highest level of immunity (i.e., $[l \in L_3]$).

3) Recovered module. It describes the behaviour of recovered individuals. Transition *ContactRi_IpToRii* (resp. *ContactRi_IsToRii*) models the natural booster that increases to l_3 the resistance level of a recovered with resistance level l_1 or l_2 after a contact with an individual with a primary (resp. repeated) infection. These transitions (i.e. *ContactRi_IpToRii* and *ContactRi_IsToRii*) can fire only if l belongs to L_1 or L_2, guaranteed by the guard $[l \in L_1 \,||\, l \in L_2]$.
Transition *ContactR_IpToIs* (resp. *ContactR_IsToIs*) describes the relapse of a recovered individual with the lowest resistance level (see guard $[l \in L0]$) due to the contact with a population member affected by a primary (resp. secondary) infection.
Transition *RecRecall* models the two vaccine recalls between 12 and 18 years old, which are possible only if all the previous three doses were successfully administrated during the first year of life. This is ensured by the guard $[(v \in V_3 \,||\, v \in V_4) \;\; \& \;\; m \in L_3 \;\; \& \;\; a \in Y]$, which enables the transition only if an individual is in the second age class (i.e $a \in Y$) with three (i.e $v \in V_3$) or four (i.e $v \in V_4$) vaccine doses already administrated. Thus, each administration increases the patient resistance level to its maximum (i.e. the transition guard $m \in L_3$). Moreover, each time transition *RecRecall* fires, one token is added to the place *VacCount* for counting the number of vaccine doses which have been administrated.

Transition *LevDecreasingR* represents the reduction of the resistance level. Observe that the immunization is totally lost after about 14 years [173] from the last infection. In particular, when the resistance level of an individual reaches the minimum value, i.e. $[l \in L_0]$, a recovered patient becomes again susceptible for infection. their relapse is modeled by transitions *ContactR_IpToIs* and *ContactR_IsToIs* respectively. Finally, the age growth and the death of a recovered patient are encoded by transitions *GrowthR* and *DeathR*.

4) Vaccinated module. It implements the vaccination policy. Similarly to the recovered module, transitions *ContactV_IpToIs* and *ContactV_IsToIs* model the infection process, while transitions *ContactVi_IpToRii* and *ContactVi_IsToRii* the natural booster, *GrowthV* the aging and *DeathV* the death. Differently from the recovered module, the reduction of the resistance level obtained by the vaccine is lost after about 7 years [173]. This process is modeled by the *LevDecreasingV* transition. The starting of the vaccination process is represented by transition *FirstVaccination*, whose guard guarantees that vaccination is administrated only to a susceptible child. To complete the vaccination schedule, the administrations of two further doses are modeled by the *Vaccination* transition. Its guard, defined as $[(v \in V_1 \ || \ v \in V_2) \ \& \ ((l \in L_3 \ \& \ m \in L_3)||(l \notin L_3 \ \& \ m = l{+}{+})) \ \& \ a \in N]$, guarantees that, under the condition to be in the first age class, (i.e. $a \in N$, only if the first or second vaccination is administrated) it is possible to move into the successive vaccination class, i.e. if $v \in V_1||v \in V_2$ then the output arc instance is characterized by $v{+}{+}$. Indeed, the resistance level increases, due to the new dose administration, only if the level is not already at the maximum value, i.e. $(l \in L_3 \ \& \ m \in L_3) \ || \ (l \notin L_3 \ \& \ m = l{+}{+})$.
Finally, every time that transitions *FirstVaccination*, *Vaccination*, and *VaccRecall* fire, a new token is added to the place *VaccCount*.

8.1.3 A workflow for studying the Pertussis in Italy

We now describe how the framework functions can be combined to obtain an analysis workflow for such model. This schema is summarized in Fig. 8.2 in which the light grey rectangles correspond to the four phases (i.e., *Model generation*, *Sensitivity Analysis*, *Model Calibration* and *Model Analysis*) im-

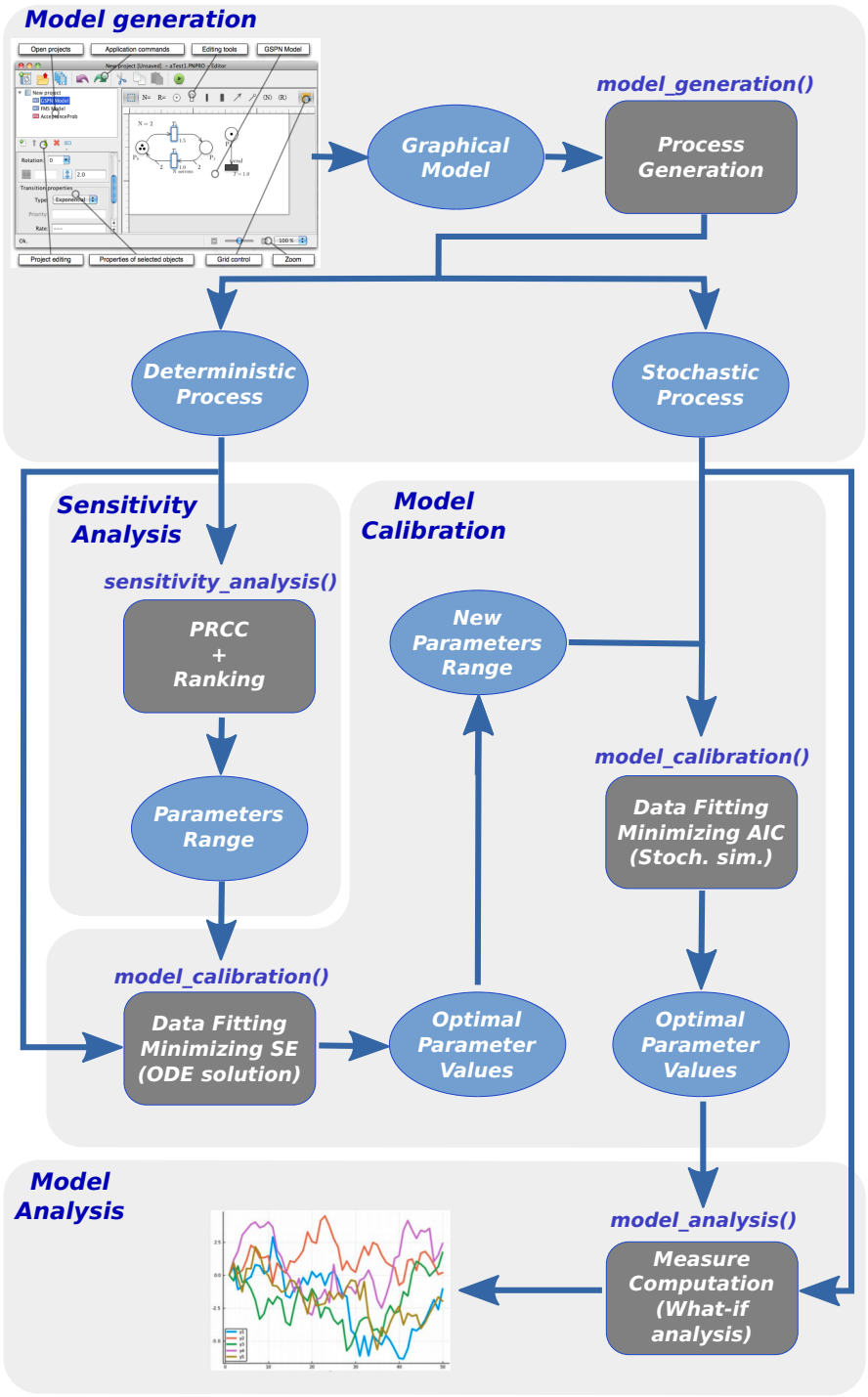


Figure 8.2: The schema of the workflow implemented for studying the ESSN model in Fig. 8.1.

plementing the analysis of our Pertussis model, while the dark grey boxes inside rectangles point out the main R framework functions exploited in each step of the analysis. The output of each task is instead highlighted by a blue circle.

Model Generation. The starting point of this workflow is the *Model Generation* phase, which derives from the Pertussis model the corresponding underlying stochastic and deterministic processes. This task can be achieved applying the R function *model_generation()* on the Pertussis ESSN model. Then the derived deterministic process is represented by a system of 179 ODEs, while the derived stochastic process is characterized by 1965 possible events. The total execution time needed to derive the two processes and to create the *.solver* file requires less than one minute on Intel Core I7 2.60Ghz. After this initial step, *Sensitivity Analysis* and *Model Calibration* are two pivotal steps to make our model consistent with real observed data, which were collected from the Italian Ministry of Health [103, 170] and Surveillance Atlas of Infectious Disease [169]. Such data report the number of Italian Pertussis cases per year from the beginning of 1974 until the end of 2016.

Sensitivity Analysis. This step allows to identify among the input parameters which are the sensitive ones (i.e., those that have a great effect on the model behaviour). This may simplify the calibration step reducing (1) the number of variables to be estimated and (2) the search space associated with each estimated parameter. In our case study, we identified 15 input parameters characterized by a high uncertainty due to their difficulty of being empirically measured. Specifically, three of them represent the probabilities of having (i) the *susceptible infection success*, i.e., the infection of a susceptible individual due to a contact with an infected individual, namely *prob_infectionS*, (ii) the *resistant infection success*, i.e., the infection of a vaccinated or recovered individual with the minimum resistance level due to a contact with an infected individual, namely *prob_infectionR_l1*, and finally (iii) *the natural boosts*, i.e., the restoring of the resistance level to the maximum when a person with resistance level different from the minimum level comes into contact with an infected individual, namely *prob_boost*.
The others 12 parameters define the proportion of susceptible and recovered individuals for each pair of age class and resistance level in the initial mark-

ing. Specifically, *init_S_a{1,2,3}* represent the initial number of susceptible individuals in each age class, while *init_R_a{1,2,3}_nv_l{1,2,3}* are the initial number of non vaccinated recovered individuals for each age class (a) and resistance level (l). Given the partial information that we have on the spreading of the infection over the Italian population at the beginning of our study (estimated from ISTAT website [103] at the beginning of 1974 decreased by the average number of infected individuals during the same year) such proportion is used to define an initial detailed situation adequate for our modeling study and compatible with the available data.
Furthermore, to provide a measure of the sensitivity of these parameters the function *sensitivity_analysis()* was applied on the deterministic process previously generated and considering the period from 1974 to 1994, when the type of vaccine was the whole-cell Pertussis (wP) vaccine. The choice of this time interval for this analysis allows to simplify our model disabling the vaccination process, since the wP vaccine era is widely considered as a good surrogate for pre-vaccine era [167].
Moreover, this model was run 64'000 times on this time interval: in every run a new input variable sample combination is generated according to the uniform distributions reported in Table 8.1, column two. Finally PRCC between the generated input variables and the obtained model outputs (using Backward Differentiation Formula method for the numerical solution of ODE system) are evaluated. A complete description of the used command line is reported in the Appendix, Section A.2.3. The execution time for this analysis is ~ 4 hours on Intel Xeon processor @ 2GHz, exploiting a parallel execution on 40 cores. The computed results are reported in Fig. 8.3 in which the PRCCs values calculated for each parameter with respect to the number of infection cases over the entire time period are showed. From this plot it is straightforward to derive that the *prob_infectionS* is the most important parameter affecting the *infects* behaviour, followed by *prob_infectionR_l1*. Differently the *prob_boost* probability and the initial number of susceptible and recovered individuals in each age class are less relevant on the infection behaviour.

In Fig. 8.4, the squared error between the real and simulated infection cases from 1974 to 1994 are plotted varying the *prob_infectionS* parameter (on the x-axis) and *prob_infectionR_l1* parameter (on the y-axis). Each point is then colored according to a linear gradient function starting from color dark blue (i.e., lower value) and moving to color light blue (i.e., higher values).

Parameter name	PRCC ranges	GENSA Init.	GENSA ranges	GENSA Output
prob_boost	$[0, 0.010]$	0.0025	$[0.0, 0.0025]$	0.002474758
prob_infectionS	$[0, 0.005]$	0.0031	$[0.0025, 0.0100]$	0.002537443
prob_infectionR_l1	$[0, 0.010]$	0.0023	$[0.0, 0.0025]$	0.002458887
init_S_a1	$[0, 866703]$	866703	$[0, 866703]$	866696
init_S_a2	$[0, 15685693$	15685693	$[0, 15685693$	15685680
init_S_a3	$[0, 37837299]$	37837299	$[0, 37837299]$	37628100
init_R_a1_nv_l4	$[0, 866703]$	0	$[0, 866703]$	7
init_R_a2_nv_l1	$[0, 15685693]$	0	$[0, 15685693]$	4
init_R_a2_nv_l2	$[0, 15685693]$	0	$[0, 15685693]$	2
init_R_a2_nv_l3	$[0, 15685693]$	0	$[0, 15685693]$	2
init_R_a2_nv_l4	$[0, 15685693]$	0	$[0, 15685693]$	2
init_R_a3_nv_l1	$[0, 37837299]$	0	$[0, 37837299]$	209184
init_R_a3_nv_l2	$[0, 37837299]$	0	$[0, 37837299]$	4
init_R_a3_nv_l3	$[0, 37837299]$	0	$[0, 37837299]$	4
init_R_a3_nv_l4	$[0, 37837299]$	0	$[0, 37837299]$	4

Table 8.1: Parameters variability range used during sensitivity and calibration analysis. In details, in the first column are listed the parameter names, then in the second and fourth columns the variability ranges used for the sensitivity and calibration analyses, respectively. The third column reports the initial parameters configuration. Finally, the fifth column is the optimal configuration discovered in the calibration analysis such that the quadratic error w.r.t. the real data is minimized. Let us remember that the population size assumed is ~ 60 Million (i.e., the Italian population of the 1974).

From this plot we can observe that higher squared errors are obtained when *prob_infectionS* assumes values greater than 0.0025 and *prob_infection_l1* values greater than 0.005, see the light blue points within the region identified by values of *prob_infectionS* $\in [0.0025, 0.005]$ and *prob_infection_l1* $\in [0.005, 0.01]$. Therefore, according to this we shrunk the search space associated with the two parameters in order to focus on the identified area.

Model Calibration. The aim of this phase is to adjust the model input parameters (e.g., *prob_infectionS*, *prob_infectionR_l1*, ...) to have the best fit of simulated behaviours to the real data. As described in above our framework implements the calibration procedure through an optimization problem which minimises a user-defined object function. Since this optimization task is computationally expensive when a stochastic process is considered, we describe now a two-steps approach to speed-up this task that can be implemented easily using our R function. The idea behind this approach is to exploit the calibration of the deterministic process, typically faster, to reduce the parameter search space in the calibration of the stochastic process.

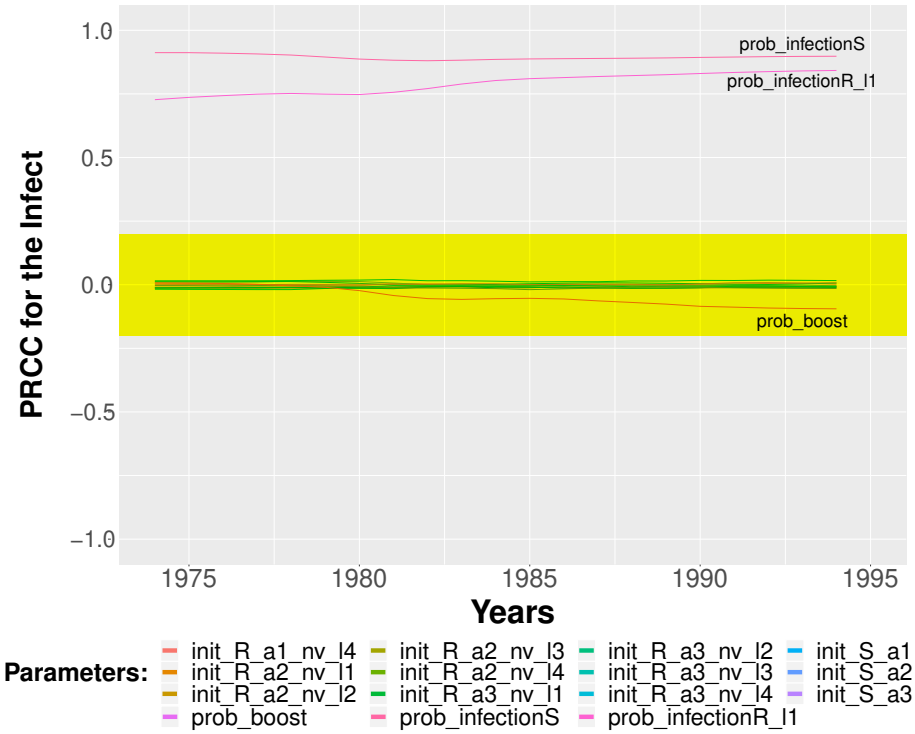


Figure 8.3: PRCCs values for the selected input parameters with respect the number of infections over the entire simulated period.

Then, in the first step the function *model_calibration()* is applied on the generated deterministic process to fit its behaviour to the real Italian infection data (from 1974 to 1994) using squared error estimator via trajectory matching, and then Generalized Simulated Annealing for Global Optimization, GenSA [176], is executed to identify the best parameter set and BDF method to solve the ODEs system. Note that the information derived by the sensitivity analysis is exploited to reduce the number of parameters to be estimated and/or their search space.

Fig. 8.5 shows a subset of all the trajectories generated by GenSA characterized by 15'000 trajectories extracted from a set of $\sim$ 90'000 trajectories obtained in $\sim$ 48 hours on an Intel Xeon processor @ 2GHz on a single core. The trajectories are colored depending on their distance (in terms of squared error) with respect to the Pertussis surveillance data (the red line). In details, the yellow color is associated with a low squared error, the purple color with a high squared error, while the optimal trajectory is showed in black. Moreover, the beam of trajectories (colored in yellow), closest to the optimal

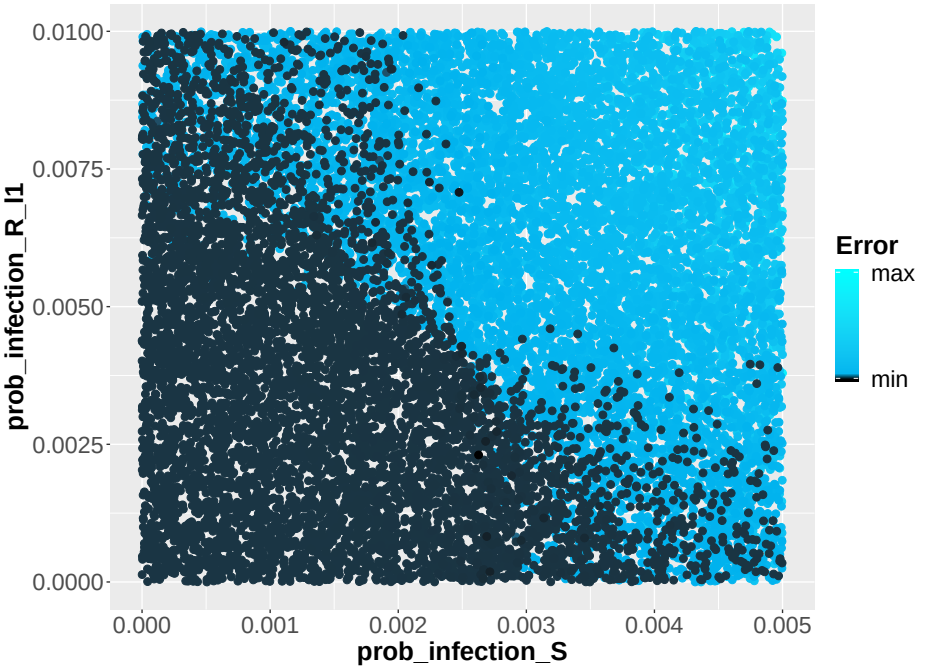


Figure 8.4: Scatter Plot showing the squared error between the real and simulated infection cases varying *prob_infectionS* and *prob_infectionR_l1*. The dark blue points represent the parameters configuration with minimum error w.r.t. the real data.

one, provides an indication on the ranges of parameter values that should be considered in the second steps of our calibration approach. In the second step, the function ***model_calibration()*** is applied on the generated stochastic process to fit its behaviour to the real infection data using Akaike Information Criterion (AIC)[3] via trajectory matching. The parameter search space of this second optimization step is then computed from the result obtained from the previous step, reported in the last column of the Tab. 8.1. Fig. 8.6 shows trajectories (grey lines) for the fifteen best parameter configurations discovered, whose range values are reported in the Tab. 8.2. The blue area contains the average trajectories derived for the first ten best parameters configuration, while the two green lines provides the associated confidence interval. We can observe that a good approximation of the surveillance data (red line) from the 1974 to 1994 is obtained. This second step required about 48 hours on Intel Xeon processor @ 2GHz, exploiting a parallel execution on 40 cores. The trajectories are generated using the extended τ-leaping algorithm (see Section

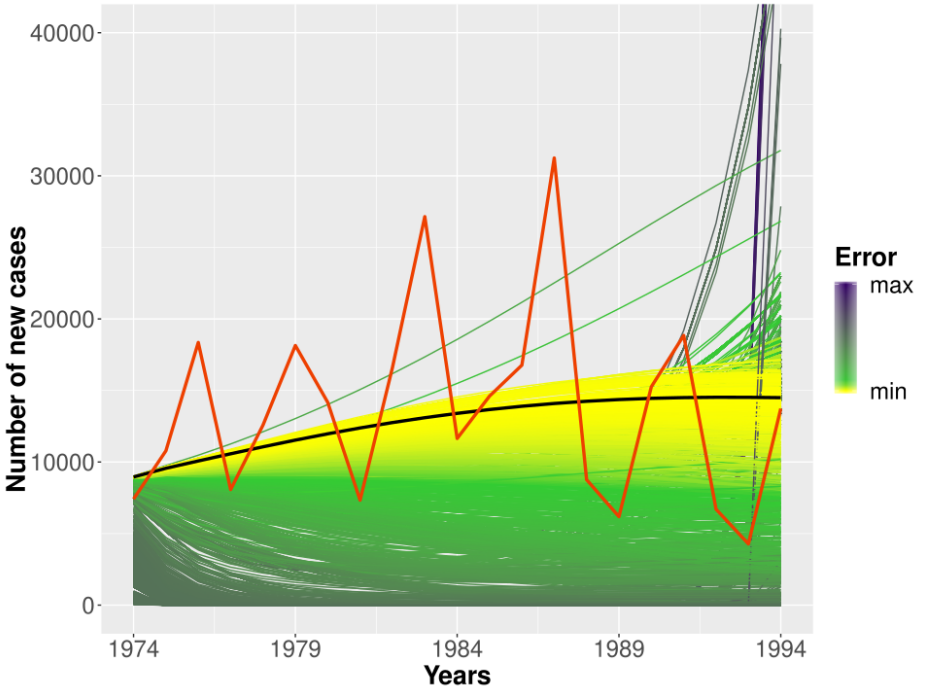


Figure 8.5: Model Calibration considering the deterministic model. It is plotted a subset of the trajectories showing the number of new cases per year obtained from GenSA considering the parameter ranges stored in the fourth column of Tab. 8.1. The color of each trajectory depends on the squared error w.r.t. the incidence of Pertussis (red line). The black line is the optimal trend obtained by minimizing the squared error.

6.3).

Model Analysis. In this last phase of our workflow the user can analyse the calibrated model to answer specific questions and to derive new insights. In our case study we show a simple what-if analysis that can be implemented tacking advantage of the R function *model_analysis()*. In particular we investigate the impact of different vaccination failure probabilities with respect to the number of infection cases. The simulated time period is from 1974 to 2016, and the pertussis vaccination program is started in 1995, with an average vaccination coverage starts from 50% and transitions linearly to 95% in 8 years, [171, 62]. The results are derived using the extended τ-leaping algorithm for generating 1024 trajectories for each case. The simulation of each case has required 4 hours on Intel Xeon processor @ 2GHz, exploiting a

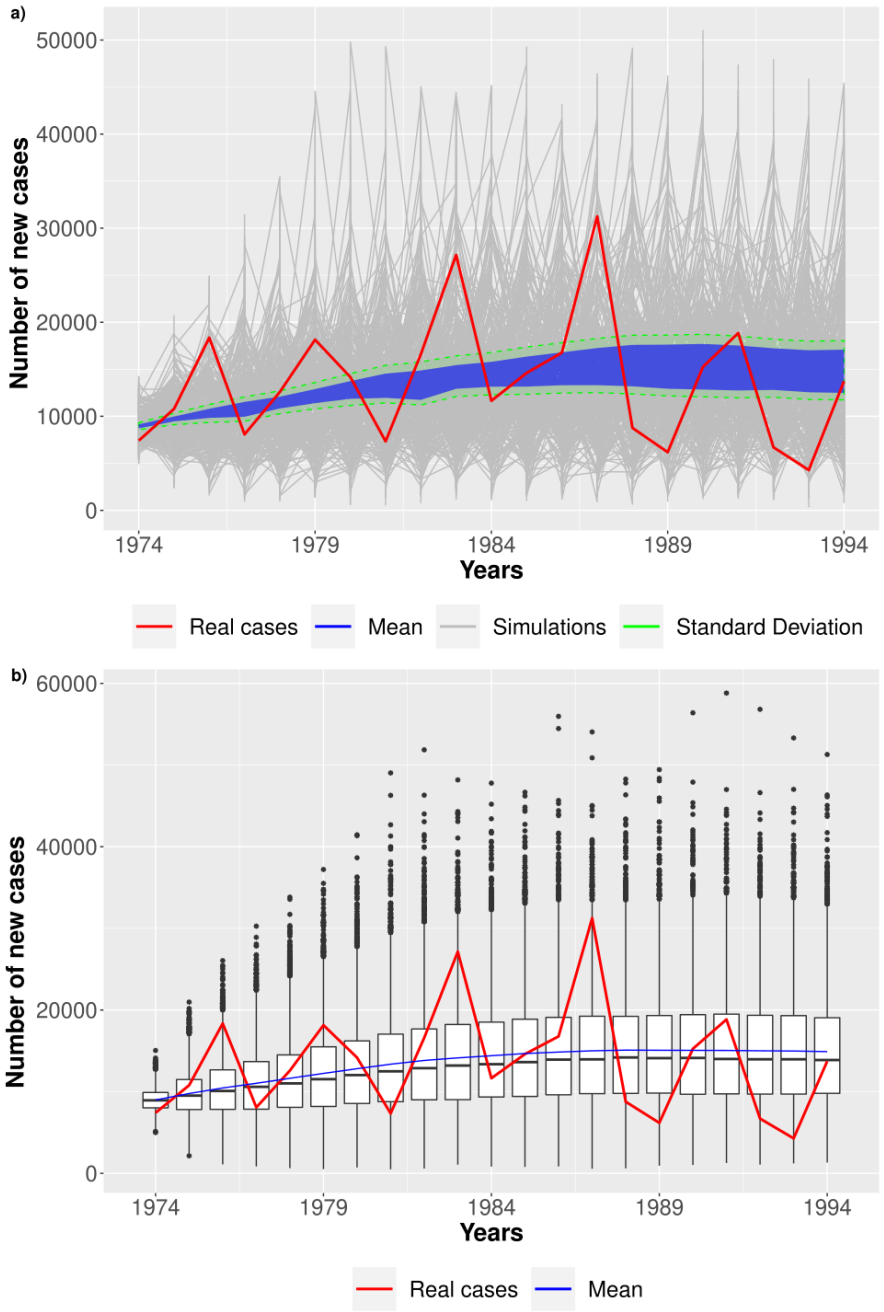


Figure 8.6: The number of new cases per year considering the stochastic model. a) 2500 trajectories (grey) over the whole time interval are reported. b) Boxplots over the time period considering the best configuration.

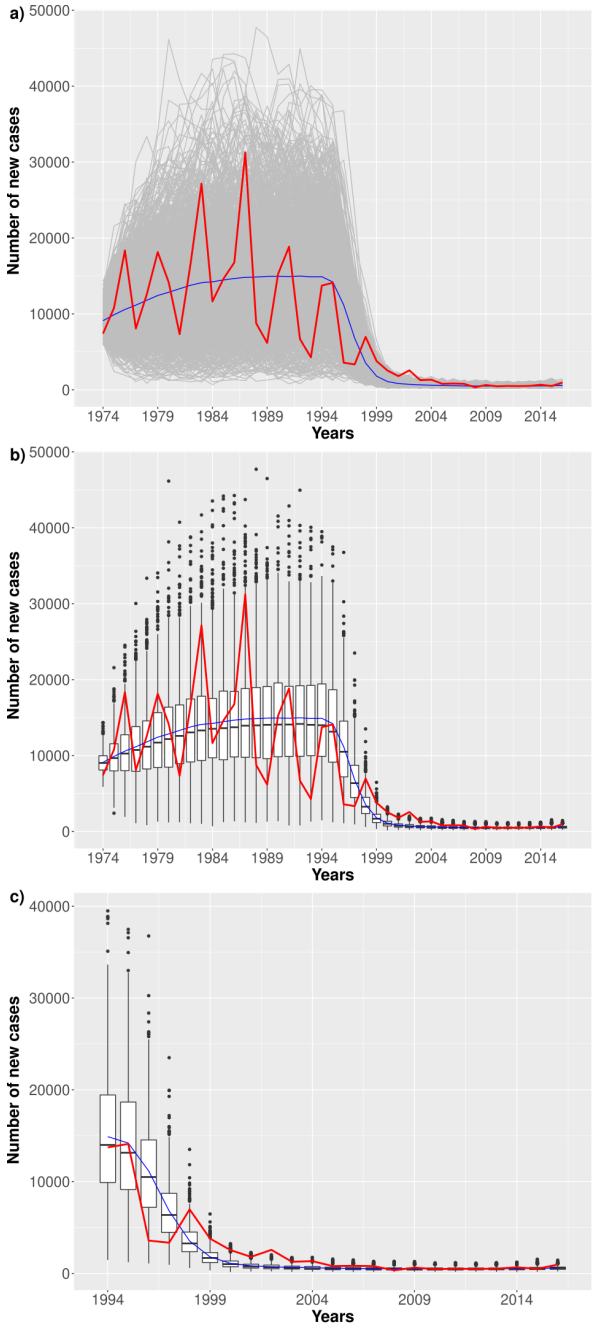


Figure 8.7: Probability of vaccine failure settled to zero. a) 1024 trajectories of the number of new cases per year (grey) considering the stochastic model over the whole time interval. The blue dashed line represents the mean trend. Finally, the red line represents the incidence of Pertussis. b) Boxplots over the time period. c) Zoom considering the last 21 years.

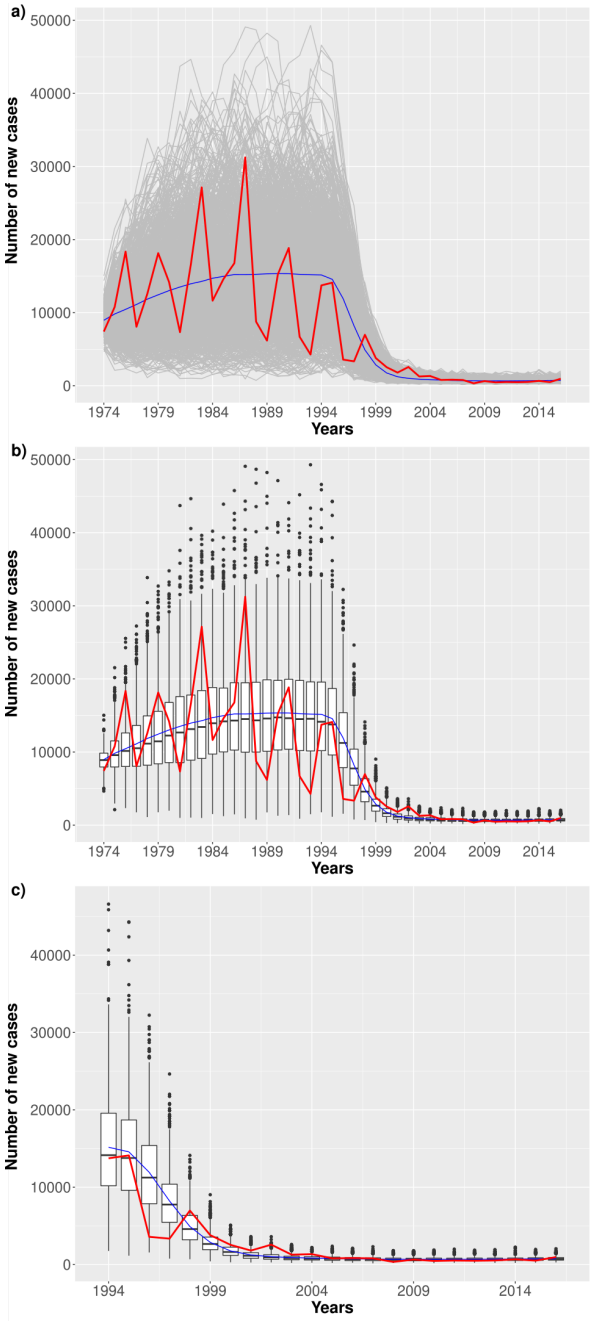


Figure 8.8: Probability of vaccine failure settled to 10%. a) 1024 trajectories of the number of new cases per year (grey) considering the stochastic model over the whole time interval. The blue dashed line represents the mean trend. Finally, the red line represents the incidence of Pertussis. b) Boxplots over the time period. c) Zoom considering the last 21 years.

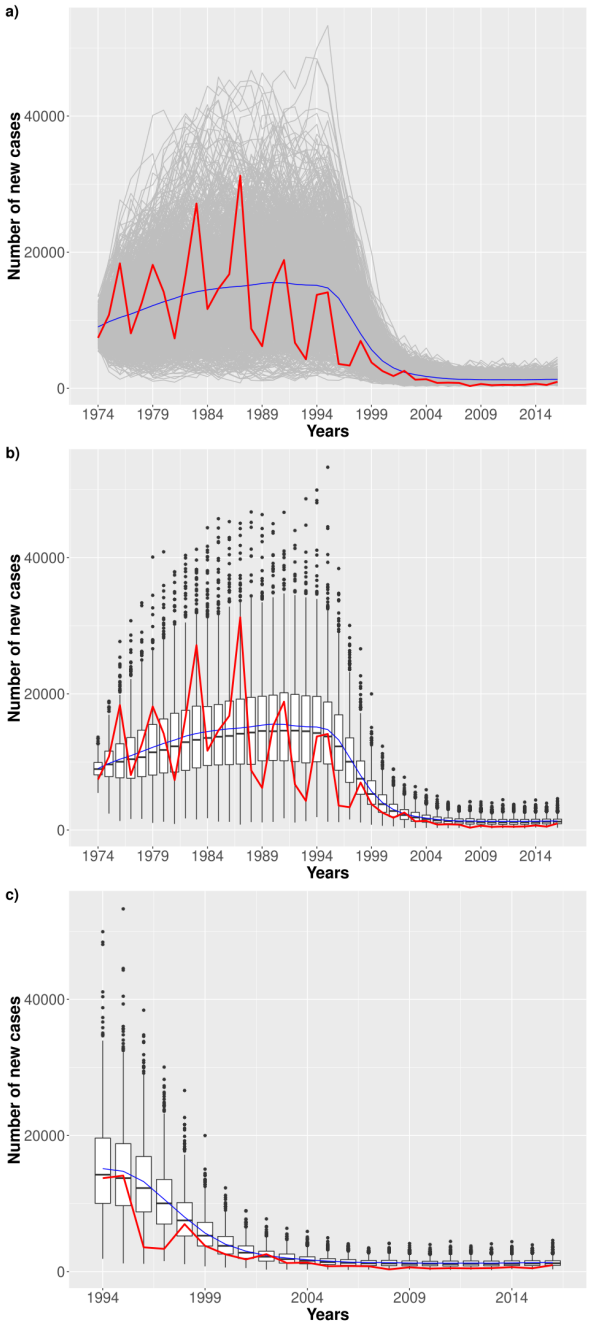


Figure 8.9: Probability of vaccine failure settled to 40%. a) 1024 trajectories of the number of new cases per year (grey) considering the stochastic model over the whole time interval. The blue dashed line represents the mean trend. Finally, the red line represents the incidence of Pertussis. b) Boxplots over the time period. c) Zoom considering the last 21 years.

Parameter name	**Final range**
prob_boost	$0.002523008 \sim 0.002531240$
prob_infectionS	$0.002528196 \sim 0.002529264$
prob_infectionR_l1	$0.002458931 \sim 0.002474028$
init_S_a1	866696
init_S_a2	15685680
init_S_a3	37628100
init_R_a1_nv_l4	7
init_R_a2_nv_l1	4
init_R_a2_nv_l2	2
init_R_a2_nv_l3	2
init_R_a2_nv_l4	2
init_R_a3_nv_l1	209184
init_R_a3_nv_l2	4
init_R_a3_nv_l3	4
init_R_a3_nv_l4	4

Table 8.2: Final parameters variability range used during the calibration of the model by solving the stochastic process τ-leaping algorithm.

parallel execution on 40 cores.

In Fig.s 8.7, 8.8, 8.9 we showed how the number of infection cases is affected by increasing the vaccination failure probabilities from 0 to 0.5. We observed that only probabilities greater than 0.3 have an effect on the number of infection cases. For a matter of space, we only report results for failure probability of 0 (the reference), 0.1 and 0.4.

Moreover, considering the same time period we further investigated the effects of varying the vaccination coverage of newborns in the period from 2006 to 2016. Fig. 8.10 and Fig. 8.11 show results for vaccination coverage of 90% and 80% respectively. The simulation of each case comprises of 1024 stochastic traces and has required 4 hours on Intel Xeon processor @ 2GHz, exploiting a parallel execution on 40 cores.

In details Fig. 8.10 a) and 8.11 a) shows how the number of new cases distribution for each year shifts upward when the fraction of vaccinated newborns decreases.

Looking at the initial vaccination years (i.e. from 2001 to 2006) of these figures it is possible to notice that the distribution of infects look quite alike, as indeed they are the realizations of the same stochastic process. On the other end, starting from 2006 the two distributions begin to differ reflecting

the changes in the vaccinated population.

Moreover, to better understand the effects on the distribution of the number of cases among the population, Fig.s 8.10b) and 8.11b) show the Empirical Cumulative Distribution Function (ECDF) of the number of cases in 2016 for both the reference data series and the one with the percentage of vaccinated newborns reduced to 90% and to 80%. Comparing the two ECDFs it is clear that reducing the vaccination coverage the probability mass is shifted toward higher number of cases in the population. Indeed, the slope of the ECDF in Fig. 8.10 b) is much more steeper in the initial stage (i.e., in the range between 1000 and 1250) than that in Fig. 8.11 b), meaning that a lower vaccination coverage remarkably increases the probability of having infection outbreak.

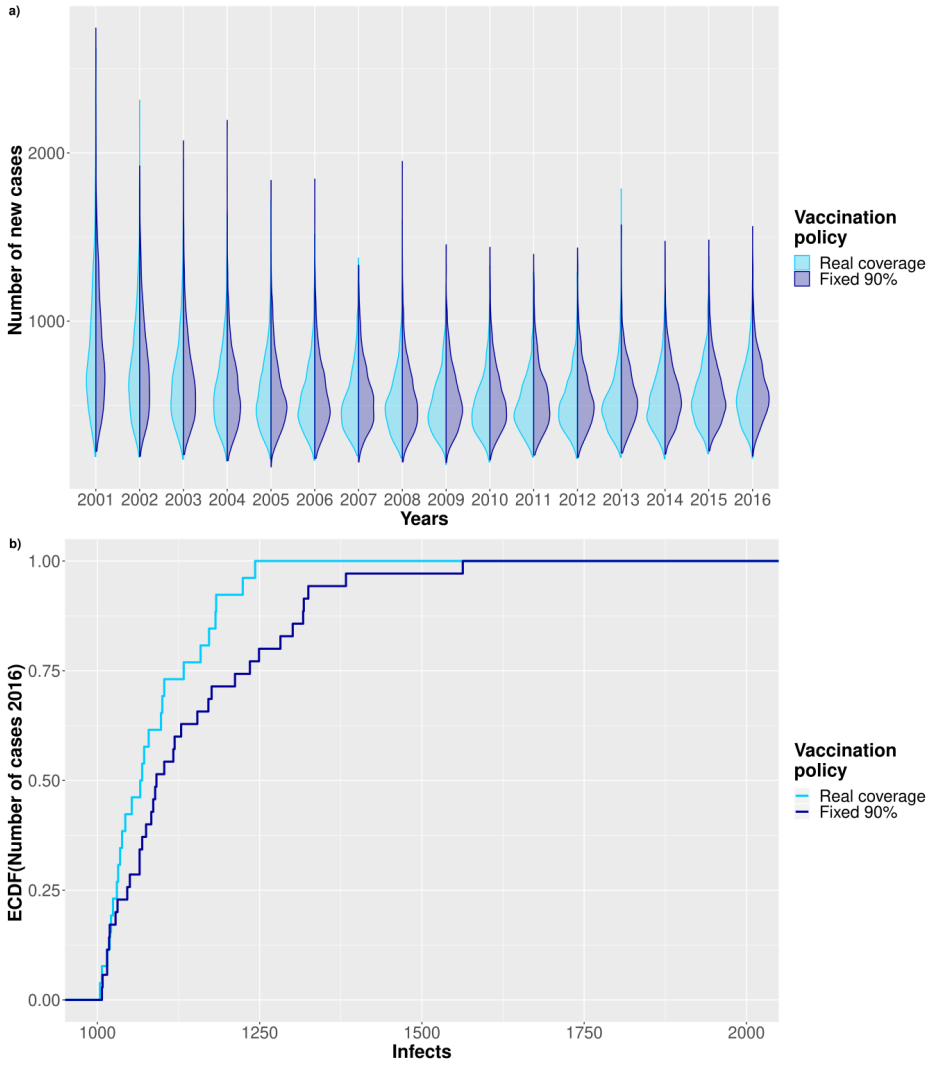


Figure 8.10: Comparison between 1024 stochastic traces following the reference data and a scenario where the population vaccinated is reduced to the 90% starting from 2006. a) Shows the violin plot comparing the distribution of the number of infected patients in the two scenarios. b) Comparing ECDF of the number of cases after 10 years of reduced vaccination rate.

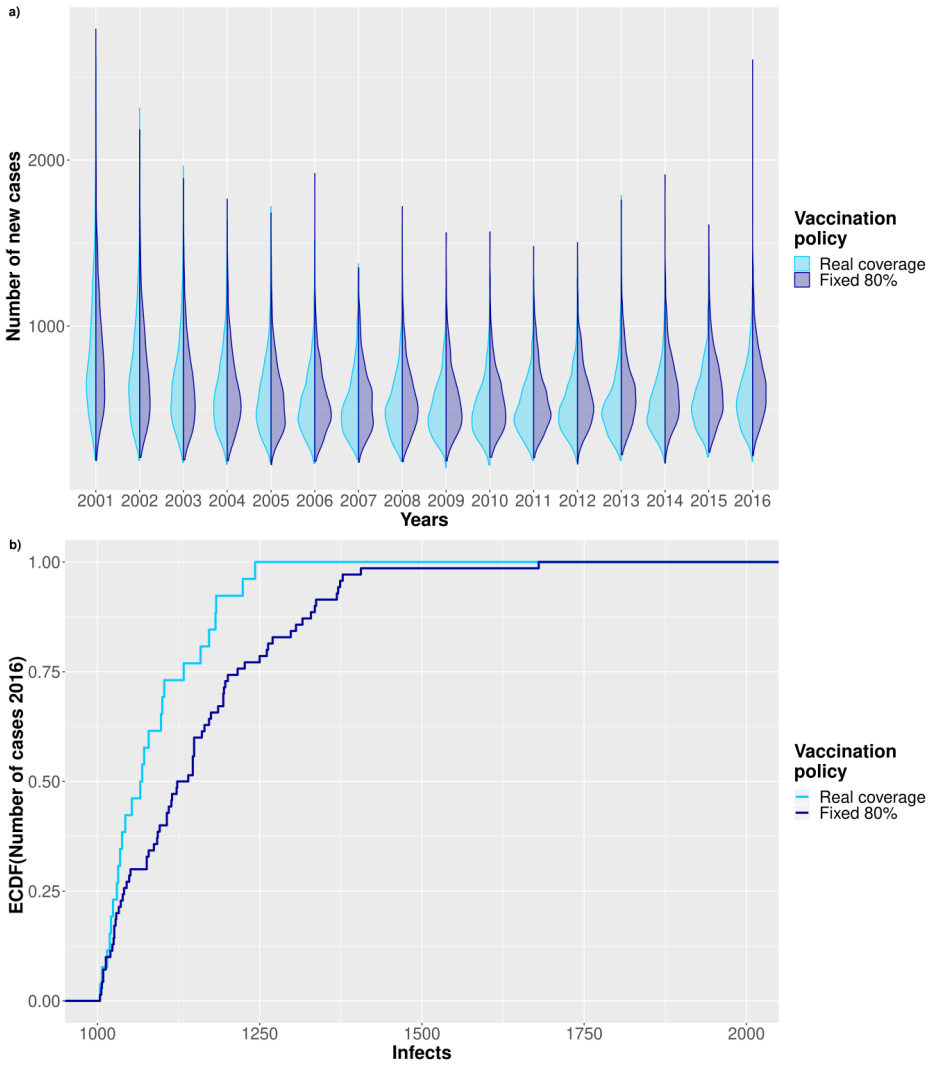


Figure 8.11: Comparison between 1024 stochastic traces following the reference data and a scenario where the population vaccinated is reduced to the 80% starting from 2006. a) Shows the violin plot comparing the distribution of the number of cases in the two scenarios. b) Comparing ECDF of the number of cases after 10 years of reduced vaccination rate.

8.1.4 Discussion

In this first case study we presented how to use our new general modeling framework, $\mathcal{G}reat\mathcal{M}od$, for the analysis of the pertussis epidemiology in Italy. The choice of studying this disease is due to the intrinsic complexity of its epidemiology and vaccination, and to the need of comprehensive studies capable

of addressing the many facets of this problem. Indeed, despite the fact that many models have been proposed since 1980s [70, 69, 167, 128, 156, 94] with the aim of providing insights on vaccination strategies, duration of immunity, and epidemic episodes, all of them share the characteristics of addressing only a subset of the specific peculiarities of the pertussis disease, and none of them faces the necessity of incorporating into a single model more details of the disease (e.g., the population age, the individual immunization level, ...) to better match the real observed dynamics and to predict the outcome of vaccination measures [19]. In Section 8.1.3 we showed that our framework can be easily exploited to construct and to analyse such a complex and comprehensive model (i.e., its underlying deterministic process is described by 179 ODEs and its underlying stochastic one is characterized by more than 1900 events). The development of such a model would be clearly unfeasible without the use of the graphical formalism; similarly, the analysis of such a representation would be difficult and error prone without the use of the suite of powerful solution tools integrated in the framework. As described in Section 8.1.3, the model was calibrated in order to reproduce the observed Italian pertussis spread from 1974 to 2016. Fig.s 8.6a) and 8.6b) show that the model provides a good approximation of the real data giving confidence on the possibility of using it to answer specific biological questions such as the impact of different vaccine failure probability and/or different vaccination coverage on the probability to have a pertussis outbreak. This shows that focusing on the analysis of specific biological questions, a model of this type can be used to perform a what-if analysis to assess the sensitivity of the model to variations of certain input parameters. The high level of parametrization and the flexibility provided by the framework gives the possibility of re-using the model and its analysis workflows for many other cases beyond the one studied in this section, representing one of the strengths of the proposed approach. For instance, with new contact matrices and new set of observed data, it would become possible to study the Pertussis (or other similar disease) in other countries other than the Italian one.

8.2 Multiple Sclerosis

In this Section we presented a second case study in which we applied the $\mathcal{G}reat\mathcal{M}od$ framework to model and analyse the Remitting Relapsing Multiple Sclerosis (Relapsing Remitting Multiple Sclerosis (RRMS)). In particular, by exploiting the main features characterizing our framework, i.e., the descriptive power of ESSN to provide a graphical representation of the complex biological system in a compact and parametric way, the sensitivity and calibration analysis to calibrate the model parameters, and the model analysis we were able to reproduce the typical oscillatory behavior relating to the onset of RRMS, followed by the analysis of different scenarios.

According to this, in Section 8.2.1 we first introduced the pathogenesis correlated to MS, highlighting the roles of the immune system cells and events that we modeled by exploiting the ESSN formalism. Then, to make easier the results presentation, we distinguished three versions of the same RRMS ESSN model[1]. Specifically, a first version of the RRMS model is presented in Section 8.2.2, which has been constructed to study the deterministic temporal dynamics of the disease, [116]. In details, two different scenarios of RRMS were considered. In the former scenario the effect of the daclizumab administration, an antibody tailored against the Interleukin-2 receptor of T cells, is investigated, while in the latter one the RRMS was studied in pregnant women. Successively, we extended in Section 8.2.3 the introduced model to analyse the deterministic spatial-temporal evolution of MS, by considering the cells movement into a three-dimensional grid. Moreover, we showed how the intrinsic symmetries of the ESSN model, see Section 6.1, can be exploited to drastically reduce the complexity of its analysis.Finally, in Section 8.2.4, a third version of the model introduced previously is proposed, in which (differently from before) we analysed the stochastic model behaviour and we included other biological entities relate to the immune system: (1) the Blood-Brain Barrier (BBB), that mainly has the function of protecting brain tissue from harmful elements present in the blood and that in MS is damaged and crossed by T cells [113]; (2) pro-inflammatory (e.g. Interleukin-17 (IL-17), Interferon gamma (IFNγ)) and anti-inflammatory (e.g. Interleukin-10 (IL-10)) cytokines, cell signaling molecules that modulate the immune response

[1]The files to reproduce these analysis are freely available at `https://github.com/qBioTurin/`.

through the activation of several pathways [113]. Moreover, the model is calibrated by exploring **experimental data** on 16 subjects, eight MS patients and eight healthy donors, in which the individual variability in terms of number of cells and cytokines production, in the blood and in the cerebrospinal fluid, had been quantified [129]. Then, the model behaviour without and with Daclizumab (DAC) administration was studied using SSA[57] method to take into account the stochasticity and the variability of the MS disease.
Let us note that for clarity and since the parameters and the functions of the general transitions are similar among the three model versions, we will report the mathematical details regarding only for the first version of the RRMS model, while for the other versions these details will be reported in the Appendix.

8.2.1 The Multiple Sclerosis disease

Multiple Sclerosis (MS) is a chronic and potentially highly disabling disease with considerable social impacts and economic consequences. In Europe it is the leading cause of non-traumatic disabilities in young adults, since more than 700,000 EU people suffer from MS [49].
Multiple sclerosis is an inflammatory autoimmune disease in which the patient's immune system reacts against itself by damaging Central Nervous System (CNS) nerve cells, i.e., compromising the ability of the neurons to send electrical signals, resulting in a progression of physical handicap until complete paralysis within 25 years in more than 30% of patients [154].

In literature four courses of MS are identified: Relapsing-Remitting MS (RRMS), Secondary Progressive MS (SPMS), Primary Progressive MS (PPMS), and Progressive Relapsing MS (PRMS). Among them the RRMS is the most common course since it is diagnosed in about 85% of MS cases. It is characterized by episodes of neurological dysfunction (i.e. relapses) followed by a complete or partial recovery (i.e. remissions). Unfortunately, within 25 years RRMS usually changes to SPMS (in about 90% of cases) increasing the severity of the disease [49].

Despite the etiology of MS is unknown, the scientific community agrees that MS involves a process mediated by immune system in which an abnormal response of the body's immune defense is directed against the CNS which is made up of brain, spinal cord and optic nerves. Within the CNS the immune system activates an inflammation process that damages the myelin (i.e.

the fatty substance that surrounds and insulates the nerve fibers), the nerve fibers themselves and the cells specialized in myelin production (i.e. Oligodentrocytes (ODC)). The myelin degradation process is mediated by self-reactive T cells which are activated in the peripheral lymph nodes and secrete pro-inflammatory cytokines (mainly IFNγ and IL-17). Among them IL-17 producing T cells sustain the pathogenesis of MS by promoting BBB disruption and inducing autoimmune inflammation in the CNS [86, 76]. Furthermore, IL-17 producing T cells are increased in the peripheral blood [48, 129], in the cerebrospinal fluid and in the CNS perivascular space of MS patients [155, 129]. These self-reactive T cells can be found also in healthy subjects but are strictly controlled by various mechanisms including suppression by T Regulatory (Treg) cells. In MS patients Treg cells are impaired in number and function [179] and allow self-reactive T cells to expand in the periphery, cross the BBB and reach the CNS, where they undergo into a secondary reactivation and induce demyelination and axonal damage [33, 42]. The triggers that convert the innocuous self-reactive T lymphocytes into pathogenic are still not understood, but a combination of genetic and environmental factors (e.g. Epstein-Barr Virus, vitamin D and smoking) [1] seems to be implicated. Some studies linked MS with Epstein-Barr Virus (EBV) infection due to the presence of higher titers of EBV antibodies in MS patients compared to age-matched controls [162].
Besides environmental factors, physiological factors also impact on the outcome of the MS disease. In particular, pregnancy represents a period of immune tolerance for patients that has important consequences on the relapse rate [177]. Indeed, pregnancy condition seems to have beneficial effects on women patients which have been associated with fewer relapses in RRMS. This phenomenon has been related with an increase in a particular type of immune cells, the Treg, which confers fetal tolerance and thus shows a protective effect of pregnancy to patients [100].

Until now a dozen treatments have been proposed to reduce the frequency of MS relapses, slow the accumulation of disabilities and contrast the RRMS progression. Such disease modifying therapies include oral agents and monoclonal antibodies (mAbs), which have been designed for a selectivity of drug action. Among mAbs DAC was selected for its ability to bind the CD25 subunit of the high-affinity Interleukin-2 receptor (IL-2R). IL-2R is a receptor-

structure able to bind a key component of the immune system, Interleukin-2 (IL-2), a cytokine that allows T cell proliferation. DAC introduced a new mechanism of action preventing the binding of IL-2 to its receptor with a consequent effect on immune cells which involves the blockade of T effector cells activation, the reduction of Treg cells and the increase of a particular Natural Killer (NK) cells subset with regulatory ability [141]. DAC efficacy was demonstrated in reducing (i) the clinical relapse rate of RRMS, (ii) the disability progression, and (iii) in improving health-related quality of life [141]. DAC appears to be generally well tolerated by MS patients with some adverse events as infections, encephalitis, and liver damages. However, the safety and efficacy results obtained after eight years of DAC treatment from the clinical trials were finally published [61].

8.2.2 The temporal model

In this section we introduced the ESSN model which will be exploited through the analysis, showing the ability of the model to reproduce the typical oscillatory behavior relating to the onset of RRMS by supposing a breakdown of the cross-balance regulation mechanisms at the peripheral level. In this contest, a challenging issue in the definition of the RRMS model was the calibration of the unknown parameters (e.g. EBV and DAC concentrations, and the transitions rate). Thus, in Paragraph **Sensitivity Analysis** the most critical parameters were identified exploiting the sensitivity analysis (see *Sensitivity analysis* paragraph in Section 7.1), and from this analysis two parameters configurations were obtained to represent a healthy subject and a RRMS patient. Then, two different sets of experiments were conducted. In the former set, reported in Paragraph **DAC therapy**, we inspected the effect of the DAC therapy in our model, which was firstly calibrated for reproducing the cells dynamics of RRMS patients. The latter set of experiments, Paragraph **Pregnancy**, were devoted to study the effect of RRMS in pregnant patients. The mechanisms at the basis of a partial MS remission during the pregnancy are not fully understood yet, leading this case particularly interesting.

The ESSN model

The cell and molecular interactions involved in the RRMS are described by the model showed in Fig. 8.12, which consists of 10 places and 22 transitions. We

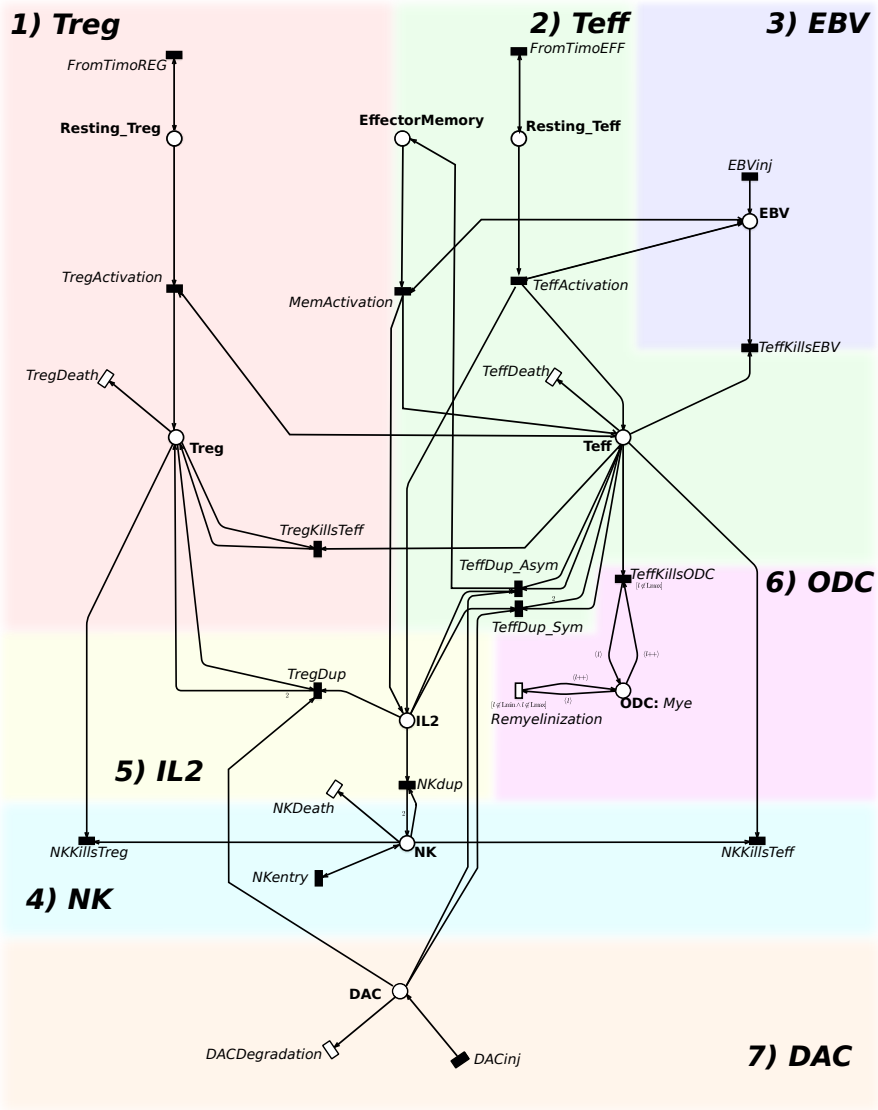


Figure 8.12: The RRMS model is composed by places corresponding to cells or molecules, and by transitions corresponding to the interactions among the entities, injections or death of molecules. The RRMS model is composed by seven modules: Treg, T Effector (Teff), EBV, NK, IL-2, ODC and DAC.

remember that the white transitions are the standard transitions following the MA law, while the black transitions are the general ones, following different kinetics described in details in Paragraph **General transitions**. For clarity, we organized the model into seven modules corresponding to the biological entities characterizing RRMS: Treg, Teff, EBV, NK, IL-2, ODC, and DAC. Let us now describe in details each module separately.

1) Treg module. The Treg cells are characterized by two places: the $Resting\_Treg$ and $Treg$. The transition *FromTimoREG* represents the arrival of new resting Treg cells from thymus. Its rate is defined in order to keep constant the number of resting cells. The transition *TregActivation* represents the activation of the resting Treg depending on the Teff cell number and EBV concentration, while *TregDeath* represents the death of Treg. The transition *TregKillsTeff* models the homeostatic regulation operated by Treg cells against self-reactive Teff cells, and *TregDup* models the Treg duplication.

2) Teff module. The second module is characterized by three places: $Resting\_Teff$, $Teff$, and *EffectorMemory*. The transitions *FromTimoEFF*, *TeffActivation*, and *TeffDeath* behave similarly to those described in module 1, but they are referred to the Teff population.
The Teff proliferation takes place in two different manners called *Symmetrical* and *Asymmetrical* processes. These two possibilities are captured in the model by assuming that one happens with probability p_{eff}^{dup} and the other with probability $p_{eff}^{mem} = 1 - p_{eff}^{dup}$. Given a replication speed named r_{dup}^{eff}, the transition *TeffDup_Sym* generates two Teff cells with the rate equals to $r_{dup}^{eff} * p_{eff}^{dup}$, for more details see Paragraph **General transitions**. Otherwise, the transition $TeffDup\_Asym$ takes place with a speed resulting from the product $r_{dup}^{eff} * p_{eff}^{mem}$ replicating one Teff cell into one T Memory effector cell and one Teff cell. The transitions *TeffKillsEBV* and *TeffKillsODC* encode the killing effect of Teff cells against EBV and ODC, respectively. Finally, *MemActivation* models the rapid activation of the Effector Memory depending on both the EBV and the Tmem concentrations. The transitions $TeffKillsEBV$ and *TeffKillsODC* encode the killing effect of Teff cells against EBV and ODC, respectively. Finally, *MemActivation* models the rapid activation of the Effector Memory depending on both the EBV and the Tmem concentrations.

3) EBV module. The third module describes the EBV behaviour. Transition *EBVinj* models the infection. The *TeffKillsEBV* transition summarizes all steps from antigen processing and presentation by EBV infected cells to Teff cells, to the activation of Teff cells.

4) NK module. In this modules the role of the NK cells is described. The transition *NKentry* models the arrival of new NK cells. The death of NK is then modeled by transition *NKDeath*. Transitions *NKKillsTeff* and *NKKillsTreg* encode the killing of self-reactive Teff and Treg cells respectively due to NK cells. Finally *NKdup* models the proliferation of the NK cells led by the presence of IL-2.

5) IL-2 module. The fifth module is focused on the $IL2$ role. $IL2$ is involved in the $Treg$, $Teff$ and NK proliferation. All these types of cells consume IL-2 which is produced by the transition *TeffActivation*.

6) ODC module. The sixth module encodes the ODC behaviour. The transition *TeffKillsODC* models the damage caused by Teff cells on ODC cells. When the myelination level reaches the lowest value, an irreversible damage occurs and the remyelinization is no more possible (i.e. the transition *Remyelinization* is permanently disabled by its guard). To model this effect, we used the color class *Mye* encoding the myelination levels of ODC. *Mye* is divided into five static subclasses ranging from *Lmin* (no myelination) to *Lmax* (full myelination).

7) DAC module. In the last module the daclizumab behaviour is modeled through the place *DAC*. The drug administration is modeled by transition *DACinj*, while the pharmacokinetic inhibiting the expansion of Treg and Teff decreases the velocity of transitions *TregDup*, *TeffDup_Sym* and *TeffDup_Asym*. Finally, its degradation is modeled by the transition *DACDegradation*.

Sensitivity Analysis

The sensitivity analysis step was exploited to select the input values (transition parameters, and the concentrations of EBV and DAC) leading the model outputs (i.e., the trajectories) towards the expected values obtained from observing the behaviour of specific quantities both in healthy and MS affected

subjects. In particular, thanks to this analysis, we were able to identify which parameters have more impact on the model outcomes. Then, the identified parameters were thoroughly investigated considering both the healthy and MS patients scenarios.

From our model (without the *DAC* module, Fig. 8.12) a system of 13 ODEs with nine unknown parameters is derived. These parameters were analysed through sensitivity analysis implemented in $\mathcal{G}reat\mathcal{M}od$, by generating 5000 parameter combinations using a *uniform* distribution whose ranges are showed in the second column of the Table 8.3.

For all the simulations, we assumed as initial marking the following parameters consistent with a space of $1mm^3$ of blood and $4mm^3$ of neural tissue: 500 ODC with level L_{max} of neuronal myelinization, 1687 resting Teff cells, 63 resting Treg cells, 375 NK cells and 1000 IL-2 molecules, and zero cells in the other places (see Table 8.4). Moreover, we defined the disease occurrence when the L_{min} level of neuronal myelinization is reached for each ODC cell, representing an irreversible damage. Then, five virus injections are simulated at regular times (every two months), introducing into the system 1000 EBV copies per injection. Finally, model solutions were calculated for each parameter combination over one year interval, $[0, 365]$ days.

Analyzing the 5000 trajectories generated, three scenarios have been identified: (i) the occurrence of the MS, represented by a huge number of dead ODC cells; (ii) the complete remission of the MS disease, characterized by a low number of dead ODC cells and with a complete elimination of the EBV virus; (iii) the partial remission of the MS disease specified by a partial elimination of the EBV virus. The Fig. 8.14 reports the EBV and ODC dynamics generated considering different set of parameters.

On these trajectories the PRCC analysis was applied to identify key model parameters affecting the system behaviour. The PRCCs values are calculated for each parameter over the entire time period, which are showed in Fig. 8.13.

The rates associated with transitions *TeffKillODC*, *TregKillTeff*, *TeffKillEBV* and *Recovery* result to be the crucial parameters affecting the *ODC* behaviour. Fig. 8.15 reports a scatter plot in which each point corresponds to a generated trajectory, its color represents the percentage of irreversible ODC damaged at the final time point (i.e. a grey color corresponds to a lowest percentage of damaged ODC and a red color to highest one). The simulations are performed changing the rates of *TeffKillEBV* (in the x-axis), the

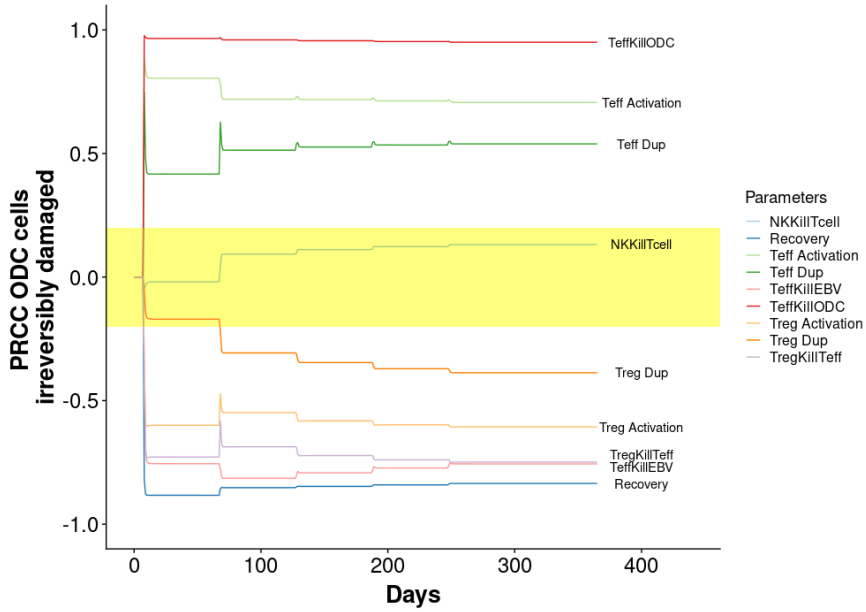


Figure 8.13: Sensitivity analysis. PRCCs over the whole time interval for each model parameter is reported. Yellow area represents the zone of non-significant PRCC values.

TregfKillTeff (in the y-axis) and *TeffKillODC* (in the z-axis). A few number of irreversible damaged ODC are obtained increasing the *TregfKillTeff* rate and the decreasing the others two rates.

The key parameters identified were deeply studied exploiting the LHS method computing 500 new combinations varying only *TeffKillEBV*, *TregfKillTeff* and *TeffKillODC*. We defined two sets of parameters (see Table 8.5), one for the MS patients and one for the healthy subjects, see Fig. 8.16.

The MS patients are modeled by a set of parameters which maximizes the ODC damage maintaining the Treg and Teff cell numbers consistent with those measured in the reality, see red trajectories in Fig. 8.16 a and b panels.

For the healthy cases, we selected the trajectory providing a Teff-Treg regulatory balance able to control the spread of the EBV virus and to minimize the irreversible damage to ODC cells, even if the amount of EBV in each injection is substantially increased, see blue trajectories in Fig. 8.16, panels a and b. In particular, we performed 500 simulations varying the amount of EBV injected in a range of $[1000 - 5000\ particles/mm^3]$. From Fig. 8.17 it is possible to observe that, even for large quantities of EBV injected, the

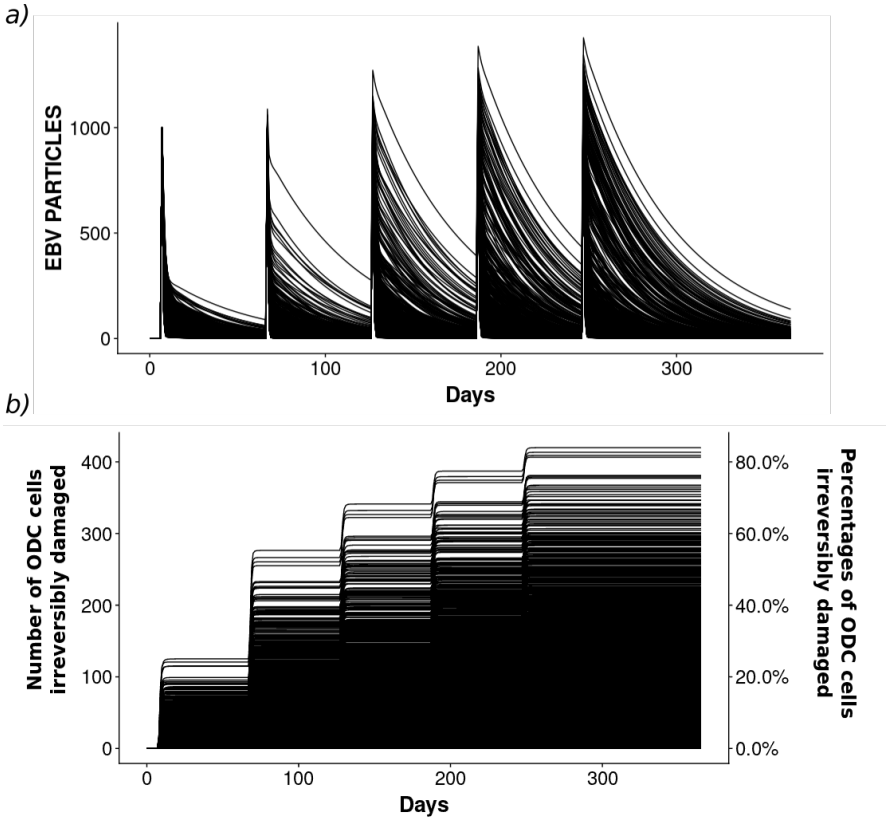


Figure 8.14: A subset of the 5000 trajectories generated by Latin Hypercube Sampling (LHS) of the EBV cells (a) and the ODC cells with an irreversible damage (b) over the whole time interval.

percentage of irreversibly damaged ODCs reaches 17% (Fig. 8.17(a)). This value is very small if compared with respect to the 77% of irreversibly damaged ODCs in the case of the disease occurrence. Moreover, independently of the quantity of EBV injected, Teff are able to eliminate EBV completely (Fig. 8.17(d)), and the abundance of EBV does not drastically affect the number of effectors or regulators in the system (Fig. 8.17(b,c)).

Finally, we can conclude this part affirming that the LHS with PRCC index identified *TeffKillODC*, *TregKillTeff*, *TeffKillEBV* as the most critical parameters to the model outcomes. This result agrees with our expectation since these parameters play a central role in the disease progression. In the analysis of the EBV behaviour in a healthy subject, it is interesting to note the effect of immune memory which increases the number of activated Teff cells from the time of second injection (see Fig. 8.18). In particular, thanks to

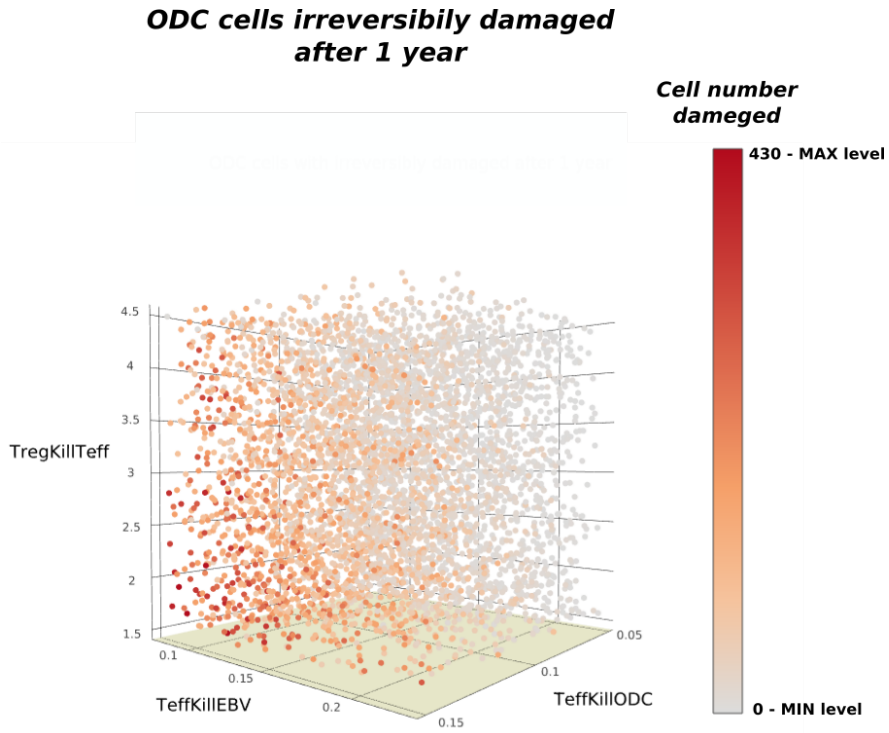


Figure 8.15: 3D scatter plot of the ODC irreversible damaged at the fixed time 365 versus the *TeffKillODC*, *TregKillTeff*, *TeffKillEBV* parameters variation.

the faster activation of the Tmem cells with respect to the Teff cells, from the second EBV injection it is possible to observe a more rapid virus annihilation. Indeed, Tmem cells have a faster activation (i.e. since they already have the memory of a previous contact with the EBV) than Teff cells, leading to a more rapid virus annihilation during the relapses.

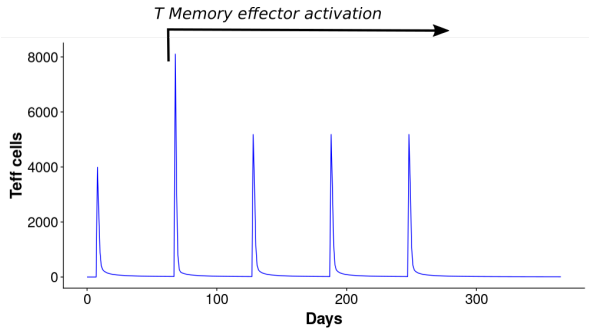


Figure 8.18: The Teff dynamics considering an MS subject.

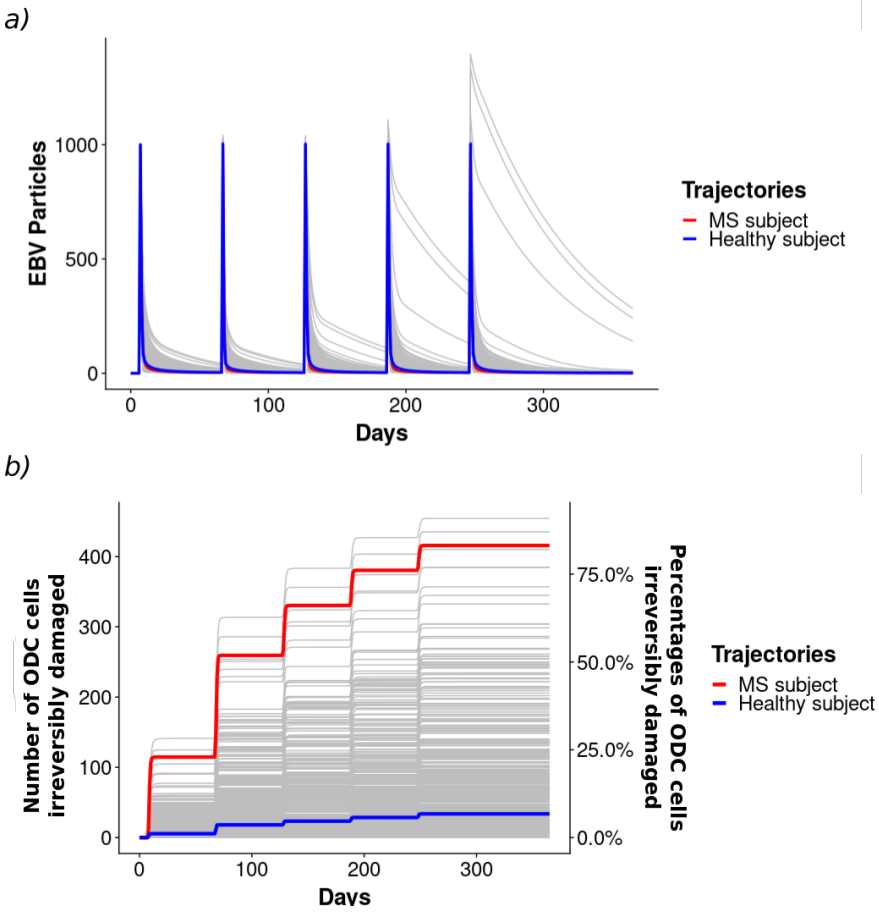


Figure 8.16: A set of the 500 trajectories generated by LHS of the EBV virus (a) and the ODC cells with an irreversible damage (b) over the whole time interval varying the *TeffKillODC*, *TregKillTeff*, and *TeffKillEBV* transition parameters.

DAC therapy

To investigate the effect of the DAC therapy in our RRMS model calibrated for MS patients, we simulated the DAC administration at the 53rd day after the first EBV injection. Our results are reported considering two important aspects in the modulation of a therapy: the drug dose and the drug degradation time. The quantity of *DAC* administrated per injection and the *DAC* cells deterioration were studied by means of LHS method. The values of these two parameters are then sampled according to two uniform distributions whose ranges are reported into the Table 8.3.

From the sensitivity analysis we clearly observed that the drug degradation

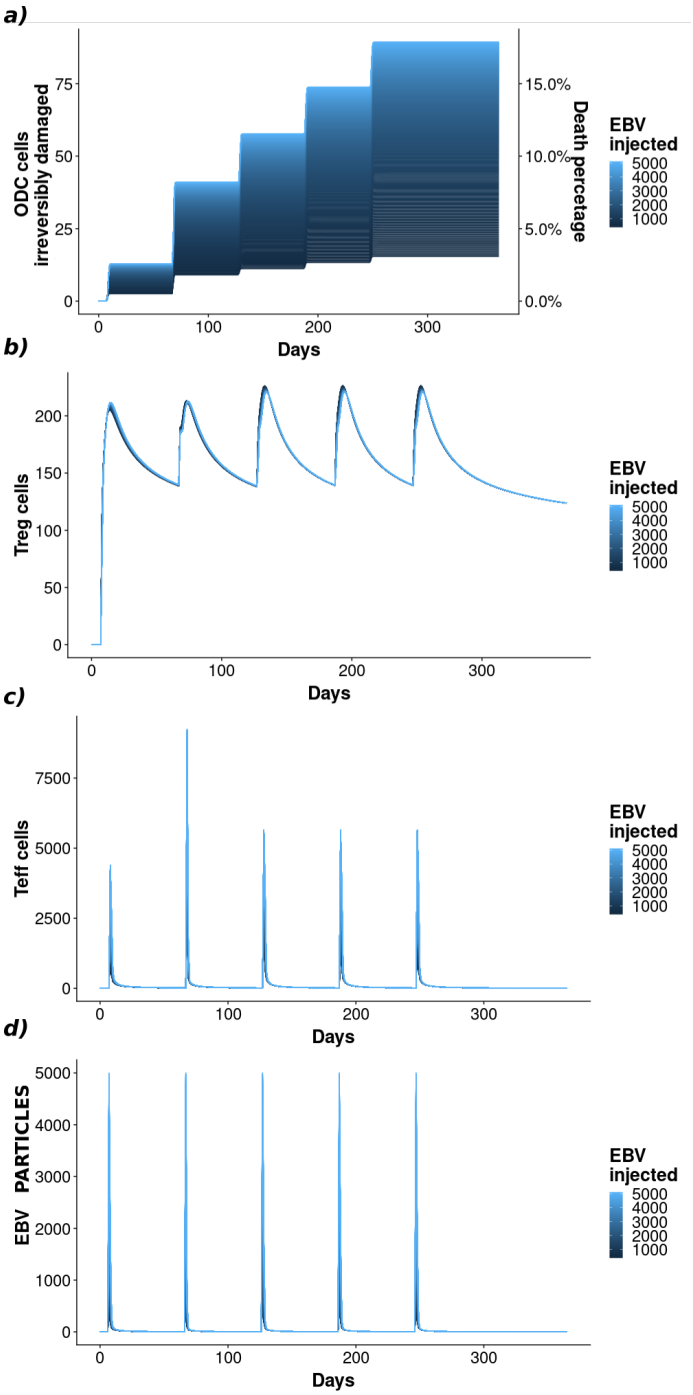


Figure 8.17: Different injections of EBV (d) are considered to check if the Teff-Treg (c-b) regulatory loop is able to control the virus spreading minimizing the irreversible damages to the ODC cells (a).

Transitions/events	Parameters	Range
Treg Death	r_{TregD}	$1/24\ h^{-1}$
Teff Death	r_{TeffD}	$1/24\ h^{-1}$
NK Death	r_{NKD}	$1/24\ h^{-1}$
NK Dup	r_{NKDup}	$1/24\ h^{-1}$
Teff Activation	r_{TeffA}	$[0.2, 0.6]\ h^{-1}$
Treg Activation	r_{TregA}	$[0.1, 0.3]\ h^{-1}$
Treg Dup	$r_{TregDup}$	$[0.045, 0.135]\ h^{-1}$
Teff Dup	$r_{TeffDup}$	$[0.25, 0.75]\ h^{-1}$
TeffKillODC	r_{TeKodc}	$[0.05, 0.15]\ h^{-1}$
TregKillTeff	r_{TrKTe}	$[1.5, 4.5]\ h^{-1}$
TeffKillEBV	r_{TeKebv}	$[0.075, 0.225]\ h^{-1}$
Recovery	r_{rec}	$[0.075, 0.225]\ h^{-1}$
NKKillTcell	r_{NKkTc}	$[0.05, 0.15]\ h^{-1}$
DACDeath	r_{DacD}	$[0.0004, 0.001]\ h^{-1}$
DACinjection	r_{DacInj}	$[5, 100]\ h^{-1}$

Table 8.3: List of the model fixed and unknown parameters with their corresponding values or (in the latter case) ranges on whose the Uniform distribution is defined.

Cell	Value	Reference
$TLymphocytes$	$[3 * 10^3 cells/mm^3]$	[4, 165]
$RestingTeff$	$[1687 cells/mm^3]$	[136, 14, 31, 133]
$RestingTreg$	$[63 cells/mm^3]$	[145]
NK	$[375 cells/mm^3]$	[136, 14, 31, 133]
ODC	$[125 cells/mm^3]$	[139]
$EBVinfection$	$[50 - 70 days]$	[7]

Table 8.4: List of the cell numbers used in the model.

Transition	Teff Activation	Treg Activation	Treg Dup	Teff Dup	TeffKillODC	TregKillTeff	TeffKillEBV	Recovery	NKKillTcell
Healthy	0.4	0.2	0.09	0.5	0.1	3	0.15	0.1	0.1
Sick	0.4	0.2	0.09	0.5	0.15	1	0.1	0.1	0.1

Table 8.5: Parameters used for simulating the Healthy version (first row) and Sick version (second row) of the disease.

time has a greater impact on the elimination of EBV virus than the amount of DAC administered, see the number of ODC irreversibly damaged in Fig. 8.19. Therefore, we decided to focus our attention on the *DACDegradation* parameter variation. Knowing that the half-life of DAC was detected around 22 days, we considered that a complete degradation of DAC ranges between 30 and 90 days [79]. The results of the simulations are reported in Fig. 8.20(a) in which it is possible to appreciate that a greater DAC permanence has the effect of reducing the number of irreversibly damaged ODC cells with respect to the case in which no therapy was considered (red line). Moreover, it is interesting to note that the RRMS model with DAC injections highlights a decrease of the long term ability of the immune system to eliminate EBV Fig. 8.20(b). Finally, in Fig. 8.21 is reported the trend of the NK cells that increase with respect to the DAC degradation rate.

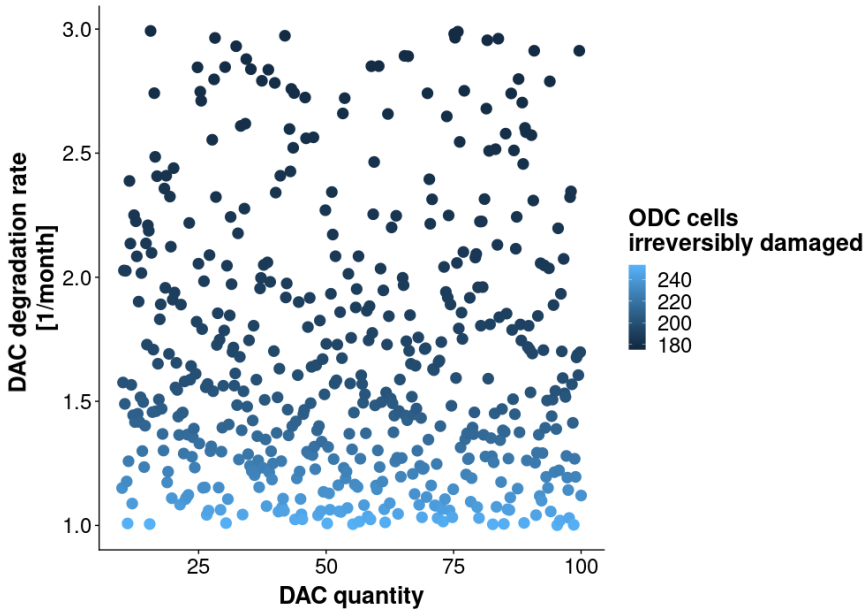


Figure 8.19: Scatter plot of the ODC irreversibly damaged variable at the fixed time 365 depending on the DAC injected (x-axis) and the DAC death rates (y-axis). The colour depends on the number of ODCs irreversibly damaged. The number of ODC is strongly dependent by the DAC degradation: the decrease of the number of damaged ODC is more influenced by an increase of the permanence time of DAC drug in the body.

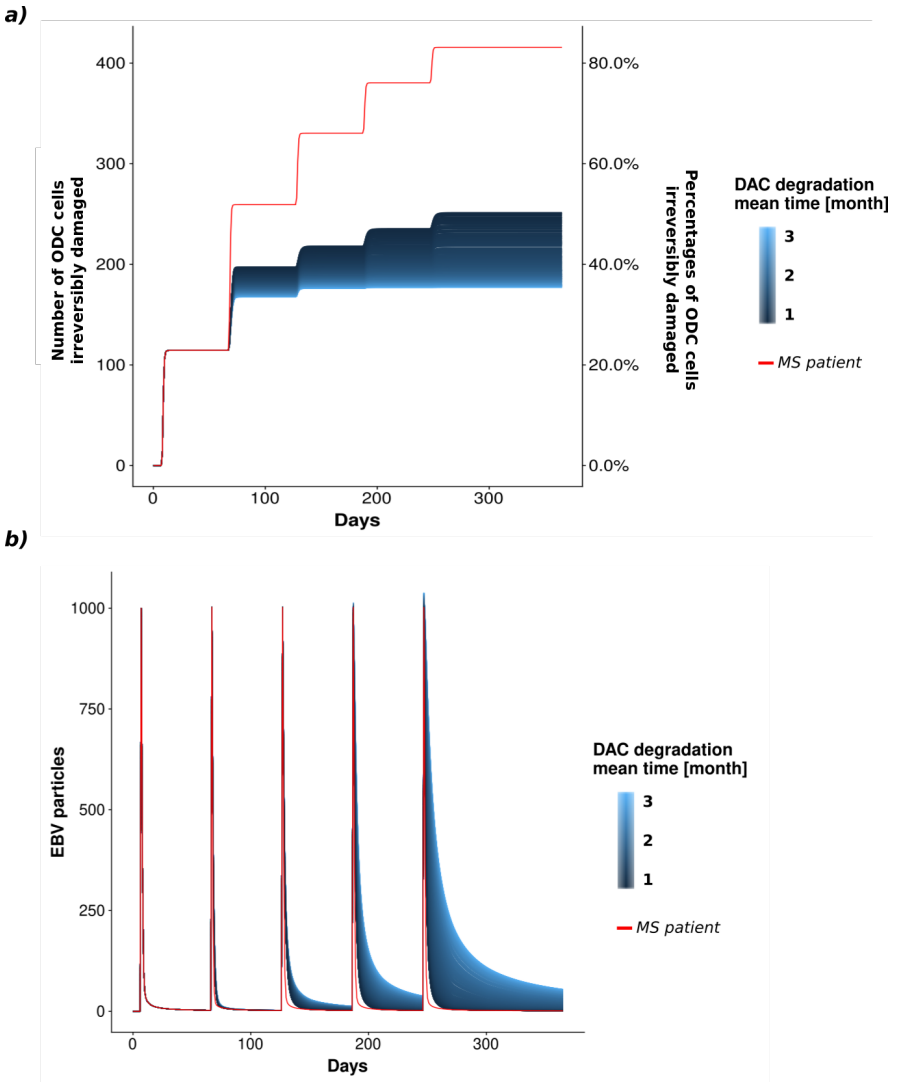


Figure 8.20: *ODC* and *EBV* trajectories colored depending on *DAC* degradation rate (expressed in months). The red line represents the starting sample without drug administration.

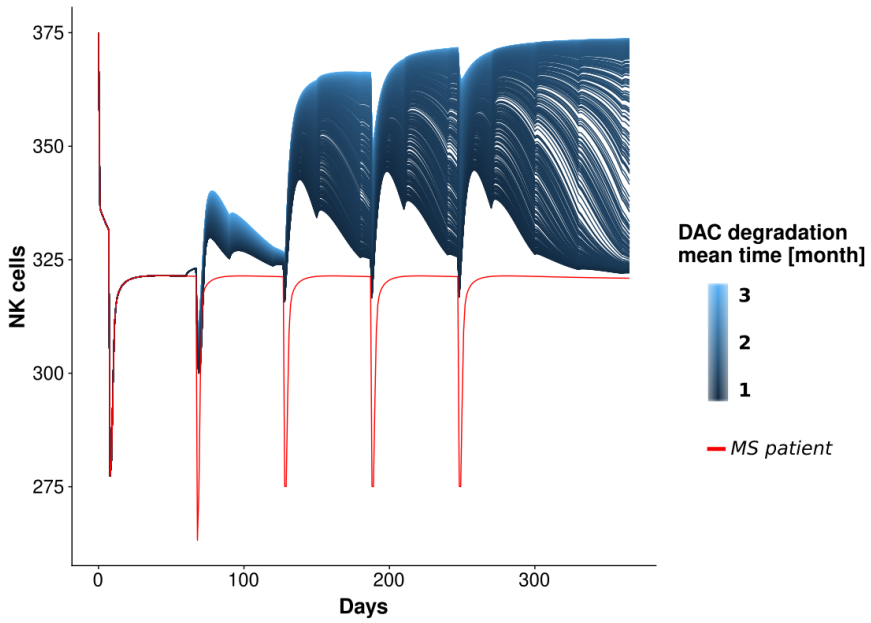


Figure 8.21: *NK* trajectory colored depending on *DAC* degradation rate (expressed in months). The red line represents the starting sample without drug administration.

Pregnancy

In this subsection we investigate the RRMS in pregnant women. As already pointed out before, pregnancy was associated with fewer relapses in RRMS and reduced activity of disease in autoimmune encephalomyelitis (EAE). Beneficial effects of pregnancy are thought to be related to pregnancy-associated changes in the maternal immune system. One of the observations is that Treg cells increase in number establishing the fetal tolerance and conferring a temporary protection to women with RRMS [135, 145].
According to the literature, we modeled the pregnancy condition changing the proportion between the activate Treg cells and the activate Teff cells decreasing the Teff activation rate and increasing the Treg activation rate proportionally to the pregnancy phase [145]. Three pregnancy phases, corresponding to the three trimesters, have been simulated. When a new trimester begins, we increased the ratio of *TregActivation* rate to *TeffActivation* rate; while at delivery time both rates return to their initial values. Thus we simulated 100 different scenarios with a increasing variation of parameters, obtaining different levels of protection from ODC damage. As expected, the model behaviour

shows a substantial reduction of the ODCs damage (see Fig. 8.22).
Regarding the immune system cells, we observed that Treg cells increase during pregnancy and then suffer a sharp decline at the time of delivery. The same effect, but in the opposite direction is showed on Teff cells. It is interesting to note that a rebound of Teff is reported in the week following pregnancy, see Fig. 8.23.
Our results are in line with what comes to us from biological knowledge and clinical observations since the resetting of the immune system is what significantly influences the course of the disease during pregnancy and also has been related with clinical manifestation of increased relapses associated with the post-partum period. From our results it is possible to appreciate the difference between the ODC cells irreversibly damages in the case of MS no pregnant patients and MS pregnant patients (specially in the case of max variation of the Treg-Teff balance). This difference increases from the first trimester to the time of delivery then returns to become not significant.

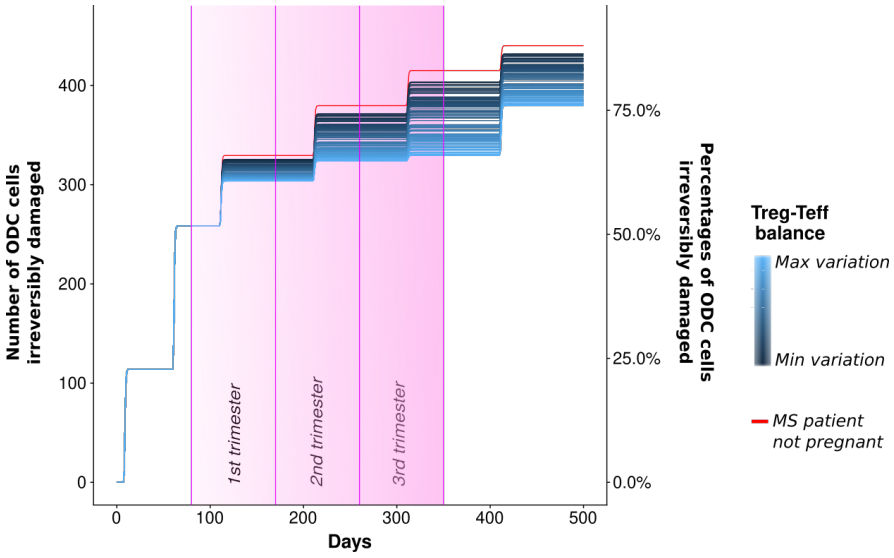


Figure 8.22: The ODCs irreversibly damaged considering the pregnant woman case of study. 100 trajectories colored depending on different variations of the *TregActivation* and *TeffActivation* parameters. The red line represents the starting sample without pregnancy. Furthermore each trimester another variation is applied to these parameters in order to represent the increasing of the maternal immune system.

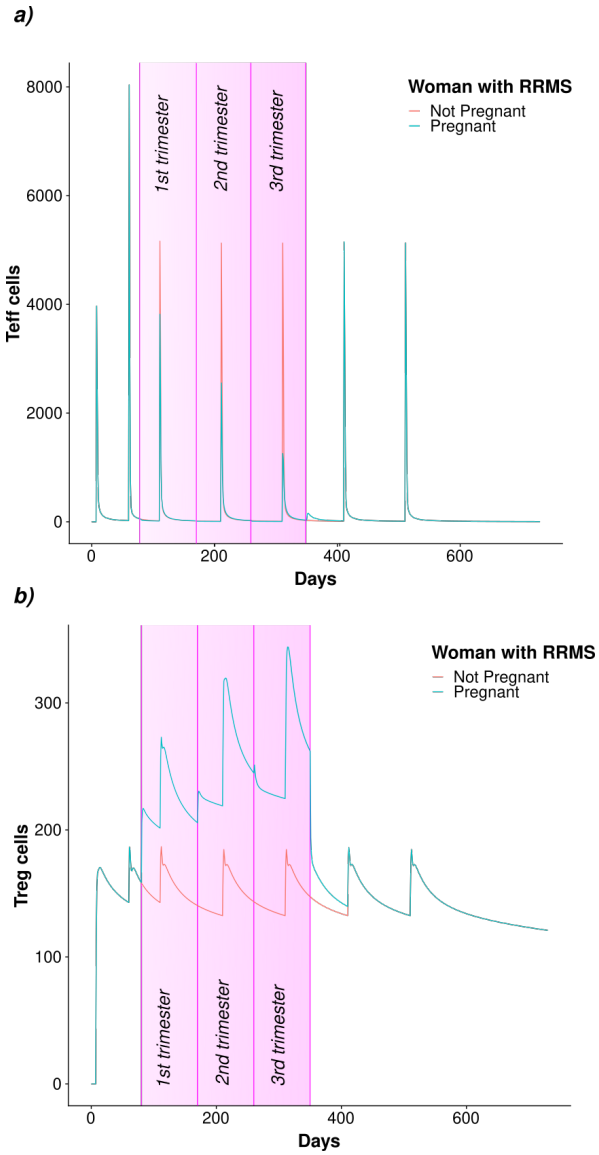


Figure 8.23: The Teff a) and Treg b) dynamics before, during and after the pregnancy. The red line represents the starting sample without pregnancy.

General transitions

In this subsection we describe in details the firing rate function associated with the general transitions in the RRMS model introduced in Paragraph **The temporal model**. These transitions model the following biological events: i) the killing of a cell, e.g., *TregKillsTeff* or *TeffKillsODC*, ii) the entry of cells into the system, such as *EBVinj*, iii) the activation of T cells, e.g., *TeffActi-*

vation, and iv) the duplication of a cell,e.g., such as *TeffDup*.
Let us recall the notations characterizing the general transition introduced in Section 5.2: $f_{\langle t,c\rangle}(\hat{x}(\nu),\nu)$ is the speed of the transition $t \in T_g$ and $\hat{x}(\nu)$ represents the vector of the average number of tokens for all the input places of t. For brevity when the function will not depend on the color instance c then we will omit it reporting just the transition t, i.e. $f_t(\hat{x}(\nu),\nu)$. All the general transitions of the model are now explained in details. All the constants and numerical values associated with the transitions are summarized in the Tables 8.3 and 8.6.

- *EBVinj*, *DACinj* inject into the system specific quantities of EBV and DAC respectively at fixed time points;

- *FromTimoREG*, *FromTimoEFF*, and *NKentry* are the transitions which keep in a constant range the number of $RestingTreg$, $RestingTeff$, and NK respectively. They are defined as

$$f_{FromTimoREG}(x_{ResTreg},\nu) = q_{RestTreg} * (1 - x_{ResTreg}/63);$$
$$f_{FromTimoEFF}(x_{ResTeff},\nu) = q_{RestTeff} * (1 - x_{ResTeff}/1687);$$
$$f_{NKentry}(x_{NK},\nu) = q_{NK} * (1 - x_{NK}/375),$$

  where $x_{ResTreg}$, $x_{ResTeff}$, and x_{NK} are the numbers of cells in the input places (i.e. $RestingTreg$ for *FromTimoREG*, etc) at time ν. Then $q_{\cdot}$ represents the quantity injected in the output place to preserve the cell quantity, i.e. 63 for the $RestingTreg$, 1687 for the $RestingTeff$ and 375 for the NK (see Table 8.4).

- *TregActivation* and *TeffActivation* transitions model the activation of the Teff and Tref cells. In particular, these transitions are defined of type general to simulate a reduced Teff activation velocity when the virus presence decreases, and a Treg activation velocity which is proportional to the number of Teffs and inversely proportional to the number of EBV particles (allowing the Teff to annihilate the virus). So the functions are defined as

$$f_{TregActivation}(\hat{x}(\nu),\nu) = r_{TregA} * \frac{x_{Teff}}{(x_{Teff} + x_{EBV} + 1)} * x_{ResTreg};$$
$$f_{TeffActivation}(\hat{x}(\nu),\nu) = r_{TeffA} * (1 - exp(-\frac{x_{EBV}}{C_{EBV}})) * x_{ResTeff},$$

where r_{TregA} and r_{TeffA} are the activation constant rates for the Treg and Teff respectively. In case of the $TregActivation$ transition, the vector $\hat{x}(\nu)$ of its input places $RestingTeff, EBV$ and $Teff$ consists of the variables $x_{ResTreg}, x_{EBV}, x_{Teff}$ respectively. Differently the $\hat{x}(\nu)$ of $TeffActivation$ transition is characterized by $x_{ResTeff}$ and x_{EBV}. Finally the constant C_{EBV} is related to the EBV particles and it is defined to reduce the activation rate with the decreasing of the virus presence.

- *MemActivation* is defined as

$$f_{MemActivation}(\hat{x}(\nu), \nu) = \begin{cases} 0 & \nu < t_{2inj}, \\ r_{MemA} * x_{Mem}(\nu) & \nu \geq t_{2inj}, \end{cases}$$

where

$$r_{MemA} = 2 * r_{TeffA} * (1 - \exp(-\frac{x_{Mem}(\nu)}{C_{Mem}}) * (1 - \exp(-\frac{x_{EBV}}{C_{EBV}}),$$

and t_{2inj} is the time corresponding to the second EBV injection. We are considering the velocity of this transition as zero $\forall \nu < t_{2inj}$, since the T Memory effectors start to react after the first virus occurrence. $\hat{x}(\nu) = (x_{Mem}(\nu), x_{EBV}(\nu))$ is the marking vector storing the number of T Memory effectors and EBV particles respectively at time ν. C_{Mem} and C_{EBV} constants are related to the Memory and EBV cells needed to slow down the activation rate with the decreasing of EBV and Memory cells. This is due to the necessity of leaving a minimum number of T Memory effectors into the system. So when in the system there are a large number of EBV particles and of T Memory effectors, the activation speed reaches its maximum given by twice the velocity of the Teff cells, r_{TeffA}.

- All the transitions modeling the killing of a specific cell are defined as follows:

$$\forall t \in \{ TregKillsTeff, TeffKillsODC, TeffKillsEBV, NKKillsTcell \}$$

then

$$f_t(\hat{x}(\nu), \nu) = \frac{1}{x_{tot}} * r_t * \prod_i \hat{x}_i(\nu),$$

where $\prod_i \hat{x}_i(\nu)$ is the product of the average numbers of tokens in the input places of the transition t, r_t is the constant rate related to the transition t, x_{tot} is the total number of cells at time ν, and $\frac{1}{x_{tot}}$ represents the probability that a specific meeting between two different cells is occurred.

- *TregDup* transition models the Treg duplication depending proportionally on the amount of IL-2 and inversely proportionate on the number of DAC cells (to simulate the reduced duplication velocity during the DAClizumab therapy), and it is defined as:

$$f_{TregDup}(\hat{x}(\nu), \nu) = \eta_{TrD}(\hat{x}(\nu), \nu) * x_{Treg} * x_{IL2} * \frac{1}{x_{tot}},$$

with

$$\eta_{TrD}(\hat{x}(\nu), \nu) = r_{TregDup} * (1 - exp(-\frac{x_{IL2}}{C_{IL2}})) * (exp(-\frac{x_{DAC}}{C_{DAC}})),$$

where $r_{TregDup}$ is the constant Treg duplication rate,
$\hat{x}(\nu) = \{x_{Treg},\ x_{IL2},\ x_{DAC}\}$ and C_{IL2} and C_{DAC} are the constant related to the IL-2 and DAC cells to slow down the duplication velocity with an increasing number of DACs and a decreasing number of IL-2 proteins.

- Considering the Teff duplication event we have to distinguish two possible cases: 1) the Teff symmetric duplication with probability p_{eff}^{dup} and a Teff asymmetric duplication, implying the T Memory effector differentiation, with probability $p_{eff}^{mem} = 1 - p_{eff}^{dup}$. This is modeled exploiting two different transitions: *TeffDup_Sym* and *TeffDup_Asym*. So let us define

$$r_{dup}^{eff} = \eta_{TeD}(\hat{x}(\nu), \nu) * x_{Teff} * x_{IL2} * \frac{1}{x_{tot}}$$

then these two transitions are defined as:

$$f_{TeffDup\_Sym}(\hat{x}(\nu), \nu) = p_{eff}^{dup} * r_{dup}^{eff}$$

and

$$f_{TeffDup\_Asym}(\hat{x}(\nu), \nu) = p_{eff}^{mem} * r_{dup}^{eff},$$

with

$$\eta_{TeD}(\hat{x}(\nu), \nu) = r_{TeffDup} * (1 - exp(-\frac{x_{IL2}}{C_{IL2}})) * (exp(-\frac{x_{DAC}}{C_{DAC}})).$$

Where $r_{TeffDup}$ is the constant Teff duplication rate,
$\hat{x}(\nu) = \{x_{Teff},\ x_{IL2},\ x_{DAC}\}$.

Constant	Value
$q_{RestTreg}$	20
$q_{RestTeff}$	500
q_{NK}	100
C_{EBV}	1000
C_{Mem}	200
C_{DAC}	$(DACinjected)/log(.1)$ [2]
C_{IL2}	200
C_{Tcell}	200
C_{Teff}	200
p_{eff}^{dup}	2/3
p_{eff}^{mem}	1/3

Table 8.6: List of the constants regarding the first version of the RRMS model.

8.2.3 The spatial-temporal model

In this section we proposed an extension of the model introduced in the previous Section 8.2.2, to consider both the spatial and temporal aspects. In details, we included the spatial coordinates of all entities in a cubic tissue portion. This gives the opportunity to model more realistic scenarios, where different quantities of virus enter into the system from different directions. Moreover, we described how the intrinsic symmetries of the derived ESSN model may be automatically exploited to reduce the complexity of the analysis step. This allows to study models which are independent from the grid size, while with the classical approach it is hard to generate the ODEs system corresponding to the model with a $5 \times 5 \times 5$ grid. In this contest, we used the experimental module in $\mathcal{G}reat\mathcal{M}od$ to exploit the model symmetries as showed in Section 6.1.

Finally, in the first Paragraph **The ESSN model**, we report how the RRMS

[2] DACinjected represents the quantity of DAC injected per time and with this formula we estimate automatically the constant in order to have $exp(-DACinjected/C_{DAC}) = .1$, i.e. the T-cells duplication rate is reduced of the 90% when all the DAC particles are present.

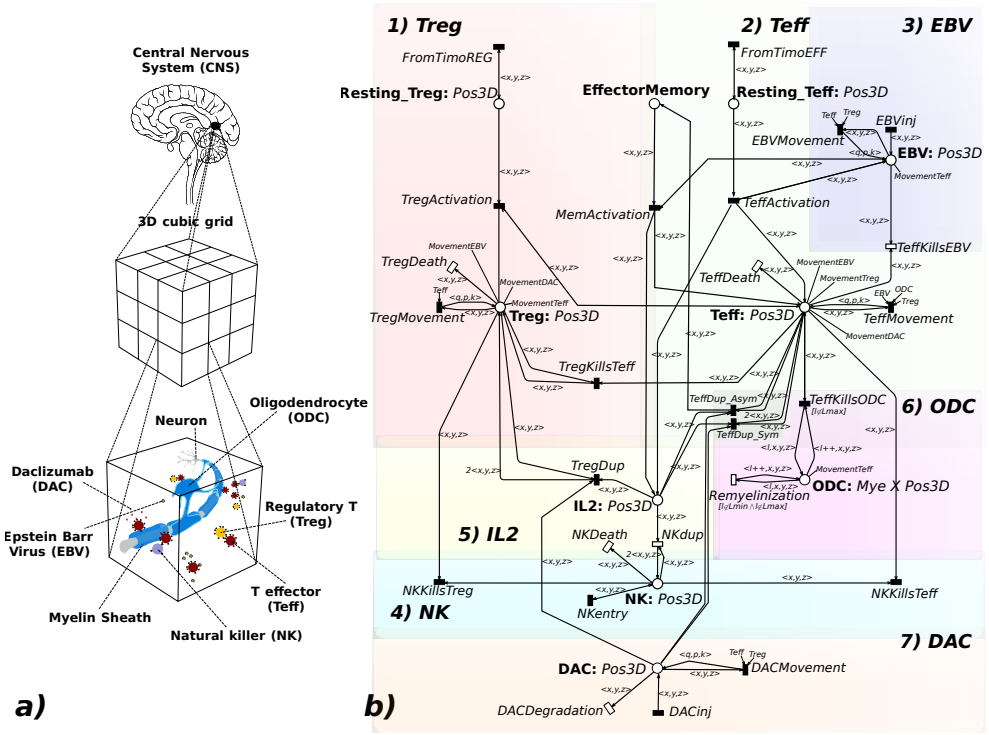


Figure 8.24: a) Representation of the three-dimensional model. b) The ESSN model representing the RRMS disease with cells movements.

model was extended to consider the spatiality, while the Paragraph **Results** focuses on the analysis of the model. The detailed expressions of the functions that encode the general transition velocities are reported in the Appendix Section A.3.1.

The ESSN model

In Fig. 8.24a) a portion of the CNS is depicted, showing: the neuron with its myelin sheath, and the 7 elements characterizing the MS disease distributed within a 3D cubic grid. The respective ESSN model is shown in Fig. 8.24b), consisting of 13 places and 25 transitions, and it is basically equal to the model depicted in Fig. 8.12 with the additions of transitions simulating the cells movements.

This model is organized in seven modules corresponding to the biological entities characterizing RRMS. The detailed description of each module is reported in Sec. 8.2.2. Briefly, the EBV module simulates the virus reactivation by means of a series of injections of virus particles in the system at given times,

while the Treg and Teff modules encode the activation of the T cells, the annihilation of the virus by the Teff action, the control mechanism of the Treg over the Teff. The NK module describes the killing of self-reactive Teff and Treg cells respectively, due to NK cells. The IL-2 module is focused on the IL-2 role. IL-2 is consumed by the Treg, Teff and NK functions and it is produced by the Teff activation. The ODC module describes instead the ODC behaviour, characterized particularly by the damage caused by Teff cells on ODC cells. Indeed, when the myelin level reaches the lowest value, an irreversible damage occurs and a remyelination of the neurons is no more possible. Finally, the DAC module encodes the drug administration and its pharmacokinetics inhibition of the expansion of Treg and Teff.

The model is characterized by four color classes: $PosX$, $PosY$, and $PosZ$ representing the coordinates of the position of a molecule in a 3D cubic grid; Mye encoding the myelination levels of ODC. Mye is divided into five static subclasses ranging from $Lmin$ (no myelination) to $Lmax$ (full myelination). Then, all the places except the ODC and EffectorMemory are characterized by the color domain defined as $Pos3D = PosX \times PosY \times PosZ$, i.e. the three-dimensional Cartesian product of the three coordinates color classes. Instead, the ODC place is characterized by the three coordinates plus the myelination levels, so that its color domain is $Pos3D \times Mye$. Finally, the EffectorMemory place has neutral color domain. Moreover, we assume that the EBV, Teff, Treg and DAC cells are able to move in all the cubic cells of the grid. Practically, the EBVs move uniformly in all the cells, the Teff cells move with higher probability towards a location in which there is higher concentration of EBV, and Treg and DAC cells move with higher probability towards a location in which there is higher concentration of Teff cells. Hereafter, the notation of the color combinations $\langle p_x, p_y, p_z \rangle$ and $\langle q_x, q_y, q_z \rangle$, representing the location coordinates, is simplified to $\langle \mathbf{p} \rangle$ and $\langle \mathbf{q} \rangle$, respectively. In particular, we define $x_{CellType_{\langle \mathbf{p} \rangle}}$ as the number of $CellType$ in the location $\langle \mathbf{p} \rangle$ at a specific time point. Hence, the movement functions can be defined as follows. The transition ***EBVMovement*** simulates the movements of EBV cells from point (with coordinates represented by the color combination) $\langle \mathbf{p} \rangle$ to point $\langle \mathbf{q} \rangle$. The speed of this movement (the rate of transition EBVMovement) is uniform in all directions and is captured in the following formula by assuming that the probability to move is equally distributed among all the grid cells.

$$f_{\langle EBVMovement,\mathbf{p},\mathbf{q}\rangle}(\hat{x}(\nu),\nu) = r_{moves} \cdot p^{EBV}_{\langle\mathbf{q}\rangle} \cdot x_{EBV_{\langle\mathbf{p}\rangle}}$$

where r_{moves} is a coefficient that we set equal to 0.1 in our numerical experiments. Differently, the transition ***TeffMovement*** simulates the movement of Teff cells from point $\langle\mathbf{p}\rangle$ to point $\langle\mathbf{q}\rangle$, and its speed is inversely related to the number of EBV cells in $\langle\mathbf{p}\rangle$ (since more the virus in $\langle\mathbf{p}\rangle$ less the Teff cells are tempted to leave the position) and depends on the number of EBV in $\langle\mathbf{q}\rangle$ (a greater number of EBV cells leads to a higher probability to move into that location). This is captured by the following formula

$$f_{\langle TeffMovement,\mathbf{p},\mathbf{q}\rangle}(\hat{x}(\nu),\nu) = r_{moves} \cdot (exp(-\frac{x_{EBV_{\langle\mathbf{p}\rangle}}}{C_{EBV}})) \cdot p^{Teff}_{\langle\mathbf{q}\rangle} \cdot x_{Teff_{\langle\mathbf{p}\rangle}}$$

where r_{moves} is again set equal to 0.1; the second term of the function, defined as $exp(-\frac{x_{EBV_{\langle\mathbf{p}\rangle}}}{C_{EBV}})$, varies in the interval $[1, 0)$, simulating the decreasing of the movement velocity with respect to the number of EBV cells present in the starting point; $p^{Teff}_{\langle\mathbf{q}\rangle} = \frac{x_{EBV_{\langle\mathbf{q}\rangle}}}{EBV_{tot}}$ represents the probability to move in the cell with coordinates $\langle\mathbf{q}\rangle$ where EBV_{tot} is the total number of EBV in the grid at time ν; and C_{EBV} is an experimental constant that we set equal to 1000. All these quantities are functions of the time ν which is omitted in the formula to keep the notation simpler.
Transitions ***TregMovement*** and ***DACMovement*** represent the movements of the Treg and DAC cells (respectively) from point $\langle\mathbf{p}\rangle$ to point $\langle\mathbf{q}\rangle$. Similarly to what explained for transition TeffMovement, their speeds are inversely related to the number of Teff and T (= Treg+Teff) cells in $\langle\mathbf{p}\rangle$ and depend on the number of Teffs and Ts in $\langle\mathbf{q}\rangle$.

Results

We studied the RRMS considering a tissue portion explicitly modeled through a cubic grid consisting of 27 cubic cells (Fig. 8.24a)). To achieve this, we defined the color classes $PosX = \{x_1, x_2, x_3\}$, $PosY = \{y_1, y_2, y_3\}$ and $PosZ = \{z_1, z_2, z_3\}$. For all the simulations, we assumed 500 ODC with level L_{max} of neuronal myelinization, 1687 resting Teff cells, 63 resting Treg cells, 375 NK cells and 1000 IL2 molecules, and zero cells in the other places.This model is equivalent to a system of 433 ODEs, but with few assumptions it is possible to derive the corresponding reduced $\widehat{ODE}$s system including only the symbolic equations.

In details, let us define the set of all the 27 location coordinates as $\mathbf{P} = \{\langle p_x, p_y, p_z\rangle : \ p_x \in PosX, p_y \in PosY, p_z \in PosZ\}$. Then, we consider three disjoint subsets of $\mathbf{P}$, namely $P1,\ P2,\ P3$; the first two correspond to grouping the EBV and DAC injection locations, respectively, while the third *P3* groups all the remaining locations. For simplicity, and to maintain the symmetries into the system as well, the EBV and DAC injection locations do not change over the simulation time and do not overlap. Given this, it is possible to derive the $\widehat{ODE}$ system characterized by 49 equations only. Indeed, the 433 original equations can be partitioned into 49 groups of similar equations. Each group is expressed in the reduced model by one representative equation. The grouping derives from the observation that the behaviors of the modeled elements do not depend on their actual positions, but only on the presence of the EBV and/or DAC cells. When the grid size grows the number of groups does not change, as long as the number of locations where different quantities of EBV and/or DAC cells appear is fixed; instead the size of each group of equivalent ODEs increases with the grid size. A further reduction is represented by the number of terms in each $\widehat{ODE}$, representative of each group of ODEs, with respect to the number of terms appearing in the ODEs in the equivalence class. This reduction is due to the factorization obtained by exploiting symmetries. Other examples are reported in Table 8.7, where the R file dimension and the number of differential equations of the complete and reduced models are compared considering different cubic grid dimensions, from $3\times3\times3$ to $5\times5\times5$. It is easy to see that an increasing number of locations is associated with an increase in the number of ODEs and of the R file containing them, while the $\widehat{ODE}$ system does not change. Note that when the $5 \times 5 \times 5$ grid is considered, the ODEs generation procedure fails because it exceeds the available memory. The advantage can also be observed from the point of view of the simulation time, we obtained a speed up from 8.927205 hours to 12.76043 secs on an Intel Xeon processor @ 2GHz, by using one core. Note that the simulation was performed considering $3\times3\times3$ cubic grid, one year interval and assuming EBV injections at regular times (every two months), and each injection introduces into the system 10000 EBV copies.

Number of locations	R File dimension ODEs / $\widehat{ODE}$s	Number of ODEs / $\widehat{ODE}$s
27 (3x3x3)	0.43 MiB / 0.023 MiB	433 / 49
64 (4x4x4)	5.0 MiB / 0.023 MiB	1025 / 49
125 (5x5x5)	Out of memory / 0.023 MiB	2001 / 49

Table 8.7: Comparing the ODE and $\widehat{ODE}$ system, varying the cubic grid dimension.

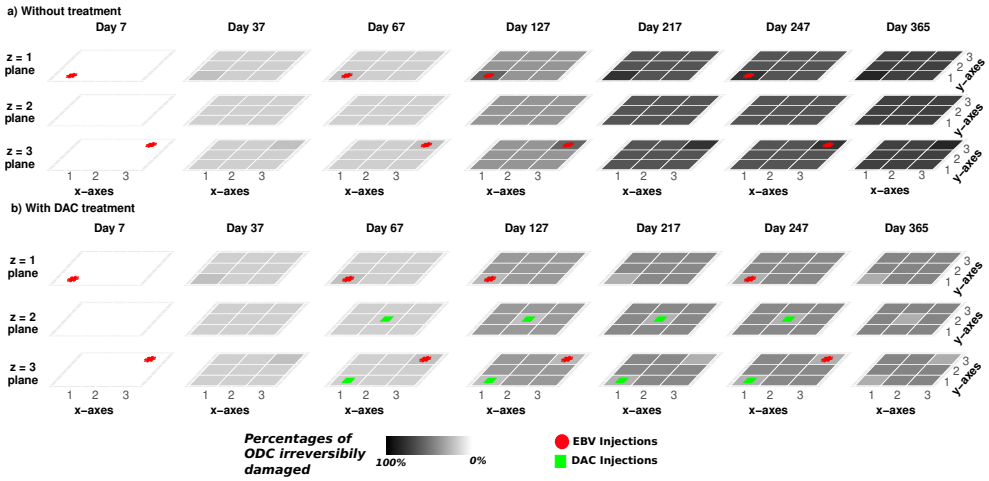


Figure 8.25: Percentages of ODCs irreversibly damaged a) without and b) with DAC treatment.

A possible evolution of the system is shown in Fig. 8.25, where the red circles represent the location in which EBV is injected. For each plot, the three rows represent the z-planes and the columns refer to the time points in which the injections are done. Fixing the time point and the z-plane, the corresponding 3×3 square reports the number of ODCs damaged into the nine grid cells obtained varying the x and y coordinates. As expected, the panel *a* of Fig. 8.25 shows the progressive accumulation of ODC irreversibly damaged until day 365. Instead, in panel *b* of Fig. 8.25 is reported the results of the simulation of the DAC effect. In details, every month after two months of simulation, two injections are simulated (green squares) introducing 300 DAC copies for each administration. These results agree with those proposed in Section 8.2.2 since the number of irreversibly damaged ODCs decreases in the case with DAC administration with respect to the case in which no drug is injected. With DAC the percentage of irreversibly damaged ODCs ranges from 28% to 45%, while with no DAC the number of irreversibly damaged

ODCs is between 70% and 85%.

8.2.4 The temporal model: exploiting real data

In this section, we extended the first model version presented in Section 8.2.2 by including different actors of the pathogenetic mechanisms of MS, as the blood brain barrier and some cytokines produced by self-reactive Teff cells that in MS overcome the BBB and reach the CNS where they damage ODCs [148, 42].
Several evidences suggest the association of viral (and, to a lesser extent, also bacterial) infections with disease onset and relapses leading to hypothesize that RRMS course could be related to a reactivation of a latent infection [148, 110]. EBV, Human Herpes Virus 6 (HHV-6), Varicella Zoster Virus (VZV), and Human Endogenous RetroViruses (HERVs) infections can cause the activation of auto-reactive Teff cells against the CNS[148, 110, 8] through a mechanism called "molecular mimicry". This is the most frequently discussed mechanism for how viruses or bacteria could induce autoimmunity in MS, that occurs when peptides from pathogens (antigens) share sequence or structural homology with host peptides (self-antigens), in the case of MS with CNS antigens (e.g. Myelin Basic Protein, Myelin oligodendrocyte glycoprotein) [90]. When Teff cells encounter such a foreign peptide, they produce IFNγ and IL-17, potent inflammatory mediators able to increase the inflammatory micro-environment. In a healthy host Teff response is tightly regulated by Treg cells to mediate effective host defense against pathogens without causing excessive tissue damage. Furthermore, Treg cells play an important role in maintaining peripheral tolerance to self-antigens. In MS patients, low number and impaired function of Treg cells [179] could explain the massive production of IFNγ and IL-17, during the exacerbation phases of MS. In this context, NK cells also give a contribution acting as first-line defense against viruses and bacteria and regulating the auto-reactive Teff cells activity, producing the pro-inflammatory cytokine IFNγ and the anti-inflammatory cytokine IL-10 [121, 87].
Current therapeutic strategies for MS are now focusing on the (i) reduction of the risk of relapses avoiding accumulation of disability and (ii) identification a trade-off between drug efficacy and side effects [106]. DAC was selected for its effects on the depletion of Teff cells, concomitant with a reduction in the number of Treg cells and the expansion of a particular NK cell subset called

CD56bright NK cells [61].

In this section we first described the computational model developed to study RRMS and then the data exploited to calibrate the model parameters; in particular two parameter configurations are identified for healthy and MS subjects. Finally, we discuss two therapy scenarios in which we investigated the effect of the DAC administration on MS patients. All the analyses are performed on a server with 6 Intel Xeon E5-2650 processors (2.00Ghz, 20MB Cache, 8 Cores) by exploiting $\mathcal{G}reat\mathcal{M}od$.

The ESSN model

The cellular interactions characterizing the immunopathology of RRMS are described by the model showed in Fig. 8.26. The model consists of 26 places and 55 transitions (in details, 40 are standard transitions and 15 are general transitions). Observe that to make immediately clear the biological role of each place in the net, we decided to represent them using four different icons whose meaning is reported in the legend of Fig. 8.26. The respectively ESSN model is showed in Fig. A.1 in the Appendix. In details, the model is divided in two compartments: the peripheral lymph node/blood vessel and the CNS. The two compartments interact with each other through the place BBB, as illustrated in Fig. 8.26. All transitions describing interactions occurring in peripheral lymph node/blood vessel have the suffix "_out" while all transitions taking place in CNS have the suffix "_in". The Antigen place, simulates the first pathogen infection and then the infection reactivation in the system through the *AntigenInjection* transition.
In the peripheral lymph node compartment of the net the places regarding the Teff cells, including Effector Memory T cells, are represented. After pathogen infection Teff cells can give rise to Teff or Effector memory cell through *TeffDup_Sym_out* and *TeffDup_Asym_out* transitions. Effector memory cells remain in this compartment and are able to respond faster to the infection reactivation. The annihilation of the pathogen by the Teff action is modeled by the transition *TeffKillsA*, while symmetrical and asymmetrical duplication of Teff cells is encoded by transitions *TeffDup_Sym_out* and *TeffDup_Asym_out*. Teff production of the inflammatory cytokines IL-17 and IFNγ is represented by the transitions *Teff_prod_IL*$_{17}$ and *Teff_prod_IFNg*, respectively. Inflammatory cytokines are degradated by *IFNgConsumption_out*

and *IL17Consuption_out* transitions. Afterwards, the arrival of new resting Treg cells from thymus, the control mechanism of the Treg over the Teff and their activation, proliferation and death are encoded by the transitions *FromTimoReg*, *TregKillsTeff_out*, *TregActivation_out*, *TregDup_out* and *TregDeath*, respectively. Treg production and degradation of the anti-inflammatory cytokines IL-10 are represented by the transitions *Treg_prod_IL*$_{10}$*_out* and *IL10Consuption_out*. Through the transition *NKarrive* the arrival of a NK cells able to kill self-reactive Teff is simulated. The killing is simulated by the transition *NKKillsTeff_out* and the NK production of IFNγ and IL-10 is represented by the transitions *NK_prod_IFNg* and *NK_prod_IL10*, respectively. The death and proliferation of the NK cells are modeled by transition *NKDegradation* and *NKdup*.

The drug administration of daclizumab (DAC) is modeled by transition *DACinjection*, and its degradation by *DACDegradation*. DAC is able to inhibit the expansion of Treg and Teff through the transitions *DACkillTeff* and *DACkillTreg*. During the relapsing phases of the disease the BBB increases its permeability leading to the passage of Teff and Treg from peripheral blood to CNS. This biological process is encoded by the place *BBB* and transitions *Teff_pass_BBB*, *Treg_pass_BBB*, *IL10_BBB* and *IL17_BBB*.

In the CNS the Resting_Teff_in are activated through the transition *TeffActivation_in*. The ODC damage due to Activated Teff cells is modeled by *TeffKillsODC* whose its fire decreases the myelination levels of ODC s by acting on the colors of the tokens in place ODC[3]. When the myelin level reaches the lowest value, an irreversible damage occurs since the remyelinization of the neurons modeled by the transition *Remyelinization* is no longer possible. Finally, transition *TeffDup_Sym_in* simulates the Teff cells proliferation. Teff production of the inflammatory cytokines IL-17 and IFNγ is represented by the transitions *Teff_prod_IL*$_{17}$*_in* and *Teff_prod_IFNg_in*, respectively. The inflammatory cytokines are degraded by *IFNgConsumption_out* and *IL17Consuption_in* transitions. Tregs in CNS (*Resting_Treg_in* and *Treg_in*) product and degrade IL-10 thought *Tref_prod_IL10_in* and *IL10Consuption_in*. Their control mechanism on Teff their activation, proliferation and death are encoded by following transitions: *TregkillsTeff_in*, *TregDup_in* and *Treg-*

[3]Observe that the tokens in ODC are colored according to color class *Mye* that is divided into five static subclasses ranging from *Lmin* (no myelination) to *Lmax* (full myelination) of the color class

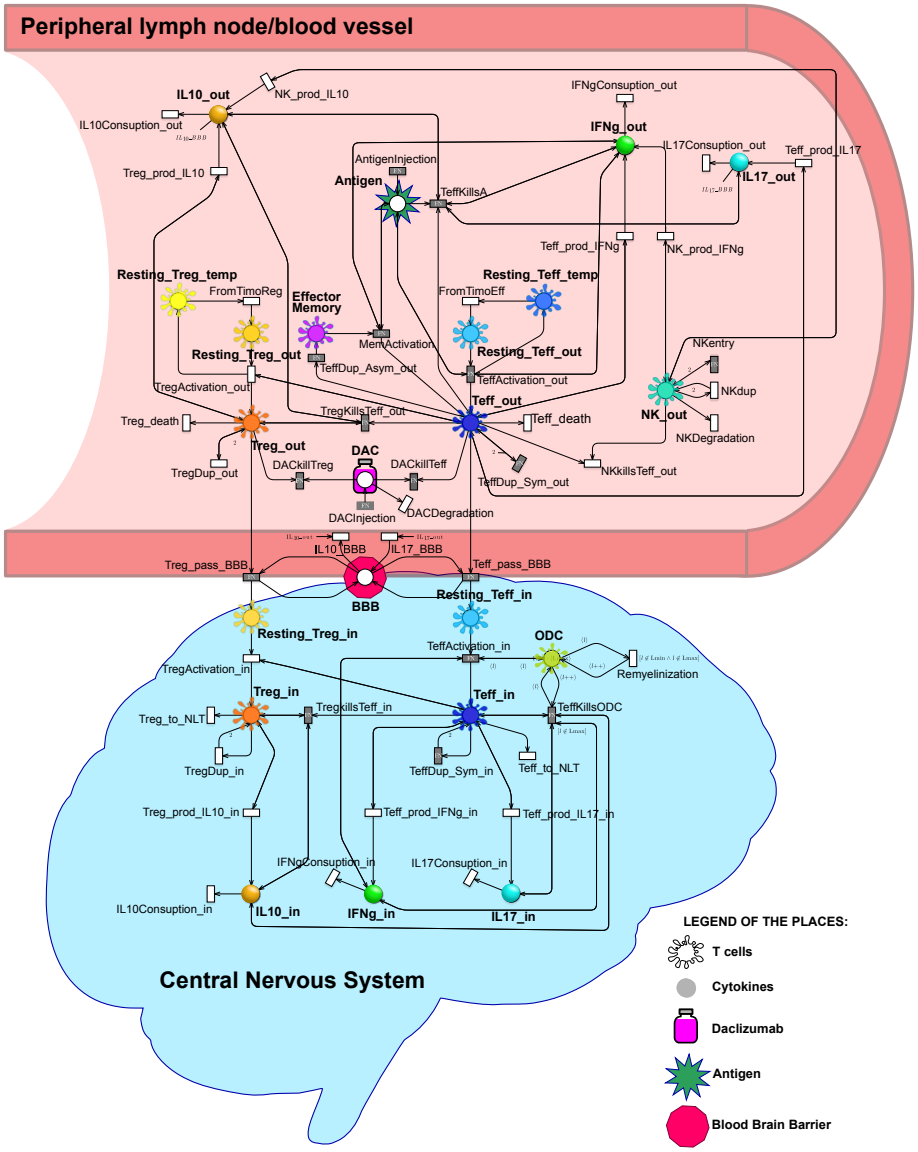


Figure 8.26: RRMS model represented by exploiting the ESSN graphical formalism. Each place has on top a sketch of its biological role as described by the legend.

_to_NLT.

Model calibration

The model calibration was performed on our model to make its behaviours in agreement with the experimental values described above. Let us recall that the mathematical details of how this step is implemented are reported in Sec. 7.1.1. In details, during the calibration analysis the same initial marking was assumed for both healthy donors (Healthy Donor (HD)s) and MS patients, as showed in Table A.6, where only the places with initial marking different from zero were reported. Observe that these values correspond to the average values computed considering only the HDs reported in Table A.5. Moreover, we considered all the 500 ODC with level $Lmax$ of neuronal myelinization at the initial time.

Then, from the model shown in Fig. 8.26 a system of 26 ODEs composed by 20 unknown parameters was derived. These parameters (reported in the appendix Table A.9) were characterized by a high uncertainty due to their difficulty of being empirically measured. To identify the parameter values, the ODE system was simulated over 30 days interval assuming a injection of 100 antigen copies at the second day. This analysis lead to identify the best fit characterized by the values described in the Appendix Sub Section A.3.2. In particular, the parameter values for HD and MS patient were identified by minimizing the difference among the numbers of IFNγ-producing, IL-17-producing IL-10-producing cells in blood and in CSF obtained from the solution of the ODE system and those experimentally measured in average for HDs and MS patients (reported in Table A.5) after 18 hours from the antigen injection. Since the real data were available at a specific time point, it was not possible to completely validate the model over the whole simulation time window against observed data. Thus, the parameters were chosen such that the model outcomes are in agreement with real data and the current knowledge of the biological system.

It is worth noting that the two sets of parameter values, respectively for HD and MS patients, obtained by the calibration step, differ only in the values of the parameters $p_{Treg\_Activation}$ and $p_{Teff\_Activation}$ associated with the transitions $TregActivation\_in(\_out)$ and $TeffActivation\_in(\_out)$.

The model behaviours derived by these two parameter configurations are reported in Fig. 8.27. From these plots it is immediately clear that the model outcomes are in agreement with real data and the current knowledge of MS disease, for example the MS patients are characterized by a higher number of

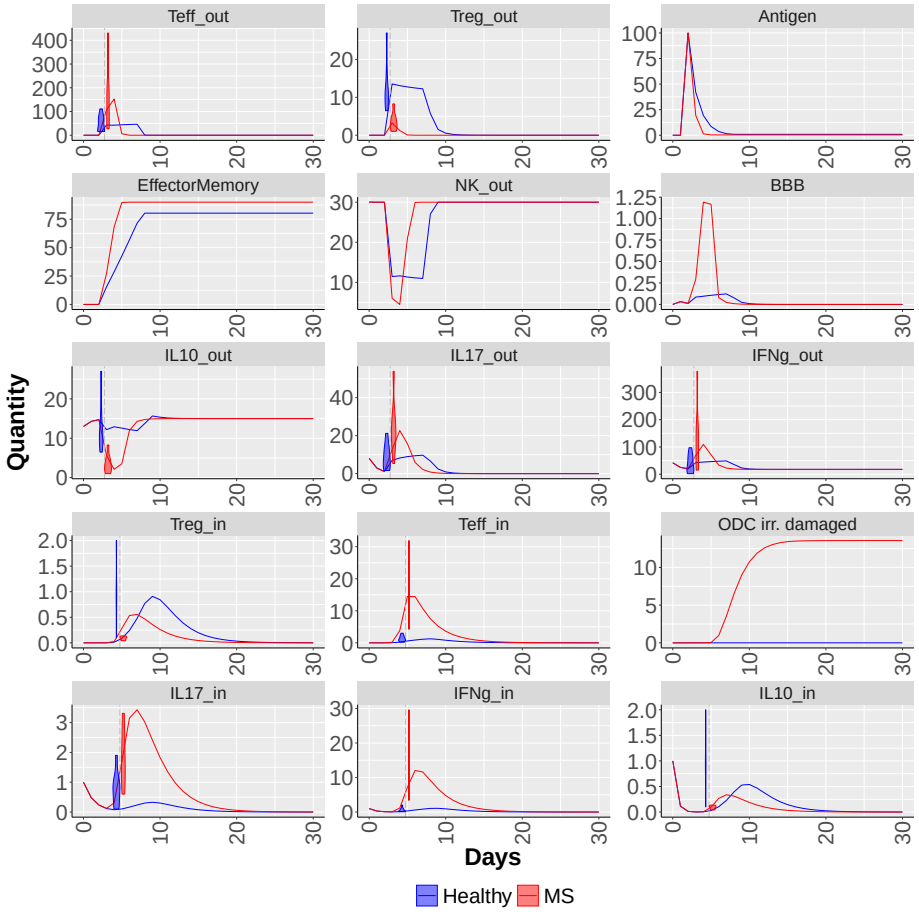


Figure 8.27: Deterministic solution of the ODEs system considering both the parameters combinations, red the MS patient and blue the healthy individual. The violin plots are the representation of the real data.

irreversibly damaged ODC cells and a more impermeable BBB than HDs.

Stochastic simulations. After the model calibration, the two parameter configurations were exploited to investigate the stochastic behaviour of the system using the SSA algorithm. In Fig. 8.28, 1000 trajectories for each scenario are plotted in grey, while the colored bold line represents the mean trajectory. The distribution of experimental data (Table A.5) is represented through violin plots. It is possible to observe that the mean trajectory and the set of 1000 stochastic trajectories are consistent with the experimental measures in both HD and MS patients. In particular, observing the 1^{st} row in Fig. 8.28, it is possible to appreciate the differences in the number of circulating Treg cells

between HD and MS patients. Accordingly, while the antigen is counteracted by the Teff cells, Treg cells try to balance the aggressiveness of the immune system and to maintain the cellular homeostasis by acting as a brake on the inflammatory response producing IL-10and killing Teff cells. In the 2^{nd} row in Fig. 8.28 the IL-10 dynamics are reported: the MS condition produces less IL-10 with respect to the healthy counterpart. IL-10 cytokines imbalance towards an inflammatory state is reflected in an increased permeability of the BBB. IL-10 contributes to promote BBB integrity, while IL-17 contributes to the damage of BBB, making it permeable to the passage of cells of the immune system and other molecules. Indeed, after less than one week we can observe that in the MS patient BBB has the highest permeability (3^{rd} row in Fig. 8.28) leading to a T cell trafficking in the CNS. This effect is observed in the form of an increased reactivation of Teff cells in the CNS of the MS patient compared to the healthy subject (4^{th} row in Fig. 8.28), as well as in the increase in the circulation of pro-inflammatory cytokines produced by the Teff cells circulating in the CNS: IL-17 (5^{th} row Fig. 8.28) and IFNγ (6^{th} row Fig. 8.28). The final result of this pro-inflammatory environment in the CNS is an increased damage to the ODCs, which simulates a neuronal damage to the myelin in the CNS actually observed during the clinical relapse (7^{th} row and second column Fig. 8.28). Conversely, ODC damage is not observable in HDs (7^{th} row and first column Fig. 8.28). The dynamics of the other cells are reported in Fig. 8.29.

Drug therapy

To investigate the effect of the DAC therapy in our model calibrated for RRMS patients, we simulated a real scenario with multiple antigen occurrences at different time in two-years time interval. In details we assumed a total of eight injections introducing into the system 100 antigen copies per injection. Most of the MS-related viruses, infect a particular type of immune cells (i.e. the EBV infects B cells) and once the initial lytic infection is brought under control, the virus persists in the immune repertoire of a subject in a state called "of latency" for the rest of his/her life and is subjected to periodical reactivation [8]. For this reason, we defined a sequence of injections at varying intervals: the first three injections were set at constant time interval, i.e., the 2^{nd}, 67^{th} and 127^{th} day, then four consecutive injections were simulated at 295^{th}, 300^{th}, 303^{th} and 307^{th} days, and finally the last injection at 600^{th}

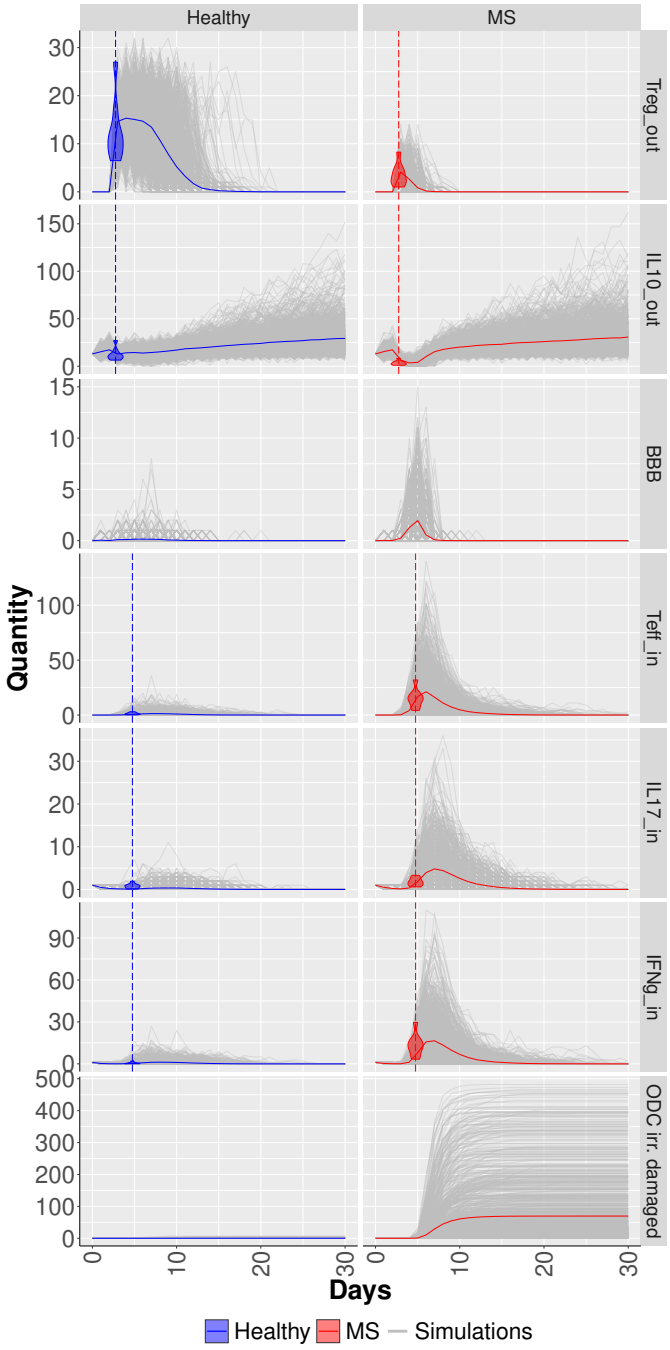


Figure 8.28: 1000 stochastic simulations considering the healthy (first column) and the MS (second column) parameters configuration. The colored bold lines represent the mean traces of the simulations, blue for the healthy and red for the MS scenario. The violin plots are the representation of the real data.

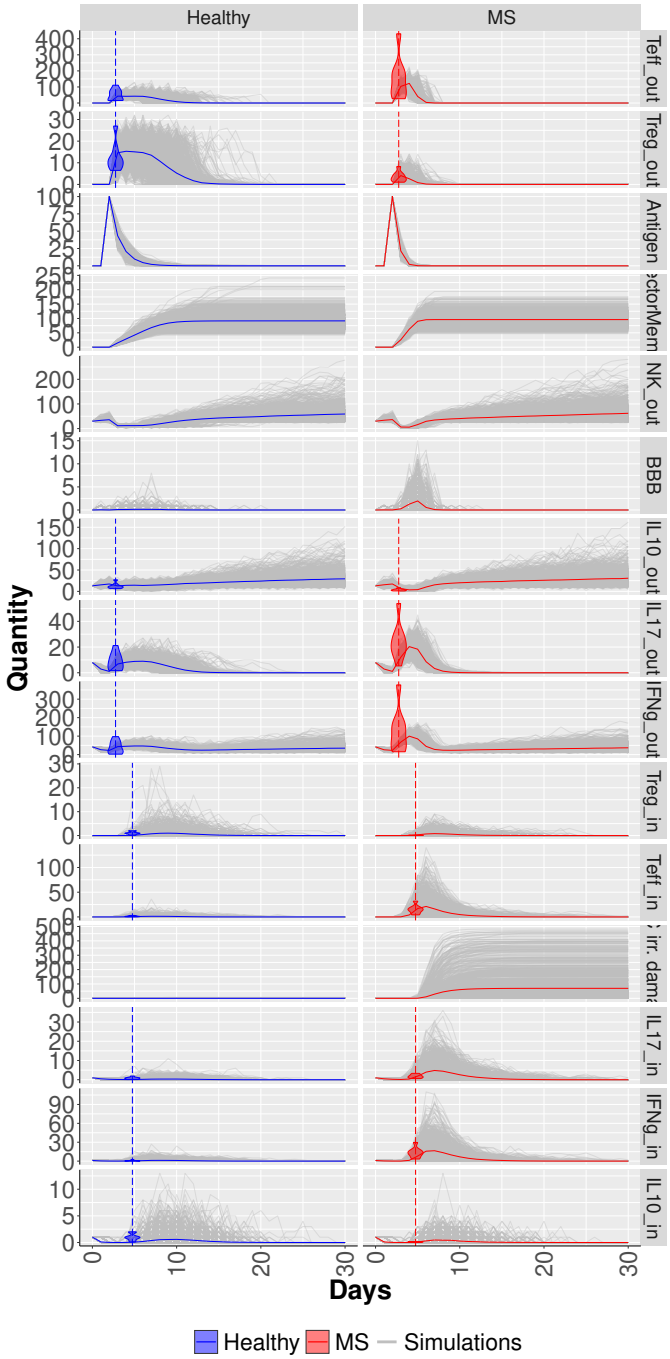


Figure 8.29: 1000 stochastic simulations considering the healthy (first column) and the MS (second column) parameters configuration. The colored bold lines represent the mean traces of the simulations, blue for the healthy and red for the MS scenario. The violin plots are the representation of the real data.

days. Two important aspects of the therapy modulation were considered in our simulations: the drug dose and the drug potency. In details, five scenarios characterized with an increasing drug dose (i.e., 1000, 2000, 5000, 10000 and 15000) were analysed. Then for each scenario two drug potencies (i.e. *weak potency* and *strong potency* characterized by $DACkillTreg$ and $DACkillTeff$ set to 0.01 and 0.03, respectively) are showed for a total of ten different scenarios. These scenarios are proposed in two course of actions: *early regime therapy* in which the DAC administration starts at the first month, and *late regime therapy* in which the DAC administration starts at the sixth month.

In Fig.s 8.30 and 8.31 the behaviour of the places *A*ntigen, Teff_out, IFNg_out, IL17_out, Treg_out, IL10_out, NK_out, BBB, and ODC irreversibly damaged in the early and late regime are reported. In each figure, the columns represent nine scenarios among the twelve described above. Specifically, we decided to omit for clarity the healthy scenario and the ones considering drug dose equals to 2000. The colored bold lines represent the median of 1000 simulations, while the colored areas are the range of the simulations between the first and third quartile. In the Appendix Fig.s A.2 and A.3, the complete list of cell types for all the scenarios in both the regime considered are reported.
In both Fig. 8.30 and Fig. 8.31, the reduction of circulating Teffs and Tregs is visible in all treatment conditions (the 2^{nd} and 5^{th} rows), with a remarkable effect at increasing doses. The same effect is visible in the amount of cytokines produced by T cells (the 3^{rd}, 4^{th} and 6^{th} row). Although the DAC is unable to cross the BBB and spread in the CNS, these effects are observable either in the blood and in the CNS (Fig.s A.2 and A.3). This is due to a reduced number of T cells that are not able to effectively reach the CNS and to cause damage. Indeed, the immunosuppressive action of DAC is mediated by the binding to CD25 molecules of IL-2R, present on activated Teff cell and on Treg cells, and results in the inhibition of their proliferation and in the induction of T cell death.

Fig. 8.32 shows the number of irreversibly damaged ODC (the blue contour boxplot) and the overall antigen concentration (the red contour boxplot) for each scenario. In details, it is showed the difference in the total amount of irreversibly damaged ODCs at the end of the two years, a behavior reflecting the protective effect of DAC therapy on the CNS.
It is observed that for the same dose and potency of DAC, the number of irre-

versibly damaged ODCs is lower in "early therapy" than in the "late therapy" condition (Fig. 8.32). Indeed, clinical practice suggests that, in MS, the early intervention reduce neuronal damage and long-term disability [106].
Moreover, an interesting effect of DAC therapy visible by our simulations is that increasing dose and potency of DAC can suppress the immune system by depleting T cells and cytokine diffusion. An overdose of the drug reproduces an immuno-compromised immune system, where T cells are depleted and the antigen persists in the circulation. This is visible in Fig. 8.32 where the red contours boxplots report the antigen concentration. It is straightforward to see that increasing drug dose is associated with a minor ODC damage at the expense of the antigen annihilation. Moreover, a stronger drug potency is not positive either from the ODC damage and the antigen annihilation point of view. The ideal dose of DAC is a trade-off between antigen annihilation and a reduction in damaged ODC. Drug efficacy must occur with a consideration of the protection from external antigens to minimize the risk of secondary infections, sometimes with serious adverse effects, that often accompany immunosuppressive therapies in MS [130].

Finally we can conclude that in our model, following the administration of DAC, a reduction of Teff cells activation was observed. Furthermore, this reduction was visible for Treg cells too, which normally expose the CD25 on their surface and use it to sequester IL-2 to Teff cells which result in an indirect inhibition mechanism of Teff cells by IL-2 competition [144]. On the other hand, NK cells, trying to make up for the lack of T cells, undergo a selective expansion, as expected [61].
We observed that a dose below 1,000 DAC molecules leads to an insufficient protection from relapses (i.e. ODC damage), while a dose above 10,000 DAC molecules totally depletes the pool of T cells, impairing the mechanisms of protection from external antigens, as shown by the increase of antigen in the system. The right dosage in our model should be set to a value between 1,000 and 10,000 DAC molecules, corresponding to the actual therapeutic range of 150-300 mg per four weeks consistently with the idea that immunosuppressive therapies cannot totally imbalance immune cell homeostasis, as a minimal level of immunesurveillance should be always mantained in order to clear phatogens. Moreover, simulation results about to the comparison between early and late interventions support the current guidelines of therapy for MS, in which is

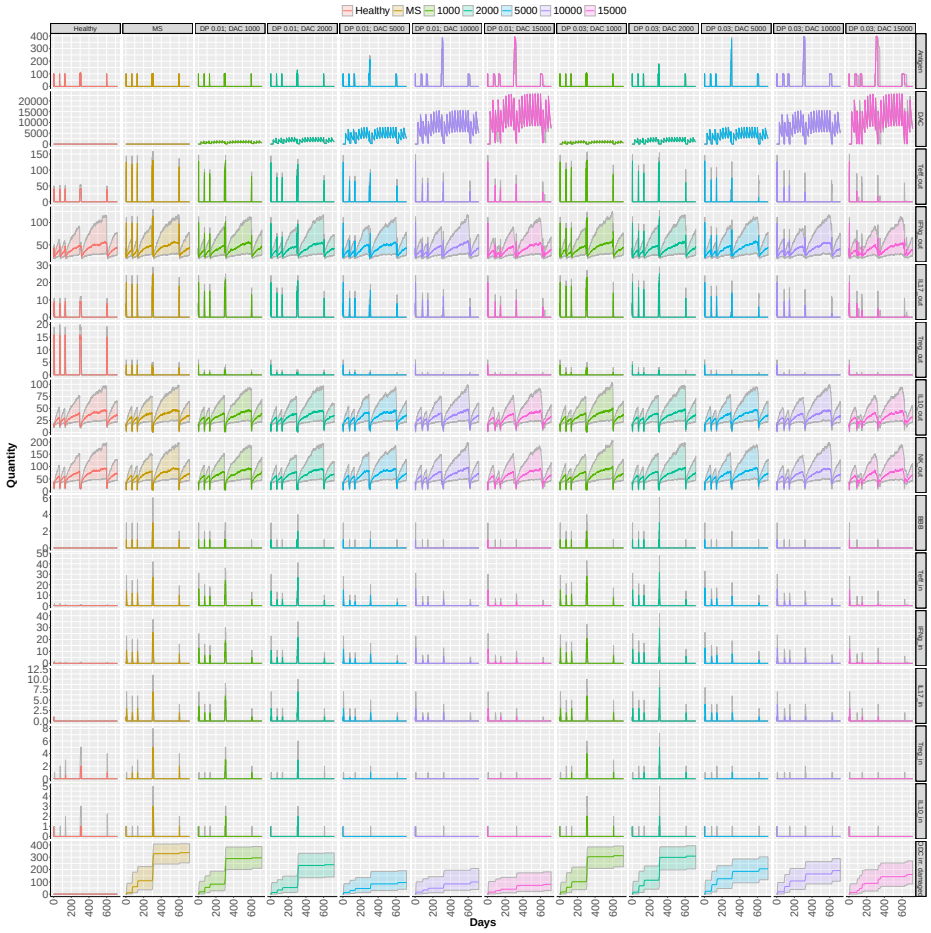


Figure 8.30: Stochastic simulations considering the *early therapy*. Different colors are associated with quantity of DAC injected for each scenario, from 1000 to 15000 cells. The first two column represent the healthy and MS scenarios. Two drug potencies (called DP) are showed, i.e., 0.01 e 0.03.

suggested that early intervention is crucial for minimize the accumulation of disability [106].

8.2.5 Discussion

In immunology the use of computational modeling is quite recent, but it is becoming increasingly important. In particular, the computational models can help the researchers to discern between potential right and wrong biological hypotheses, whose confirmation cannot be acquired through *in vivo* or *in vitro* experiments, to find novel treatments, to validate or deduce the

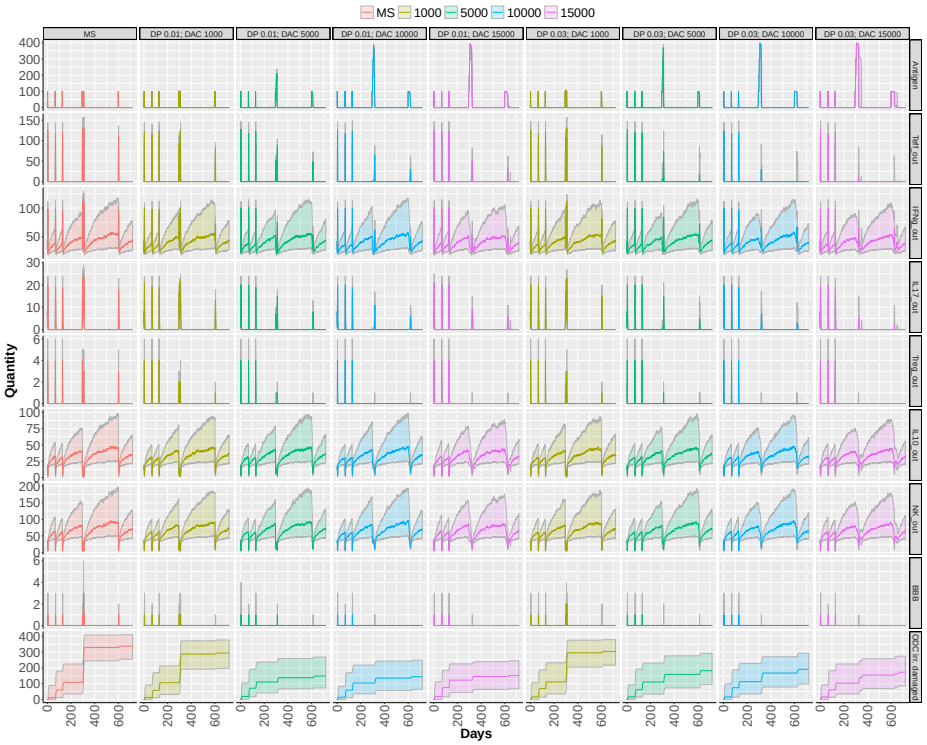


Figure 8.31: Stochastic simulations considering the *late therapy*. Different colors are associated with quantity of DAC injected for each scenario, from 1000 to 15000 cells. The first two column represent the healthy and MS scenarios. Two drug potencies (called DP) are showed, i.e., 0.01 e 0.03.

mechanisms of actions of existing ones, and to optimize timing and dosage of treatments. The use of such modeling approaches is gaining attention also by regulatory agencies that are starting to foster their application also in the field of personalized-medicine.

In this context, the construction of mathematical models and their solutions remain a challenging tasks mainly due to the lack of general framework easily accessible even by researchers without advanced modeling and mathematical skills. To deal with this aspect, we showed how our framework, $\mathcal{G}reat\mathcal{M}od$, can provide a friendly environment for the modeling and the analysis of the RRMS disease. Indeed, RRMS represents a very challenging case study, due to the complexity of the disease which involves many different biological agents, ranging from molecular to environmental factors. Therefore, by exploiting the descriptive power of ESSN to provide a graphical representation of the complex biological system in a compact and parametric way, we were able

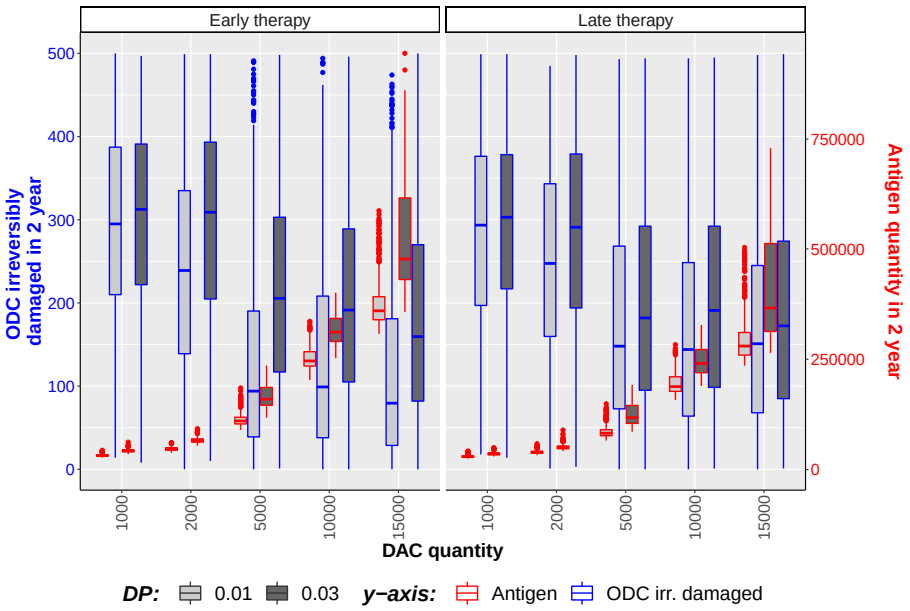


Figure 8.32: Box-plots representing the 1000 trajectories for each type of i) therapy (i.e., early and late therapy, and the healthy an MS cases without therapy), ii) quantity of DAC injected, and iii) the drug potency. The ODC irreversibly damaged at the end of the two years are plotted with the blue boxplot contour (referring to the left blue y-axis), differently the areas under the antigen curve over the whole two years interval are showed with the red boxplot contour (referring to the right red y-axis).

to easily modified the same ESSN model to simulate different scenarios and biological systems characterizing the RRMS disease. In details, we started from an ESSN model composed only by the biological entities which play the key role in the immune system. Hence, we showed the ability of the model to reproduce the typical oscillatory behavior relating to the onset of RRMS by supposing a breakdown of the cross-balance regulation mechanisms at the peripheral level. Moreover, considering different what-if analysis by simulating a specific therapy, such as with Daclizumab, it might help scientists to define the mechanisms of actions of this drug and to theorize the possible causes of its observed side-effect on the patients. While, the experiments simulating RRMS in pregnant women can contribute to define the mechanisms at the basis of the variation of the Treg and Teff cells.
Successively, we studied the RRMS considering a tissue portion explicitly modeled through a cubic grid. Thus, we proposed to extend the model in-

troduced above by adding color classes to model the different positions that each cell may have. Indeed, this study was a perfect example to show how the model symmetries can be exploited to reduce its complexity. In this case, we were able to reduce the original ODEs system composed by 433 equations to an equivalent symbolic ODEs ($\widehat{ODE}$s) system characterized by 49 equations only. A further reduction is represented by the number of terms in each $\widehat{ODE}$, representative of each group of ODEs, with respect to the number of terms appearing in the ODEs in the equivalence class. Finally, the results obtained by simulating this model agree with those proposed in previous section, since the number of irreversibly damaged ODCs decreases in the case with DAC administration with respect to the case in which no drug is injected.
In the last version, we decided to further extend our first model (the one introduced in Section 8.2.2) by including different actors of the pathogenetic mechanisms of MS, as the blood brain barrier and some cytokines produced by self-reactive Teff cells that in MS overcome the BBB and reach the CNS where they damage ODCs [148, 42]. Interestingly, we identified two parameters configurations representing respectively an healthy and MS patient, which configuration differed just in the two parameters associated to the activation of Teff and and Treg cells. By increasing the Teff and reducing the Treg ones, we were able 1) to fit the real data and 2) to represent the Teff-Treg cell balance characterizing the healthy subject and its unbalance in the MS patient. In particular, following the administration of DAC, a reduction of Teff cells activation was observed. Furthermore, this reduction was visible for Treg cells too, which normally expose the CD25 on their surface and use it to sequester IL-2 to Teff cells which result in an indirect inhibition mechanism of Teff cells by IL-2 competition [144]. On the other hand, NK cells, trying to make up for the lack of T cells, undergo a selective expansion, as expected [61].

In conclusion, some aspects of our model can be further improved and others have to be implemented in order to assert its reliability in representing the disease course even in presence of other disease modifying therapies. More specifically, the success of therapies tailored against B cells in MS (e.g. ocrelizumab) has shown how B cells can contribute to the pathogenesis of MS [55], that for a long time was erroneously attributed to T cells only. Also DAC itself has a reduction effect on the B cell population in MS [55]. It is worth mentioning that the role of B cells in MS seems to relate on both antibody-

dependent and antibody-independent functions of these cells [8]. Antibody-independent functions are represented by the presentation of the antigen to T cells and modulation of T cell function by secreting pathogens and / or protective cytokines in the CNS [55]. Indeed, the secondary reactivation in the CNS is known to be mediated by contact between antigen-presenting cells (e.g. B cells) and self-reactive T cells. These important aspects of the relationship between B and T cells in MS pathogenesis are worth to be investigated. Moreover, the inclusion of different components and cell types in the model will give us the opportunity to model the action of other therapies besides DAC.

8.3 COVID-19

In this section we showed how $\mathcal{G}reat\mathcal{M}od$ framework was exploited for the last case study regarding the recently coronavirus epidemics considering the Piedmont region. In particular, we used the ESSN formalism to model an extended version of the Susceptible-Exposed-Infected-Removed-Susceptible (Susceptible-Exposed-Infected-Recovered-Susceptible (SEIRS)) model accounting for population age structure. The infectious population is divided into three subgroups: *(i)* undetected infected individuals, *(ii)* quarantined infected individuals and *(iii)* hospitalized infected individuals. Moreover, the strength of the government restriction measures and the related population response to these are explicitly represented in the model. More details regarding the model are reported in Section 8.3.2. Successively to the model definition, in Section 8.3.3 we showed how the calibration phase was carried out to fit the model outcome with the infection and death data from February 24^{th} to May 2^{nd} 2020. Given the good level of accordance with the available data we proposed in Section 8.3.4 multiple what-if analysis to investigate different scenarios of the COVID-19 spread and the implementation of different infection-control measures and testing approaches. We also forecast the optimal combination of individual-level measures and community surveillance to contain the new wave of COVID-19 spread after the re-opening work and social activities. In conclusion, our model introduces important novelties in the modeling strategies used to investigate the COVID-19 outbreak, and may be used to support government decision-makers.

8.3.1 The COVID-19 disease and how can be modeled

Italy was the first European country affected by the coronavirus 2 (SARS-CoV-2) outbreak, with the first autochthonous case identified in Lombardy on February, 21st, 2020 [26]. During the following weeks the number of people who tested positive for SARS-CoV-2 swab rapidly increased, exceeding 100,000 cases by the end of March 2020 [43, 147].

Undetected infections, being generally characterized by mild or no symptoms, can expose a large portion of the population to the virus and play a relevant role in the SARS-COV2 transmission. To reduce the spread of COVID-19, the Italian government introduced different restrictions, starting in the northern regions, where the first cases were detected, and then in the entire country. The first line of control was addressed to the closure of schools and museums. Later, people were encouraged to start smart working, and all the sports events were performed behind closed doors (February, 25th). The second intervention was focused on the closure of all the public activities involving crowd of people, restaurants and commercial activities; moreover, it was forbidden to cross the municipal borders (March, 8th). Finally, the latest control strategy imposed the total lockdown of the country halting non-essential production, industries and businesses (March, 21st). In the weeks following the third restriction, a slow but constant decrease of the infected cases was registered showing that the adopted control strategies had been effective in limiting the outbreak progression.
Starting from May 4th, these restrictions were gradually relaxed by the Italian government. In particular, work activities as manufacturing and wholesale were re-activated, and outdoor activities and the movements within each region boundaries were permitted. A complete reactivation of all the work activities was planned for the first week of June, while the school re-opening was postponed to September. At the same time, the government has required the intensification of infection-control measures (i.e., mask, gloves, social distancing), including specific rules to be adopted in workplaces, public places and transportation. The potential of tracing the cases' contacts and testing was also increased. In these contexts, computational models can be very helpful for evaluating COVID-19 epidemic evolution, and the effects of different infection-control strategies such as human interaction controls, and other social measures that can impact on disease spreading dynamics.

Several models, often with conflicting results [73], have been proposed to

investigate the COVID-19 pandemic. Models can be roughly classified as phenomenological and mechanistic. The former [131, 105] are formulated with the main aim of describing the epidemic pattern and make short-term predictions. The latter, such as the one we are proposing here, model the disease spread under various assumptions about the transmission process and the human and social contexts of the epidemic. These are used to obtain long-term forecasts and, possibly more important, to simulate different scenarios modulating the parameters that characterize variations in disease features and control measures. The model proposed by Ferguson and colleagues [53] had a strong impact in shaping the policies of several European countries and the US. In another influential model, Kissler and colleagues [80] explored the dynamics of COVID-19 over a period of several years, raising the possibility that repeated lockdowns may be necessary to keep levels of COVID-19 hospitalisations and deaths to manageable levels. Specific models have been proposed to describe the Italian epidemic development. The modeling study proposed by the Imperial College team [163] evaluates different scenarios for a relaxation of isolation measures, using increase in mobility as a proxy, and attempts to predict the *second wave* in terms of infection and death excesses. Another model, proposed by Giordano and colleagues [60] takes into account the distinction between diagnosed and non-diagnosed cases and points to the necessity of combining social-distancing measures with widespread testing and contact tracing to control the epidemic. A similar result was proposed in [89], in which the authors showed the key contributions of symptomatic and asymptomatic infections to onward transmission, considering the municipality of Vo', in the Veneto region. In particular they showed that viral transmission could be effectively and rapidly suppressed by combining the early isolation of infected people with community lockdown. Another interesting and differently work is proposed in [65], in which the authors proposed to exploit the Markovian Agents [17], a modeling technique with spatial features, to study the evolution of COVID-19 in Italy. This approach is based on agents that stochastically evolve among states coherently with the SIR logic, by taking in account both the spatial location of the population and, in particular, the impact of the infection among the Italian regions.

In this case study, we proposed an extended version of the SEIRS model to investigate COVID-19 spread disease. In particular, the novelties and

strengths of this model can be summarized as follows: (i) the division of the infected subjects in three categories: undetected, quarantined, and hospitalized; (ii) the explicit representation of the population age structure; (iii) the usage of age-specific and location-specific contact matrices; (iv) the modeling of the government actions and the corresponding population response depending on the public perception of the disease severity; (v) the modeling of different infection-control measures, including individual-level measures, whose efficacy is subjected to the public perception of current disease severity, and SARS-CoV-2 swab testing capability. Our model is then used to investigate different scenarios of COVID-19 diffusion in the Piedmont region by taking into account for the next three months following the gradual re-opening of May 4^{th}. In particular, in the next sections we studied how the COVID-19 spread in Piedmont could be kept under control by the implementation of the infection-control measures based on the use of individual-level measures (i.e., mask, gloves and social distancing), and on the intensification of the surveillance methods including contact tracing, the identification of undetected cases by swab testing, and early isolation of infected individuals.

8.3.2 The ESSN model

We propose an extended version of the SEIRS model to account for the population age distribution, that was classified into three groups: young individuals 0-19 years, adults 20-69 years, old adults aged at least 70 years. The corresponding transmission flow diagram for a specific age class i is shown in Fig. 8.33A, where the circles represent population partitions and the arcs describe the disease progression.

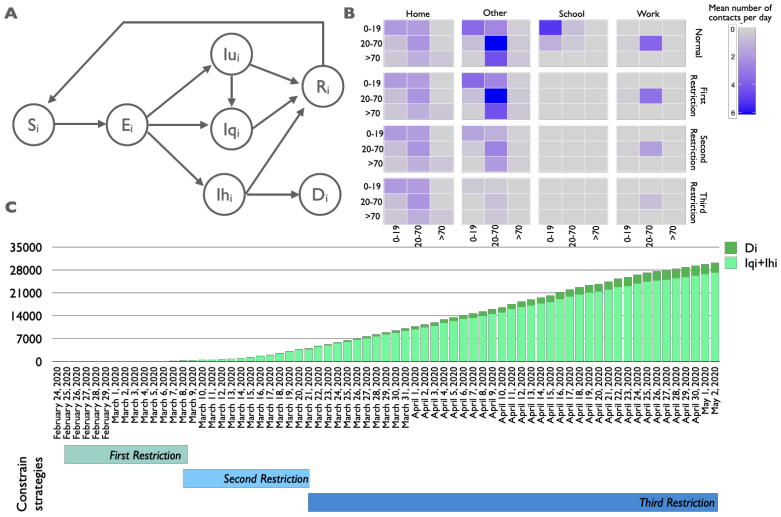


Figure 8.33: SEIRS model and surveillance data on Piedmont region. (A) The transmission flow diagram of our age-dependent SEIRS model. (B) Age-specific and location-specific contact matrices. The intense of the color indicates higher propensity of making contact. (C) Cumulative number of infected cases as sum of quarantined (I_{qi}) and hospitalized (I_{hi}) infected (light green) and deaths (D_i) (dark green) from February 24th to May 2nd. The periods of the activation of the three control strategies are reported below the stacked bars plot.

The population of the age class i is partitioned in the following seven compartments: *susceptible* (S_i), *exposed* (E_i), *undetected infected*(I_{ui}), *quarantined infected* (I_{qi}), *hospitalized infected* (I_{hi}), *recovered* (R_i), *dead* (D_i). The partition of the infectious population allows to model quarantine practices and the effects of government control strategies specifically for each sub-class of individuals. Similarly, the division of population in three-age groups allows to define age-dependent rates for the system events (e.g., infection, recovery, ...). We can thus model a scenario in which younger individuals, known to be more often asymptomatic or pauci-symptomatic, are more likely undetected than the older ones.

With respect to the classical SEIRS model, we have added a transition from I_{ui} to I_{qi} to model the possibility to identify undetected cases and isolate them. In this way an individual in I_{ui} tested as positive to the SARS-CoV-2 swab will be moved in the quarantine regime, I_{qi}. This feature is crucial to capture the time varying diagnostic ability throughout the epidemic evolution, as shown by the increasing number of tests performed [124], as well as

to forecast the effect of enhanced or decreased testing capabilities.
The social mixing pattern in the population is described by the age-specific and location-specific contacts depicted in the matrices reported in Fig. 8.33B. Social contacts change across contexts (i.e. home, work, school and other locations) and age-groups [123]. The effect of the public restrictions imposed by the Italian government was simulated by reducing social mixing contacts in all categories.
The *force of infection (FOI)* adopted in the model is a time and age class dependent function and includes the following four terms:

- the *infection rate*, depending on the age classes of both the susceptible and the infected individuals who come into contact according to the contact matrix;
- the *strength of governmental restriction* defined through a time-depended step function, modeling the severity of the public restrictions;
- the *compliance with the governmental restriction*, reporting how effectively the population adheres to the restriction measures imposed by the Italian government. The higher the disease severity (i.e., the severity of the epidemic in terms of number of deaths and hospitalized individuals in the last 40 days), the better the population compliance [91];
- the *compliance with individual-level measures*, considering how different infection-control measures are properly adopted by the population.

All demographic changes in the population (i.e., births, deaths, and ageing) are explicitly disregarded in our model as negligible due to the short time interval considered in our study.

Let us now introduce the mathematical details and the ESSN model depicted in Fig. 8.34. In the ESSN model we modeled the seven compartments introduced above with five modules characterized by 46 places and 66 transitions. The **Susceptible module** is characterized by the place s representing the susceptible individuals, and it models individuals not exposed to the pathogen and may be infected if get in touch with an infected patient, through the *Infection* transition, becoming an exposed patient. The exposed patient, modeled by the e place, does not show symptoms until 5 to 15 days are elapsed. Such phenomena is known as incubation period, and it is modelled through

the **Exposed module**. At the end of the incubation period, exposed subjects start showing infection symptoms becoming infectious, which are modeled by the i place in the **Infected module**. In particular, three transitions may occur: L_u, L_q and L_h, to distinguish if the infected patient will be undetected, quarantined or hospitalized, respectively. Indeed, in the first case when the infected patient is undetected, the patient keeps on with its social interactions and the contact matrix is not reduced. Differently, when an individual is detected, i.e. the quarantine or hospitalization occur, its interaction with other peoples are strongly reduced. Furthermore, the *Detection* transition which moves the undetected infected in quarantine modeling the surveillance methods including contact tracing, the identification of undetected cases by swab testing, and early isolation of infected individuals.
After a certain time period the infected patients may:

1. recover becoming a recovered patient, which is not contagious anymore. In this case we are moving from the i place in Infected module to the r place in **Recovered module**. Let us note that until the recovered patient returns susceptible through the *ReturnS* transition, he/she can not be infected again remaining out of the disease dynamics;

2. die moving from the i place in Infected module to the d place in **Death module**. Let us note that we are considering the patient death may occur only for the hospitalized infected considering the last two age classes. In particular, the death rate of the *Old* class is greater than the *Adult* one, given the strong age gradient in risk of death [159].

The remaining places, denominated c_*, need to count how many times the respectively input transition fires. Thus, the place c_Lu represents the cumulative number of the undetected cases.

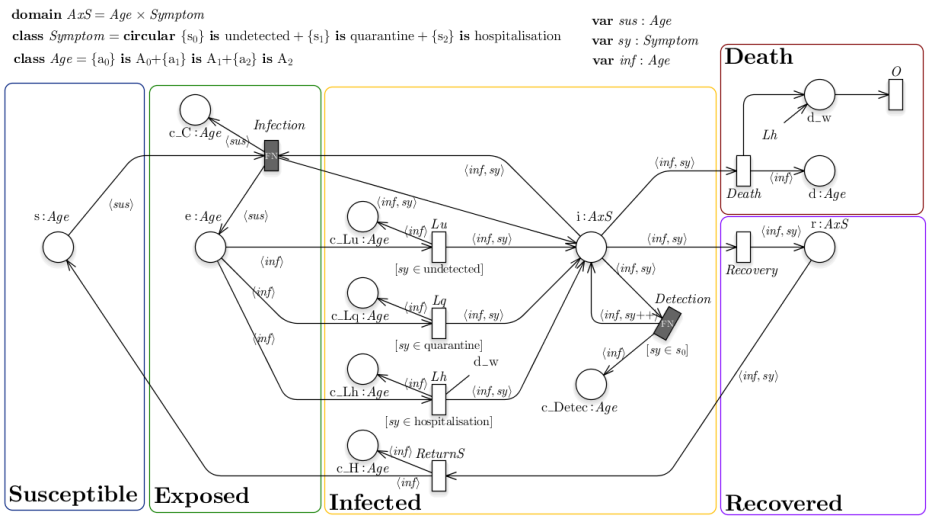


Figure 8.34: The ESSN model regarding the COVID-19 spreading.

The color classes in the model are: *Age* and *Symptom*. The former is subdivided in three sub classes A_0, A_1, and A_2, representing the population age structure which is defined by three age sub-classes $\mathcal{A} = \{a_i |\ a_1 \in (0 \sim 19],\ a_2 \in [20 \sim 69],\ \ a_3 \in 70^+\}$. Within each age class, individuals are homogeneous both for their usual social activities and for the susceptibility to the disease. The latter color class represents the category of infected which a susceptible may become after the infection, and it is defined by three sub color classes: *undetected*, *quarantine* and *hospitalized*.
Furthermore, this layered population enables the model to track age specific incidence of the disease and the age specific social activity patterns. Specifically, we consider four different contact types, as reported in [122]: within the family unit ($Home$), at school ($School$), at work ($Work$) and other contacts ($Other$). Clearly, direct contacts are the main driver in the diffusion of the pathogen, while the quarantine regimes (η), the actions undertaken by the policy-makers (α) and the adoption of Personal Protective Equipment (p) are targeted to reduce them.

We report here the compartmental model formulation:

$$\begin{cases} \frac{dS_i}{dt} &= -\sum_{j=1}^{|\mathcal{A}|} \iota_i S_i \frac{(B(i,j,t,u)I_{uj} + B(i,j,t,q)I_{qj} + B(i,j,t,h)I_{hj})}{N_j} + \nu R_i \\ \frac{dE_i}{dt} &= \sum_{j=1}^{|\mathcal{A}|} \iota_i S_i \frac{(B(i,j,t,u)I_{uj} + B(i,j,t,q)I_{qj} + B(i,j,t,h)I_{hj})}{N_j} - (\lambda_u + \lambda_q + \lambda_h) E_i \\ \frac{dI_{ui}}{dt} &= \lambda_u E_i - \rho I_{ui} - \theta(i,t) I_{ui} \\ \frac{dI_{qi}}{dt} &= \lambda_q E_i - \rho I_{qi} + \theta(i,t) I_{ui} \\ \frac{dI_{hi}}{dt} &= \lambda_h E_i - (\sigma_i + \rho) I_{hi} \\ \frac{dR_i}{dt} &= \rho I_{ui} + \rho I_{qi} + \rho I_{hi} \\ \frac{dD_i}{dt} &= \sigma_i I_{hi} \\ \frac{dP}{dt} &= \sum_{i=1}^{|\mathcal{A}|} (\sigma_i I_{hi} + \lambda_h E_i) - \mu P \\ N_i &= S_i + E_i + I_{ui} + I_{qi} + I_{hi} + R_i, \end{cases} \tag{8.1}$$

where $B(i,j,t,s)$ represents the FOI and it is defined as a time t, age class i,j, and infected categories s dependent function

$$B(i,j,t,s) = \omega(t)\ p(t)\ \sum_{c \in \mathbf{C}} \alpha(t,c)\ \eta_{s,c}\ \beta^c_{i,j}, \qquad i,j \in \mathcal{A} \wedge s \in \mathcal{S}. \tag{8.2}$$

In details,

- $\mathcal{A}$, $\mathcal{S}$ represent the set of age classes and infected categories, respectively;
- $\alpha(t,c)$ is the strength of governmental action, defined as a step function depending on the time period;
- $\eta_{s,c}$ is the amount of social interaction considering the contact category $c \in \mathbf{C}$ of the infectious sub-class $s \in \mathcal{S}$;
- $\beta^c_{i,j}$ is a constant **contact rate**, depending on (1) the age classes of both the susceptible (i) and the infected (j) individuals, and (2) the contact categories [122];
- $p(t)$ describes the comply the Personal Protective Equipment (PPE) and social distancing measures imposed by the Italian government;

- $\omega(t)$ represents the intensity of the population response given the disease severity (k) with respect to the number of deaths and hospitalized infected individuals by CODIV-19 (P) in the last 40 days, assuming that the higher is P, the more prone to reduce the contacts is the population [91]. This is defined as follow:

$$\omega\left(t\right) = \left(1 - \frac{P(t)}{\sum_{i=1}^{|\mathcal{A}|} N_i}\right)^k . \tag{8.3}$$

Furthermore, Table A.10 in the Appendix provides a detailed description of the parameters used in the compartmental model and further provides the values of those known parameters together with the reference to the original contribution.

8.3.3 Model calibration

The surveillance Piedmont data available at the website of the Italian Ministry of Health / Civil Protection [124] were used to calibrate our model. Among the data, the surveillance report publishes three categories of monitored individuals: quarantined infected (I_{qi}), hospitalized infected (I_{hi}) and deceased (D_i), whose cumulative trends are reported in Fig. 8.33C. In the same Figure, the time points at which the control strategies were imposed are also shown. The calibration phase was performed to fit the deterministic model outcome with the infection ($I_{qi} + I_{hi}$) and death (D_i) data from February 24$^{\text{th}}$ to May 2$^{\text{nd}}$ using squared error estimator via trajectory matching.
From the ODEs system in Eq. 8.1, 13 parameters are characterized by a high uncertainty, due to the recent onset of the disease and their difficulty of being empirically measured. In details, three parameters represent the probability of infection for each age class, four parameters reflect the governmental action strength at time epoch t (i.e., $\alpha(t)$ with $t \in \{$ March 8$^{\text{th}}$, March 21$^{\text{st}}\}$), one parameter describes the intensity of the population response (i.e., k), two parameters represent the death rate for the hospitalized patients (i.e., $\sigma_i,\ i = 2,\ 3$, fixing $\sigma_1 = 0$), two parameters are the initial condition for the undetected and quarantine infected individuals, and the remainder parameter represents the detection rate for the third age class starting from the 1^{st} April. Due the great number of these parameters, in addition to well reproduce the available data, their estimation is further controlled by the reproduction number R_0, calculated by exploiting the Next Generation Matrix method [157].

Nonetheless, the available information in [124] only provides an aggregated overview of the number of quarantine infected individuals with mild symptoms, the number of hospitalized infected individuals with moderate or severe symptom and the number of deaths, without taking into account any age subdivision. Therefore, in order to provide a realistic distribution of cases in the population classes defined as in $\mathcal{A}$, the number of real cases has been distributed according to the disease incidence reported in Table 8.8 for the Piedmont region. Furthermore, the model calibration was carried out considering the proportion between undetected and detected infected individuals (i.e., given by the sum of the quarantined and hospitalized infected individuals) to be one-to-one on average as reported in [108]. Neglecting the fraction of undetected, the remaining portion of the cases is divided into Ih and Iq as reported in [124]. The number of hospitalised patients, which is in average on the time interval from 24^{th} February to the 4^{th} May the 45% of the total reported cases, is considered for simplicity equal for each age class.
According to [168], we assumed that the initial system state is $S_1 = 733130$, $S_2 = 2780600$, $S_3 = 842676$,; while the initial number of infected individuals is estimated as $Iq_2 = 4$, $Iu_1 + Iu_2 + Iu_3 = 100$, where the undetected individuals were distributed proportionally to the age class population size. All the other compartments are set to zero.

Table 8.8: Incidence of COVID-19 infections and deaths in Piedmont

		Age classes			Reference
	Updated at	$0-19$	$20-69$	$70++$	
percentage of infection	2020-03-17	1%	60.1%	38.9%	[46]
	2020-03-22	0.8%	61.7%	37.5%	[47]
	2020-04-16	0.89%	55.1%	44.01%	[44]
	2020-04-26	1.3%	53.55%	45.15%	[45]
percentage of death	2020-04-15	0%	16.5%	83.5%	[151]

Finally, the estimated parameter values are reported in Table 8.9, and the basic reproduction number resulting from such parameters is $R_0 = 2.7$ considering the initial stage of the pandemic in Piedmont the period from February 21^{st} to February 25^{st}.

As since April 1^{st} a tangible increment of the SARS-CoV-2 swab tests in long-stay residential care homes was implemented in Piedmont [153], we

Table 8.9: Estimated parameters of the best fitting trajectory, where ι_i represents the age class (i) dependent probability of infection given a contact, σ_i the death rate for patients with sever symptoms in age class i, $\alpha(t, c)$ the governmental action strength at time t considering the contact c, and k the intensity of the population response.

Parameter	Value			
	i =	1	2	3
ι_i		0.0095	0.08	0.285
σ_i		-	0.019	0.33
$\alpha(t, c)$	c = / t =	$Work$	$Other$	
	March 8$^{\text{th}}$	0.75	0.65	
	March 21$^{\text{st}}$	0.4	0.3	
k		60		
Iq_2		4		
$Iu_1 + Iu_2 + Iu_3$		100		
$\theta\,(3, \text{April } 1^{\text{st}})$		0.12		

explicitly modeled the diagnostic ability to identify undetected cases among the old adults (70+) starting from the beginning of April.

The plots in Fig. 8.35A and 8.35B show the time evolution of infected and deceased individuals derived by the model considering the optimal parameter values estimated by the calibration phase (Table A.12 in the Appendix). In details, the stacked bar chart in Fig. 8.35A shows the proportion of infected individuals in each of the three infectious sub-classes considered. The purple line represents the simulated number of cases diagnosed in the elderlies, that would have been undetected, had the testing procedures not changed since April 1$^{\text{st}}$, while the red line reports the detected infected individuals derived from the surveillance data.

Fig. 8.35A reveals a good level of accordance between the infected individuals derived by the surveillance data and those derived by the calibrated model (i.e., given by the sum of the I_{qi} reported with light blue bars and I_{hi} reported by blue bars). Consistently, Fig. 8.35B shows that the calibrated model is able to mimic consistently the observed death cases (red line). The plots reporting infected and deceased individuals for each age class in the same time interval are shown in Fig.s A.4 and A.5 in the Appendix.

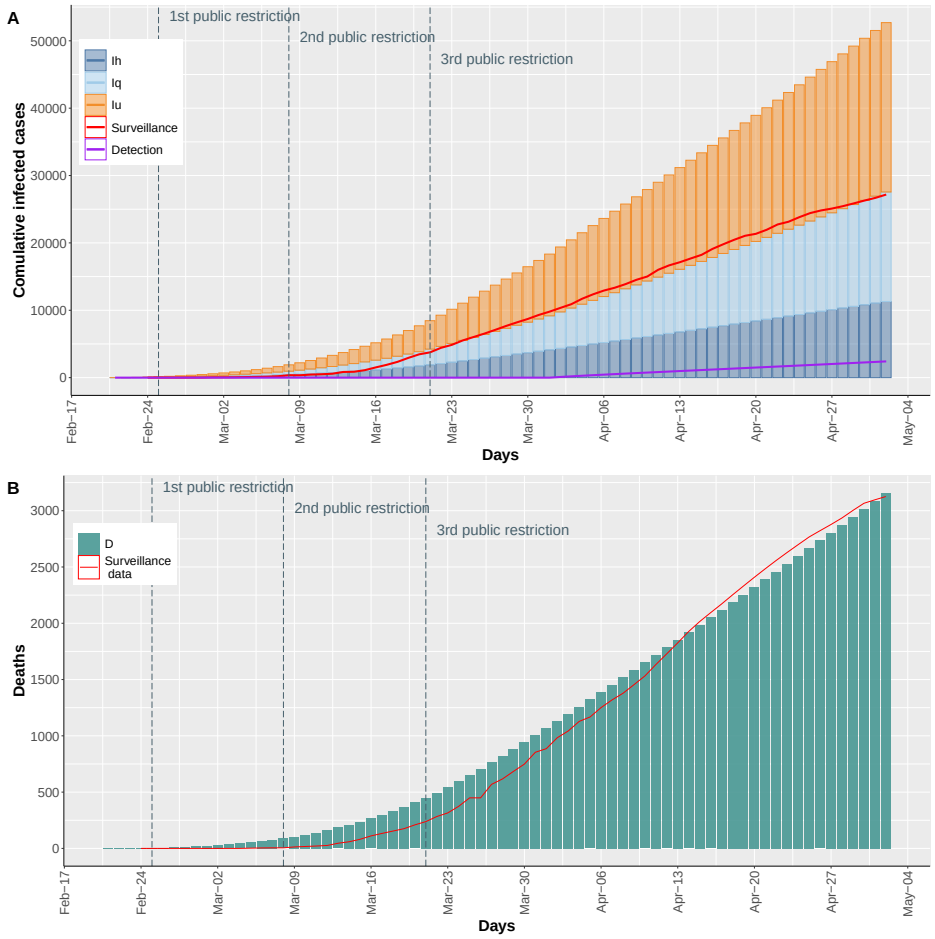


Figure 8.35: (A) Stacked bars plot reports the cumulative trend of the infected individuals in which the undetected infected are showed in orange, the quarantine infected in light blue, and hospitalized infected in blue. The purple line reports the cumulative trend of the undetected cases diagnosed by the simulated SARS-CoV-2 swab tests. (B) Histogram shows the cumulative trend of deaths. In both histograms the surveillance data are reported as red line.

8.3.4 Model analysis

The COVID-19 spread and the government control interventions.

To study how the government control interventions and the corresponding population response affected COVID-19 diffusion, we focused on the third restriction, when a strict lockdown was enforced in Italy between from March 21st to May 1st. In particular, we used our model to compare the infection

spread under the following three scenarios: (i) the third restriction is activated from March 21st and the population response is estimated by the surveillance data; (ii) the model extends the second restriction beyond March, 21st without implementing the third restriction and the population response is the one estimated from surveillance data; and (iii) the third restriction is activated from March 21st and the population response is higher than the one estimated by the surveillance data.

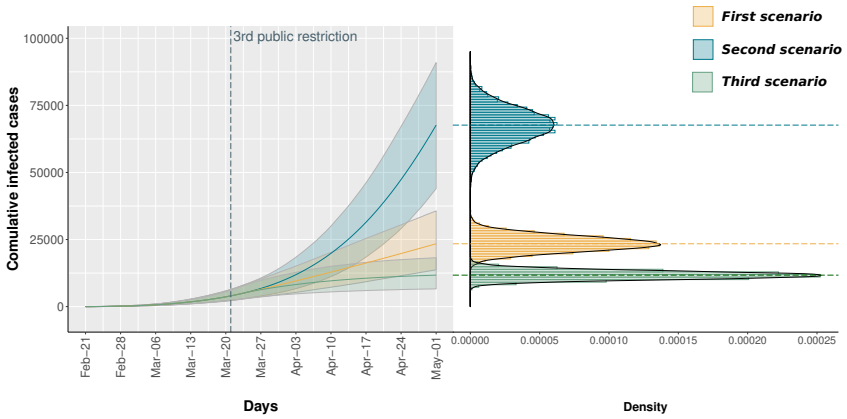


Figure 8.36: Stochastic simulation results reported as traces (on the left) and as density distributions (on the right). Three scenarios are implemented. In the *First scenario* the model is calibrated to fit the surveillance data (yellow). In the *Second scenario* the model extends the second restriction beyond March, 21st without implementing the third restriction (blue). In the *Third scenario* the model consider a higher population compliance to the third governmental restriction (green).

Figure 8.36 shows the stochastic simulation traces (on the left) and the density distributions on May 1th (on the right) of the total number of detected infected individuals, considering the three scenarios proposed above: yellow, blue and green for the first, second and third scenario, respectively. For each scenario 10'000 traces are simulated and the corresponding median trace is reported as a bold line. It is possible to appreciate that the third restriction was effective in containing the spread of the virus. In particular, the distribution under the first scenario, representing the observed data, is much closer to the third scenario, in which an almost complete compliance with the restriction is simulated, than to the second scenario assuming no lockdown

and yielding a dramatically higher number of cases.

COVID-19 epidemic containment strategies.

From May, 4th the restrictions imposed by the Italian government were gradually relaxed: the roadmap for lifting COVID-19 restrictions defined by the Italian Government sets out three reopening phases that are depicted in Figure 8.37. In this context, our model can be effectively used to forecast the daily trend of the infected individuals until September 1st considering this progressive increment of the social mixing patterns and implementing different infection-control measures.

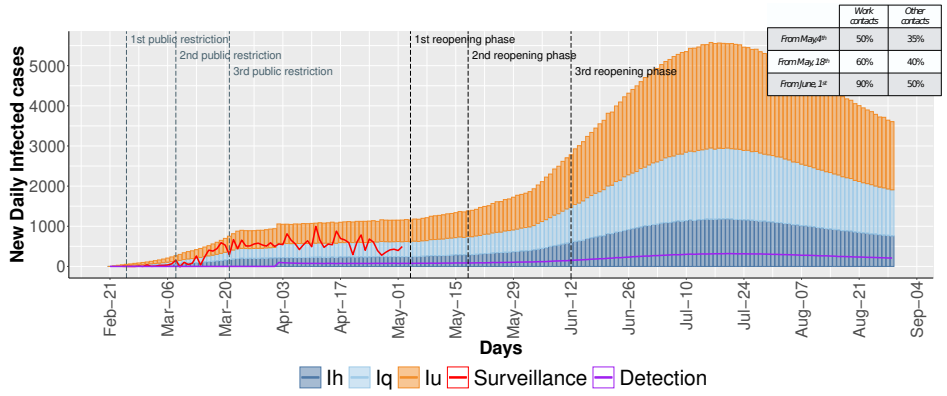


Figure 8.37: The daily evolution of infected individuals computed by the stochastic simulation. The stacked bars report the undetected infected (orange), the quarantine infected (light blue), and hospitalized infected (blue). The red line shows the trend of the infected cases from surveillance data. The purple line is the cumulative trend of the undetected cases diagnosed by the modeled SARS-CoV-2 swab tests.

Figure 8.37 shows a pessimistic scenario in which the gradual reopening is not counterbalanced by any infection-control strategies. The stacked bars report the predicted infected cases (blue and light blue) and the number of undetected individuals (orange), whereas the red line shows the surveillance data until May 1st. Moreover, the purple line reports the daily trend of otherwise undetected cases diagnosed by SARS-CoV-2 swab tests by the model.

After a first constant increment of the infected individuals in February and March, a plateau was reached from April, 3rd to May, 1st. From that moment, the gradual re-opening of the working activities would cause a new increment of infected individuals, reaching a peak of about 7,000 daily new infected cases

on July, 20^{th} when a gradual decrease would be produced by the population response to the severity of the epidemic. Starting from this worst-case scenario, we analyzed the cost-benefit trade-off between the implementation of infection-control measures and the relaxation of public restrictions. In particular, we consider 15 different scenarios arising from the combination of different levels of implementation/efficacy of two control measures: (i) use of individual-level measures, and (ii) increased case detection by contact tracing, swab testing and early quarantine regime of identified cases.

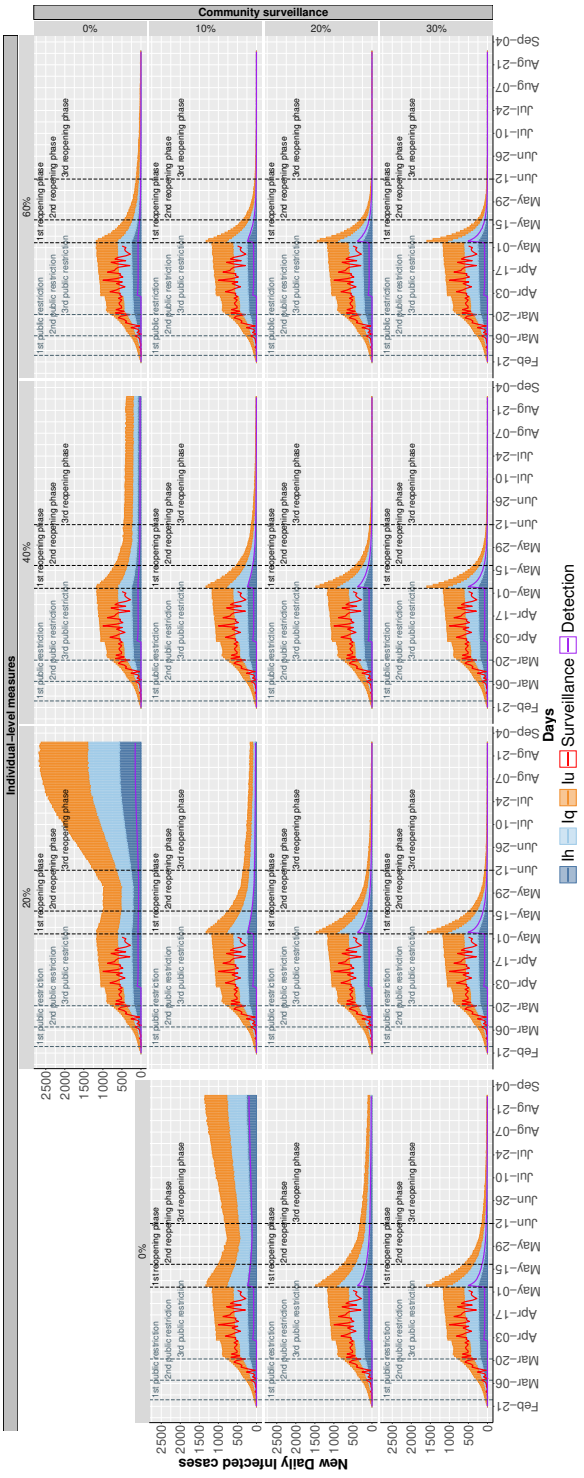


Figure 8.38: The daily evolution (median of stochastic simulations) of infected individuals is shown varying on the columns the the efficacy of individual-level measures and on the rows the efficacy of community surveillance. The sudden increment associated with May, 1st is due to the increasing of the surveillance from that time.

Specifically, in Figure 8.38 we show the daily forecasts of the number of infected individuals with the efficacy of individual-level measures ranging from 0% to 60% on the columns (increasing by steps of 20%) and, on the rows, increasing capability (from 0% to 30%, by 10% steps) of identifying otherwise undetected infected individuals. These results are obtained as median value of 5000 traces for each scenario obtained from the stochastic simulation.

Comparing results in Figure 8.38 with the pessimistic scenario of Figure 8.37 it is important to notice that, regardless the efficacy combination we pick, the employment of infection-control strategy always exhibits a positive effect on the number of infected individuals, either flattening the peak or the number of infected individuals toward zero. Furthermore, the proposed set of scenarios shows that the combination of the two infection-control measures leads to envisage reasonable levels of protection arising from the adoption of individual-level measures (in the range of 20-40%) and the necessity to identify a feasible fraction of undetected cases. It is important to notice that the fraction of revealed undetected cases is not the reference measure for the number of people to test for Sars-CoV-2 infection. This rather represents the fraction of all undetected patient that, thanks to an enhanced testing approach, are eventually tested and identified. To reach this goal, the actual number of swabs to perform depends on the positive predictive value of the test, which, in turns, depends on the prevalence of SARS-CoV-2 positive individuals in the tested population. Thus, the same result can be achieved with a thorough contact tracing and targeted testing in high risk groups or with a larger number of untargeted (or less targeted) tests. The former approach is more efficient and feasible.

8.3.5 Discussion

In this study, we exploited our modeling framework to analyse a multi-group SEIRS model, considering an age-structured population. Differentiating age classes allows us to mimic the real incidence of COVID-19 both for infections and fatalities. We also introduced a further layer to characterize categories of infected, so that also the severity of the infection and its associated quarantine regime can be modeled. The multi-group nature of our model allows us to investigate the peculiarities of the COVID-19 transmission, including different incidences or symptoms severity in different age classes. In our model, the time evolution of the infection is affected by governmental policies and the

associated population awareness. Given the complex dynamics driving the behaviour of the population, we further modeled the perception of the hazard of COVID-19, relating it with the number of fatalities.
The high level of parametrization and the flexibility provided by the framework gives the possibility to study different scenarios to simulate 1) the impact of the control strategies to limit contacts in the population, and 2) the containment strategies based on individual-level measures. Firstly, we focused on the impact of the control strategies by modulating the parameters that modify the strength of the governmental restriction and the population compliance with these restriction. Intervention scenarios that intensify the limitation of the person-to-person interactions correspond to an increase in these terms. We investigated the impact of the third public government restrictions on the evolution of the COVID-19 epidemic in Piedmont and found that the strict national lockdown and the related population response had a strong impact on the epidemic control. Model outcomes clearly highlight that the implementation of the sole second restriction would have not been enough to counteract the outbreak. Secondly, we pinpointed the optimal combination of containment strategies based on individual-level measures and community surveillance to cope with the COVID-19 spread. In particular, considering the roadmap for lifting COVID-19 restrictions defined by the Italian Government, the model shows that if none of the infection-control measures is applied, the number of infected cases is bound to increase, leading to a second wave of infections. However, this can be substantially contained when infection-control measures are implemented. In particular, the model results highlight that the combinations of individual-level measures with undetected infection diagnosis can be effective to controlling the virus spread even when their singular implementation does not reach a high level of efficacy (e.g. 40% and 20%, respectively).

Chapter 9

Conclusion and future work

In this thesis three original contributions were presented, concerning a) the definition of new high level formalisms based on the PN, and b) the development of general and efficient analysis techniques for modeling biological complex systems. These theoretical results are then gathered into a new general modeling framework, which can be exploited even by researchers without advanced mathematical and computational skills.
In details, the first contribution regards the extension of PN formalism to provide a more general tool able to simulate the general dynamics governing the events of a system. The two new formalisms, namely Extended Stochastic Petri Net (ESPN) and Extended Sthocastic Symmeric Net (ESSN), obtained by extending the Sthocastic Petri Net and Sthocastic Symmeric Net respectively, are introduced in Chapter 5. We showed the advantages of these extended formalisms considering three complex case studies: the Pertussis disease, the Multiple Sclerosis disease, and the COVID-19 epidemic. In particular, thanks to the high level of parametrization and flexibility provided by these graphical formalisms it is possible to model more complex dynamics with respect to the ones defined by the MA law, but also to combine different networks (e.g., signaling, metabolic, and regulatory networks) in a unique model, as we showed in Section 6.2. Considering these formalisms, as future works, we will investigate 1) how spatial constraints could efficiently be introduced,

for instance, considering partial differential equations instead of the ODEs, and 2) how to use the power of the ESSN to model systems with higher level of details typical of the agent based model [143].
The second contribution consists of three new computational techniques developed to make more efficiently the analysis of complex and general models, described through ESPN and ESSN formalisms. The first technique is presented in Section 6.1, and it is a new method to automatically derive the reduced ODEs system from the ESSN model exploiting the systems symmetries, by avoiding the complete net unfolding. In particular, we started defining this new method considering the SSN formalims, and then we discussed which constrains have to be defined on the functions characterizing the general transitions in the ESSN. The second technique is described in Section 6.2, in which we showed that the ESPN formalism can be exploited as a "*metaformalims*" to create hybrid model characterized by some sub-components detailed as a mechanism model and others as a constraint model. This is motivated by the needs of an increasingly importance of integrating into unique model data coming from different sources. Thus, in this thesis we provided some indications on how this integration can be carried out. In particular, we showed how the FBA approach can be used whenever the knowledge of the rate coefficients of every reaction of a net component is missing. Of course, this assumes that all the components solved through FBA reach their steady state faster than all the other components. Obviously, when this assumption is not satisfied, then an approximation of model behavior is introduced. As future work in this area, we will apply this new approach for modeling a more complex and real case study, such as the Colon cancer disease by combining the bacteria behaviour (modeled through the FBA model) and the Cancer growth (modeled through the ODEs system).
Finally, in Section 6.3 we presented the third technique, that is a generalisation of the τ-*leaping algorithm* to deal with generic dynamics as those allowed by ESPN and ESSN formalisms. Indeed, we highlighted that, differently from the SSA, the $\tau-$leap approach depends on the transition velocity law and its partial derivatives, which are needed to correctly derive τ. Therefore, if the firing intensities are defined as continuous functions, then it is no longer possible to know in advance the propensity functions and so their partial derivatives. To overcome this problem we proposed to exploit a method of approximation of derivatives, called **Richardson's Extrapolation**, showing good results by

comparing the simulations obtained from the classical $\tau-$leaping approach and our extended version.

The last contribution concerns the development of *GreatMod* (Chapter 7), a general modeling framework for the analysis of biological and epidemiological systems, in which all the theoretical results proposed in this thesis are implemented. The novelties and strengths of the proposed framework can be summarized as follows: (1) the use of graphical formalisms (i.e., ESPN and ESSN) for the model creation; (2) a user-friendly interface based on R language; (3) framework portability and reproducibility of the results; (4) the possibility to integrate user-defined workflows. The implemented functionalities can be grouped into four classes: (i) model generation implementing the generation of the stochastic and deterministic process underlying the graphical model; (ii) sensitivity analysis implementing Partial Rank Correlation Coefficient analysis; (iii) model calibration adjusting the model parameters to obtain a best fit between the model outcomes and real data; (iv) model analysis solving the model and generating an output representing the time evolution of the model. As future works related to this framework, we are evaluating the possibility to introduce new calibration algorithms, the multi-objective optimization paradigm and parallel evolutionary algorithms will be taken into account, in particular with the possibility to port some of the optimizations algorithms on GPU architecture to further speed up the model calibration step.

The effectiveness of this framework and the theoretical results is shown through three different case studies in which we investigated i) the Pertussis epidemiology in Italy and its vaccination policy (Section 8.1), ii) the Multiple Sclerosis disease and its treatments (Section 8.2), and iii) the COVID-19 epidemic spread in Piedmont (Section 8.3). In particular, by using the *GreatMod* functions, in all these case studies we are able 1) to easily create the model and its underlying mathematical processes, 2) to analyse the model parameters, and 3) to simulate different scenarios.

For all these reasons we believe that this framework represents a substantial advance in the field of computational epidemiology and system biology, and will be beneficial for the entire community. For instance, we are already applying *GreatMod* to study the spreading of the West Nile Virus (WNV) disease in Piedmont, one of the most recent emerging mosquito-borne diseases in Europe and North America.

Appendix A

Appendix

A.1 $\mathcal{G}reat\mathcal{M}od$ installation

The installation of the workflow requires the downloading of

1. the extended version of the GreatSPN editor at `http://www.di.unito.it/~amparore/mc4cslta/editor.html`,
2. Docker `https://docs.docker.com/engine/installation/`,
3. the R library at `https://github.com/qBioTurin/epimod`.

In particular, a step by step guide to use and install the framework is reported at `https://qbioturin.github.io/epimod/`.
Finally, after installing these three requirements, the R function ***download_images()*** prepares the docker environment downloading the docker images needed by the framework.

A.2 Supplementary information: Pertussis model

A.2.1 Parameters

All the parameters file reporting their values can be found at the following link `https://github.com/qBioTurin/epimod/`.

Contact rates. We are considering the contact matrix provided by [107], in which the Italian contact rates depending on the age are reported.

	a_1	a_2	a_3
a_1	0.2136752137	0.5586592179	0.1438848921
a_2	0.5586592179	0.0205212395	0.0364033491
a_3	0.1438848921	0.0364033491	0.0063742988

Table A.1: Contact rates between patients divided by the class they belong to, obtained from [107]

.

Initial marking. The initial marking is a vector defined by 179 variables, that are all the places of our ESSN associated with all the corresponding color

classes combinations (given by the color domain **A**, **V**, and **L**). Since the simulations start from the 1974, from when we are considering no vaccination, all the places with colors different to *NV* (i.e., no vaccination) are settled to zero. For obvious reasons, all the places representing counters (birth, vaccination, and infection counting) are settled to zero. From [62] and the surveillance data, we were able to estimate the number of infects in each age class in the 1974. In details, during the whole year there were reported 7'400 cases distributed as follow: 15% in N, 80% in Y, and 5% in O. Supposing for simplicity: (1) to split as equals the number of infects between the I_p and I_s places, and (2) to obtain the mean initial number of infects to scale their values with a factor of 21/365, since the mean length of the infectious period is 21 days. This is necessary because the time scale of our simulations is *days*.
Finally, the remaining places to estimate their initial markings are given by: the susceptible in each age class (*S_a1, S_a2, S_a3*), and the recovered without vaccination in each age and resistance level class (*R_a1_nv_l4, R_a2_nv_l2, R_a2_nv_l3, R_a2_nv_l4, R_a3_nv_l2, R_a3_nv_l3, R_a3_nv_l4*). For simplicity we consider *R_a1_nv_l1, R_a1_nv_l2, R_a1_nv_l3* equal to zero since the time necessary to decrease the immunity level is greater than 1 year.
Since it is known the size during the 1974 of the Italian populations and how they are distributed among the age classes, removing the infects, we have to estimate through the sensitivity and calibration analysis their distribution among the susceptible and recovered, and the resistance levels as well.

Initial condition	*Value*
Ip_a1	3.198904e+01
Ip_a2	1.748466e+02
Ip_a3	6.415068e+00
Is_a1_nv	3.198904e+01
Is_a2_nv	1.748466e+02
Is_a3_nv	6.415068e+00
S_a1 + R_a1_nv_l4	866703
S_a2 + R_a2_nv_l1+ R_a2_nv_l2 + R_a2_nv_l3 + R_a2_nv_l4	15685693
S_a3 + R_a3_nv_l1+ R_a3_nv_l2 + R_a3_nv_l3 + R_a3_nv_l4	37837299

Table A.2: Initial conditions.

Vaccination rate. The vaccination rate is defined through the vaccination policy and the properties characterizing the Exponential Negative Distribution. So to estimate the vaccination rate χ let us introduce three i.i.d. random variables :

1. death in the first year: $D \sim Exp(\delta_{a_1})$,
2. growth from the first age class to the second one: $G \sim Exp(\lambda_{a_1,a_2})$,
3. vaccination: $V \sim Exp(\chi)$, (equal for each vaccination dose),

and the events (i) $V_i = \{i^{th}\text{vaccination dose is submitted}\},\ i = 1, 2, 3$, and (ii) $A_1 = \{\text{belonging to the first age class}\}$. Therefore, requiring that the probability to complete the vaccination cycle in absence of disease during first year is equal to p, i.e.

$$\mathbb{P}\{(3vaccination\ doses\ are\ submitted)|(before\ the\ first\ year)\} = \mathbb{P}\{V_1 \cap V_2 \cap V_3 | A_1\} = p, \tag{A.1}$$

we are able to deduce that

$$\begin{aligned}\mathbb{P}\{V_1 \cap V_2 \cap V_3 | A_1\} &= \prod_{i=1}^{3} \mathbb{P}\{V_i | A_1\} \\ &= (\mathbb{P}\{V_1 | A_1\})^3 \\ &= (\mathbb{P}\{V = \min(D, G, V)\})^3 \qquad \text{(A.2)} \\ &= \left(\frac{\chi}{\chi + \delta_{a_1} + \lambda_{a_1,a_2}}\right)^3 = p. \qquad \text{(A.3)}\end{aligned}$$

Where we suppose that the random variables $D,\ G, V$ are i.i.d., the three events $V_i,\ i = 1, 2, 3$ as well, and that one vaccination dose is submitted iif the transition modeling the vaccination fires before the transitions modeling the death or the growth (and so the exit form the first age class). Then, the properties characterizing the exponential distributions [1] are exploited to obtain an analytic formula depending on the unknown parameter χ, and the known parameters $\delta_{a_1},\ \lambda_{a_1,a_2},\ p$. Indeed solving the following equation

$$\left(\frac{\chi}{\chi(\nu) + \delta_{a_1}(\nu) + \lambda_{a_1,a_2}}\right)^3 - p(\nu) = 0,$$

[1] Let $X_1, \ldots, X_n$ be independent random variables, with X_i having an $Exp(\lambda_i)$ distribution, respectively. Then the distribution of $\min(X_1, \ldots, X_n)$ is $Exp(\lambda_1 + \cdots + \lambda_n)$, and the probability that the minimum is X_i is $\frac{\lambda_i}{\lambda_1 + \cdots + \lambda_n}$.

we are able to estimate the vaccinate rate χ depending on the vaccine policy defined a priori. The probabilities p were calculated from the percentage of newborns who completed the vaccination cycle. These percentages were reported every year from 1994 to 2016. For this reason p is time dependent, and they were obtained from [62, 171].

Birth rate. The birth rate has been computed directly from data provided by ISTAT for the period 1974 $\sim$ 2016. In details, it is defined as the average births per day in the reference period with a dependence on the year.

Death rates. Similarly to the birth rate, the death rates are defined as the average deaths per day from 1974 to 2016, with a dependence on the age class and the year.

A.2.2 General transition functions

In this section we describe in details the firing rate function associated with the generic transitions in our Pertussis model. Let us recall that:

1. $f_{\langle t,c\rangle}(\hat{x}(\nu),\nu)$ is the speed of the transition $t \in T_g$ and $\hat{x}(\nu)$ represents the vector of the average number of tokens for all the input places of t. For brevity when the function does not depend on the color instance c then we omit it reporting only the transition t, i.e. $f_t(\hat{x}(\nu),\nu)$.

2. given a transition with a specific color domain, the unfolding procedure generates automatically the transition name combined with all the possible combinations of the color classes associated. For instance, the transition $ContactS\_IpToIp$ is unfolded into $ContactS\_IpToIp\_a_1\_a_1$, $ContactS\_IpToIp\_a_1\_a_2$, $ContactS\_IpToIp\_a_1\_a_3$, etc.

All the general transitions of the model are now explained in details and all the constants are summarized in Tables A.3.

Transitions modeling the contacts. All the transitions representing the contact between a person belonging to the age class a_i with one from a_j, which

Symbol	*Parameter*	*Value*
γ	Recovery rate	21 days
θ^{vacc}	Decay of the resistance derived from vaccination	7 years (21 months per level)
θ^{inf}	Decay of the resistance derived from infection	14 years (42 months per level)
X_0	Initial population distribution	see Sec. A.2.1
$\chi(\nu)$	Vaccinations rate (calculated imposing a fixed vaccine coverage)	Minimum of exponential distributions see Sec. A.2.1
$\mu(\nu)$	Birth rate depending on the time	Obtained from ISTAT see Sec A.2.1
$\delta_{a_i}(\nu)$	Death rates depending on time and on the age class	Obtained from ISTAT see Sec A.2.1
$\lambda_{a,b}$	Contact rates between subjects in the age classes a and b	see Table A.1
prob_boost	prob. of a natural boost occurrence	To be estimated
$prob\_infetionS$	prob. of infection success of a susceptible	To be estimated
$prob\_infetionR\_l1$	prob. of infection success of a recovered with minimum resistance	To be estimated

Table A.3: Parameters of the ESSN model.

are grouped in the following set:

$$\begin{aligned} T_{contact} = \{&ContactVi\_IpToRii\_a_i\_a_j, ContactVi\_IsToRii\_a_i\_a_j,\\ &ContactV\_IpToIs\_a_i\_a_j, ContactV\_IsToIs\_a_i\_a_j,\\ &ContactR\_IpToIs\_a_i\_a_j, ContactR\_IsToIs\_a_i\_a_j,\\ &ContactRi\_IpToRii\_a_i\_a_j, ContactRi\_IsToRii\_a_i\_a_j,\\ &ContactS\_IpToIp\_a_i\_a_j, ContactS\_IsToIp\_a_i\_a_j\}_{i,j=1,2,3} \end{aligned}$$

are defined by the following function:

$$f_t(\hat{x}(\nu), \nu) = prob(t) * \frac{\lambda_{a_i,a_j}}{x^{tot}_{a_j}(\nu)} \prod_i \hat{x}_i(\nu) \tag{A.4}$$

where $t \in T_{contact}$, $x^{tot}_{a_j}(\nu)$ represents the number of people in the a_j class at time ν, λ_{a_i,a_j} the contact rate between the a_i and a_j classes corresponds to the i-row and j-column of the contact matrix A.1. $prob(t)$ is a function which returns depending on the transition t if its rate has to be multiplied by a specific probability and it is defined as follows

$$prob(t) = \begin{cases} prob\_infectionS & t \in \{ContactS\_IpToIp\_a_i\_a_j, ContactS\_IsToIp\_a_i\_a_j\} \\ prob\_boost & t \in \{ContactRi\_IpToRii\_a_i\_a_j, ContactRi\_IsToRii\_a_i\_a_j, \\ & \quad ContactVi\_IpToRii\_a_i\_a_j, ContactVi\_IsToRii\_a_i\_a_j\} \\ prob\_infectionR\_l1 & t \in \{ContactV\_IpToIs\_a_i\_a_j, ContactV\_IsToIs\_a_i\_a_j, \\ & \quad ContactR\_IpToIs\_a_i\_a_j, ContactR\_IsToIs\_a_i\_a_j\} \end{cases}$$

Finally, $\prod_i \hat{x}_i(\nu)$ is the product of the average numbers of individuals in the input places of the transition t.

Transitions modeling deaths. Let us define the set of the transitions modeling the death of a person in age class a_i as

$$T_{death} = \{DeathS\_a_i, DeathIp\_a_i, DeathIp\_a_i, \\ DeathR\_a_i, DeathV\_a_i\}_{i=1,2,3}.$$

Then we can define the function providing the speed of a transitions $t \in T_{death}$ as

$$f_t(\hat{x}(\nu), \nu) = \delta_{a_i}(\nu)\hat{x}(\nu) \tag{A.5}$$

where $\delta_{a_i}(\nu)$ is the death rate with respect the age class a_i, and it changes its value depending on the current year:

$$\delta_{a_i}(\nu) = death[i, \nu], \tag{A.6}$$

where $death[i, \nu]$ is the death rate referred to the year given by ν and the age class a_i considering the rates matrix defined from 1974 to 2016 (columns) for each of three age classes (rows), reported in `https://github.com/qBioTurin/epimod/`.

Transitions modeling births. The birth events are modeled by the transition *Birth*, which is characterized by the following function:

$$f_t(\hat{x}(\nu), \nu) = \mu(\nu) x^{tot}(\nu), \tag{A.7}$$

where $\mu(\nu)$ is the birth rate per person estimated from 1974 to 2016, obtained from the ISTAT. Since this rate represents the mean number of Italian newborns (in the year ν) per person, it must be multiplied by the total number of Italians (in the year ν), i.e. $x^{tot}(\nu)$.

Transitions modeling vaccination. The vaccination process starts with the transition *FirstVaccination*, and continues (for the administrations of two further doses) by the *Vaccination* transition. Both of them are characterized by the following function:

$$f_t(\hat{x}(\nu), \nu) = \chi(\nu)\hat{x}(\nu), \tag{A.8}$$

where $\chi(\nu)$ is administration rate of one vaccine dose, computed as explained in Sec.A.2.1. This rate is then multiplied by $\hat{x}(\nu)$, which represents the number of newborns without vaccination, so in the color class NV, (referring to the *FirstVaccination* transition), or the number of newborns already vaccinated one or two times, (referring to the *Vaccination* transition). When the possibility of vaccination failure is also considered, than the function A.8 is multiplied to the vaccination failure probability p_v, obtaining the following rated for the *FirstVaccination* and *Vaccination* transitions:

$$f_t(\hat{x}(\nu), \nu) = \chi(\nu)\hat{x}(\nu) * (1 - p_v). \tag{A.9}$$

To model the vaccination failure further transitions are drawn in the model for simulating the vaccine administrations without an increasing of the resistance level, their rates are defined as follows:

$$f_t(\hat{x}(\nu), \nu) = \chi(\nu)\hat{x}(\nu) * (p_v). \tag{A.10}$$

A.2.3 Package functions

The *epimod* package exposes four main functions that enable the user to perform all the steps required to design, develop and analyse an epidemiological model of disease diffusion. Indeed, the main framework functions can be exploited to implement a workflow for studying the diffusion of a disease

in a population over the years. Starting from the graphical model *epimod* main functions allow one to derive (*model_generation*), calibrate (*sensitivity_analysis* and *model_calibration*) and to perform analysis through the developed model (*model_analysis*). Each step of the pipeline is self-contained in one function.

In the following, we provide the R commands used to generate the results presented in the paper. Furthermore, we show some examples of the input files required to run the analysis. For the complete script used for the paper analysis, please visit `https://github.com/qBioTurin/epimod`.

Function *model_generation*

The *model_generation* function provides the necessary tools to convert the graphic model to an executable one.

Parameters

- *net_fname* .PNPRO file storing the model as ESPN. In case there are multiple nets defined within the PNPRO file, the first one in the list is the will be automatically selected;
- *functions_fname* C++ file defining the functions managing the behaviour of general transitions.

```
library(epimod)

model_generation(net_fname = "~/input/Pertussis.PNPRO",
                 functions_fname = "~/input/transitions.cpp")
```

Example Files and functions

The header of a C++ function, to be correctly included and used in the solver, have to match the following specification:

- return value: *double* value representing the transition speed in the current marking;
- function name: the name specified as transition rate field in the GreatSPN GUI.
- Its input arguments:

- *double *Value*: data structure encoding the current marking;
- *map <string,int>& NumTrans*: data structure providing a match between transitions' name and their numeric identifier within the solver;
- *map <string,int>& NumPlaces*: data structure providing a match between places' names and their numeric identifiers within the solver;
- *const vector<string> & NameTrans*: data structure listing all transitions' names;
- *const struct InfTr* Trans*: transition's input places;
- *const int T*: identifier of the transition for which the speed must be computed;
- *const double& time*: current solution time;

```
double compute_transition_intensity(double *Value, map <string,int>&
    NumTrans, map <string,int>& NumPlaces,const vector<string> &
    NameTrans, const struct InfTr* Trans, const int T,
    const double& time)
{
    ifstream f("params_file_name");
    double param = ....; /* Read parameters value from file*/
    double intensity = 1.0;
    for (unsigned int k=0; k<Trans[T].InPlaces.size(); k++)
    {
            intensity *= pow(Value[Trans[T].InPlaces[k].Id],
                          Trans[T].InPlaces[k].Card);
    }
    return param * intensity;
}
```

Listing A.1: Example C++ function providing the behaviour of a general transition, following the Mass Action Law

Function *sensitivity_analysis*

The function *sensitivity_analysis* preforms a Monte Carlo sampling of the input model parameters. The user has to specify what parameters to be explored and how to generate them, providing either a base R function or a customised

one. In the latter case, functions can be provided through a R file and a .csv file listing all the functions to call, the parameters to use and a name file where the parameters will be saved. The list in the .csv file is used to link a specific function to a (set of) parameter(s). The user must be define a file name coherently with the one used in the general transitions file. After generating the necessary configurations, simulations are executed. The output of such simulations will be used to compute the PRCC, pinpointing the most relevant parameters. To this aim, the function finds out which parameters change through all the configurations generated and computes the PRCC only to them. The user must provide a function combining the model outputs to define the model response used in PRCC analysis.

Eventually, the function computes a ranking associating a score to each solver output. Again, the user has to provide a function to compute such score with respect to a reference data set and the model response.

Parameters

- *parameters_fname* .csv file listing all the parameters to generate through either base R functions or user defined ones.
- *functions_fname* File hosting the user defined functions to generate instances of the parameters
- *solver_fname* .solver file (generated in with the function model_generation)
- *f_time* Final solution time
- *s_time* Time step at which explicit estimates for the system are desired
- *reference_data* Data to compare with the simulations' results
- *distance_measure_fname* File containing the definition of a distance measure to rank the simulations'. Such function takes 2 arguments: the reference data and a simulation output. It has to return a vector with the same dimension as the input ones, containing the distance between the two input vectors. Please, note that file name and function name must match.
- *target_value_fname* File providing a function to obtain the relevant measure to evaluate the output. Please, note that file name and function name must match.

```
sensitivity_analysis(n_config = 2^12,
                     parameters_fname = "~/input/Functions_list.csv",
                     functions_fname = "~/input/Functions.R",
                     solver_fname = "~/Pertussis.solver",
                     f_time = 365*21,
                     s_time = 365,
                     timeout = "1d",
                     parallel_processors=40,
                     reference_data ="~/input/reference_data.csv",
                     distance_measure_fname="~/input/msqd.R",
                     target_value_fname="~/input/infects.R"
)
```

Example Files and functions

```
params <- function(n_params)
{
    x <- runif(n_params)
    return(x / sum(x))
}
```

Listing A.2: Example function to generate *n_params* summing up to 1

```
parms_file_name, parms, 5
other_parms_file_name, runif, 5
```

Listing A.3: Example list controlling the generation of two subsets of parameters, one through the user defined function *params* and the other through the base R function *runif*

```
infects<-function(output)
{
    ynames <- names(output)
    col_names <- "(InfectCount_a){1}[0-9]{1}"
    col_idxs <- c( which(ynames %in% grep(col_names, ynames,
        value=T)) )
    # Reshape the vector to a row vector
    ret <- rowSums(output[,col_idxs])
    return(as.data.frame(ret))
}
```

Listing A.4: A example of function combining the model outputs to define the model response used in PRCC analysis

```
msqd<-function(reference, output)
{
    ynames <- names(output)
    col_names <- "(InfectCount_a){1}[0-9]{1}"
    col_idxs <- c( which(ynames %in% grep(col_names, ynames,
        value=T)) )
    infects <- rowSums(output[,col_idxs])
    infects <- infects[-1]
    diff<-c(infects[1],diff(infects,differences = 1))
    ret <- sum(( diff - reference )^2 )
    return(ret)
}
```

Listing A.5: Example a function to compute a score with respect to a reference data set and the model response.

Output files The function *sensitivity_analysis* generates several output files. Supposing that the *out_fname* parameter being set as *sensitivity*, the relevant file produced are: *seeds_sensitiviy.RData*, *ranking_sensitivity.RData*, *parms_prcc_sensitivity.RData* and *prcc_sensitivity.RData*.

The function stores configurations information in *seeds_sensitiviy.RData* and *parms_prcc_sensitivity.RData* files. Initial and final seeds used in the execution are saved to *seeds_sensitiviy.RData* and they can be used either to extend the current experiment or to reproduce it. The file *parms_prcc_-sensitivity.RData* contains all the details concerning the simulation's settings.

Results concerning the ranking and the PRCC analysis are stored respectively in *ranking_sensitivity.RData* and in *prcc_sensitivity.RData*. The first file contains two columns, one for the simulation identifier and the other its score. To get the configuration used to run a specific simulation, users have to employ the identifier to index the configuration stored in *parms_prcc_sensitivity.RData* file.

In addition to this files, all the simulation's traces are saved into file matching the following pattern: *out_fname-identifier.trace* (e.g., sensitivity-1.trace).

Function *model_calibration*

The function *model_calibration* searches for a parameter combination providing the most accurate match w.r.t. reference data. In order to generate configuration, the user has to provide base R functions or customised functions to modify all the parameters not defined in the graphical model. This functions corresponds to those passed to the *sensitivity_analysis* function but must accept an additional parameter (refer Listing 6).

To drive the optimization process it is necessary to provide a scoring function used as objective function to be minimized. The user has to provide such function by following the same step as specified for the *sensitivity_analysis* function. Furthermore, the *model_calibration* function can be exploited to perform the calibration of several types of model solutions, just by changing the *solver_type* parameter.

Parameters

- *parameters_fname* .csv file listing all the parameters to generate through either stock R functions or user defined ones.
- *functions_fname* File hosting the user defined functions to generate instances of the parameters;
- *solver_fname* .solver file (generated in with the function model_generation);
- *solver_type* controls the type of solution adopted (ODE-E or ODE-RKF or ODE45 or LSODA or HLSODA or (H)SDE or HODE or SSA or TAUG or STEP);
- *f_time* Final solution time;

- *s_time* Time step at which explicit estimates for the system are desired;
- *reference_data* Data to compare with the simulations' results;
- *distance_measure_fname* File containing the definition of a distance measure to rank the simulations'. Such function takes 2 arguments: the reference data and a simulation output. It has to return a vector with the same dimension as the input ones, containing the distance between the two input vectors. Please, note that file name and function name must match;
- *target_value_fname* a function combining the model outputs to define the model response used in PRCC analysis. Please, note that file name and function name must match;
- *ini_v* Vector providing the initial condition of the optimization problem;
- *ub_v* vector defining the upper bound limits on the optimization problem variables;
- *lb_v* vector defining the lower bound limits on the optimization problem variables.

```
library(epimod)
model_calibration(parameters_fname = "~/input/Functions_list.csv",
                  functions_fname = "~/input/Functions.R",
                  solver_fname = "~/Pertussis.solver",
                  f_time = 365*21,
                  s_time = 365,
                  timeout = "1d",
                  parallel_processors=40,
                  reference_data = "~/input/reference_data.csv",
                  distance_measure_fname = "~/input/msqd.R",
                  # Vectors to control the optimization
                  ini_v = c(0.05, 0.07, 0.1, 1, 0, 1, 0, 0, 0, 0, 1, 0,
                      0, 0, 0),
                  ub_v = c(0.25, 0.1, 0.25, 1, 1, 1, 1, 1, 1, 1, 1, 1,
                      1, 1, 1),
                  lb_v = c(0, 0, 0, 1e-7, 1e-7, 1e-7, 1e-7, 1e-7, 1e-7,
                      1e-7,
                           1e-7, 1e-7, 1e-7, 1e-7, 1e-7),
                  ini_vector_mod = TRUE
)
```

Example Files and functions

```
params <- function(total, x)
{
    allocation <- total * x / sum(x)
    return(allocation)
}
```

Listing A.6: Example function to allocate some resource according to an allocation vector x.

Output files The function *model_calibration* generates several output files. Supposing that the *out_fname* parameter being set as *calibration*, the relevant file produced are: *seeds_calibration.RData*, *calibration_optim.RData*, *calibration_optim-trace.csv*.

The function stores initial and final seeds used in the execution to *seeds_calibration.RData*. The output of the optimization is stored to *calibration_optim.RData* (refer to GenSA package for further details). Nonetheless, the

optimization process is tracked through the output file *calibration_optim-trace.csv*, where at each optimization iteration the function stores the identifier, the score and the vector used to compute the current solution.
In addition to this files, all the simulation's traces are saved to file matching the following pattern: *out_fname-identifier.trace* (e.g., calibration-1.trace).

Function *model_analysis*

The function *model_analysis* allows one to execute and test the behaviour of the developed model. Furthermore, by changing the input parameters, it is possible to perform *what-if* analysis or forecasting the evolution of the diffusion process in the future.

The mechanism employed to provide the required parameters are the same used in the aforementioned functions.

Parameters

- *parameters_fname* .csv file listing all the parameters to generate through either stock R functions or user defined ones.
- *functions_fname* File hosting the user defined functions to generate instances of the parameters;
- *solver_fname* .solver file (generated in with the function model_generation);
- *solver_type* controls the type of solution adopted (ODE-E or ODE-RKF or ODE45 or LSODA or HLSODA or (H)SDE or HODE or SSA or TAUG or STEP);
- *int_fname* File containing the initial marking, if required;
- *f_time* Final solution time;
- *s_time* Time step at which explicit estimates for the system are desired;
- *ini_v* Vector providing the initial condition of the optimization problem;

```
library(epimod)
model_analysis(solver_fname = "~/Pertussis.solver",
              f_time = 365*43,
              s_time = 365,
              n_config = 1,
              n_run = 2^8,
              parallel_processors = 40,
              solver_type = "SSA",
              parameters_fname = "~/input/Functions_list_wif.csv",
              functions_fname = "~/input/Functions.R",
              timeout = '1d',
              ini_v = c(0.1028314,0.06422971,0.1173022,0.3996977,
                        0.04611963,1.082384,0.2943042,0.02526472,
                        0.01475493,8.009888e-08,0.9688963,0.3208456,
                        0.06788778,0.000676801,0.0001424256),
              ini_vector_mod = TRUE)
```

Output files Supposing that the *out_fname* parameter being set as *analysis*, the function *model_analysis* generates one output storing initial and final seeds used in the execution (*seeds_analysis.RData*). In addition to this file, all the simulation's traces are saved to file matching the following pattern: *out_fname-identifier.trace* (e.g., analysis-1.trace).

A.3 Supplementary information: RRMS model

A.3.1 Second model

General transitions

Let us note that in the next section we extend the functions already introduced in Sub Section 8.2.2, except those characterizing the movements, in order to consider the new color instances regarding the positions, which were not present in that section. In particular, we recall that transitions which represent i) the killing of a cell, e.g., *TregKillsTeff* or *TeffKillsODC*, ii) the entry of cells into the system, such as *EBVinj*, iii) the activation of T cells, e.g., *TeffActivation*, iv) the duplication of a cell, e.g., such as *TregDup*, and v) the cell movements e.g., *TregMovements*, are all modelled as general transitions

because they do not follow the Mass Action (MA) law.
Let us recall the following notations:

- $f_{\langle t,c\rangle}(\hat{x}(\nu),\nu)$ is the speed of the transition $t \in T_g$ and $\hat{x}(\nu)$ represents the vector of the average number of tokens for all the input places. For brevity, when the function will not depend on the color instance c, we will omit the c and we will simplify the specification of the function in the following way $f_t(\hat{x}(\nu),\nu)$.

- $exp(-\frac{y}{Cost})$ represents a coefficient varying in $[0,1]$ and it is directly related to the y value. This is usually multiplied by a constant rate, so that an increase of y involves a decrease in the associated rate.

- $x_{tot_{\langle p_x,p_y,p_z\rangle}}(\nu)$ is the total number of cells in the grid cell with coordinates $\langle p_x,p_y,p_z\rangle$ at time ν. For simplicity, we will omit the indication of the position $\langle p_x,p_y,p_z\rangle$ and of the time ν, so in the case of $x_{tot_{\langle p_x,p_y,p_z\rangle}}(\nu)$, it will be x_{tot}. In general the notation will be $x_{Name_{\langle p_x,p_y,p_z\rangle}}(\nu) = x_{Name}$.

All the general transitions of the model are now explained in details and all the constants are summarized in Tables 8.3 and 8.6.

- *EBVinj* and *DACinj* inject into the system specific quantities of EBV and DAC respectively at fixed time points;

- *FromTimoREG*, *FromTimoEFF*, and *NKentry* are the transitions which keep in a constant range the number of $RestingTreg$, $RestingTeff$, and NK respectively. Their speed is defined as

$$f_{\langle FromTimoREG,p_x,p_y,p_z\rangle}(x_{ResTreg},\nu) = q_{RestTreg} * (1 - x_{ResTreg}/63);$$
$$f_{\langle FromTimoEFF,p_x,p_y,p_z\rangle}(x_{ResTeff},\nu) = q_{RestTeff} * (1 - x_{ResTeff}/1687);$$
$$f_{\langle NKentry,p_x,p_y,p_z\rangle}(x_{NK},\nu) = q_{NK} * (1 - x_{NK}/375),$$

  where $x_{ResTreg}$, $x_{ResTeff}$, and x_{NK} are the numbers of cells in the input places (i.e. $RestingTreg$ for *FromTimoREG*, etc) at time ν and position $\langle p_x,p_y,p_z\rangle$. Then q_- corresponds to the quantity injected in the output place to preserve the cell quantity, i.e. 63 for the $RestingTreg$, 1687 for the $RestingTeff$ and 375 for the NK.

- *TregActivation* and *TeffActivation* transitions model the activation of the Teff and Tref cells. In particular, these are specified as general

transitions to simulate a reduced Teff activation velocity with respect to a decreasing virus presence, and a Treg activation velocity proportional to the number of Teffs and inversely proportional to the number of EBV particles (allowing the Teff to annihilate the virus). So the functions are defined as

$$f_{\langle TregActivation,p_x,p_y,p_z\rangle}(\hat{x}(\nu),\nu) = r_{TregA} * \frac{x_{Teff}}{(x_{Teff}+x_{EBV}+1)} * x_{ResTreg};$$

$$f_{\langle TeffActivation,p_x,p_y,p_z\rangle}(\hat{x}(\nu),\nu) = r_{TeffA} * (1-exp(-\frac{x_{EBV}}{C_{EBV}})) * x_{ResTeff},$$

where r_{TregA} and r_{TeffA} are the activation constant rates for the Treg and Teff respectively. $\hat{x}(\nu)$, in case of the $TregActivation$ transition, consists of the variables $x_{ResTreg}, x_{EBV}, x_{Teff}$ corresponding respectively to places $RestingTeff, EBV$ and $Teff$; differently the $TeffActivation$ transition speed depends on $x_{ResTeff}$ and x_{EBV}. Finally constant C_{EBV} is related to the EBV particles and it is meant to reduce the activation rate with the decreasing of the virus presence.

- *MemActivation* is defined as

$$f_{\langle MemActivation,p_x,p_y,p_z\rangle}(\hat{x}(\nu),\nu) = \begin{cases} 0 & \nu < t_{2inj}, \\ r_{MemA} * x_{Mem}(\nu) & \nu \geq t_{2inj}, \end{cases}$$

where

$$r_{MemA} = 2 * r_{TeffA} * (1-\exp(-\frac{x_{Mem}(\nu)}{C_{Mem}}) * (1-\exp(-\frac{x_{EBV}}{C_{EBV}}),$$

and t_{2inj} is the time corresponding to the second EBV injection. We considere this velocity as zero since the T Memory effectors start to react after the first virus occurrence. $\hat{x}(\nu) = (x_{Mem}(\nu), x_{EBV}(\nu))$ is the marking vector storing the number of T Memory effectors (no position dependency) and EBV particles in the place $\langle p_x, p_y, p_z\rangle$, respectively at time ν. While C_{Mem} and C_{EBV} constant related to the Memory and EBV cells needed to slow down the activation rate with the decreasing of EBV and Memory cells. This because we have to leave a minimum number of T Memory effectors into the system. So when in the system

there are large number of EBV particles and of T Memory effectors, the activation speed reaches its maximum that is given by twice the velocity of the Teff cells, r_{TeffA}.

- The speed of all the transitions that model the killing of a specific cell is defined as follows:

$$\forall t \in \{\textit{TregKillsTeff}, \textit{TeffKillsODC}, \textit{TeffKillsEBV}, \textit{NKKillsTcell}\}$$

then

$$f_{\langle t,p_x,p_y,p_z\rangle}(\hat{x}(\nu),\nu) = \frac{1}{x_{tot}} * r_t * \prod_i \hat{x}_i(\nu),$$

where $\prod_i \hat{x}_i(\nu)$ is the product of the average numbers of tokens in the input places of transition t, r_t is the constant rate associated with transition t , and $\frac{1}{x_{tot}}$ represents the probability that a specific meeting between two different cells occurs in the specific grid cell.

- *TregDup* transition models the Treg duplication, its speed depends proportionally on the amount of IL-2 and is inversely proportional on the number of DAC cells (to simulate the reduced duplication velocity during the daclizumab therapy), and it is defined as follows:

$$f_{\langle TregDup,p_x,p_y,p_z\rangle}(\hat{x}(\nu),\nu) = \eta_{TrD}(\hat{x}(\nu),\nu) * x_{Treg} * x_{IL2} * \frac{1}{x_{tot}},$$

with

$$\eta_{TrD}(\hat{x}(\nu),\nu) = r_{TregDup} * (1 - exp(-\frac{x_{IL2}}{C_{IL2}})) * (exp(-\frac{x_{DAC}}{C_{DAC}})).$$

where $r_{TregDup}$ is the constant Treg duplication rate,
$\hat{x}(\nu) = \{x_{Treg},\ x_{IL2},\ x_{DAC}\}$ and C_{IL2} and C_{DAC} are the constants related to the IL-2 and DAC cells to slow down the duplication velocity with an increasing number of DACs and a decreasing number of IL-2 proteins.

- Considering the Teff duplication event we have to distinguish two possible cases: 1) the Teff symmetric duplication with probability p_{eff}^{dup} and a Teff asymmetric duplication, implying the T Memory effector differentiation, with probability $p_{eff}^{mem} = 1 - p_{eff}^{dup}$. This is modeled exploiting two different transitions: *TeffDup_Sym* and *TeffDup_Asym*. So let us define

$$r_{dup}^{eff} = \eta_{TeD}(\hat{x}(\nu),\nu) * x_{Teff} * x_{IL2} * \frac{1}{x_{tot}}$$

then the speed of these two transitions is defined as:

$$f_{\langle TeffDup\_Sym\rangle,p_x,p_y,p_z}(\hat{x}(\nu),\nu) = p_{eff}^{dup} * r_{dup}^{eff}$$

and

$$f_{\langle TeffDup\_Asym,p_x,p_y,p_z\rangle}(\hat{x}(\nu),\nu) = p_{eff}^{mem} * r_{dup}^{eff},$$

with

$$\eta_{TeD}(\hat{x}(\nu),\nu) = r_{TeffDup} * (1 - exp(-\frac{x_{IL2}}{C_{IL2}})) * (exp(-\frac{x_{DAC}}{C_{DAC}})).$$

Where $r_{TeffDup}$ is the constant Teff duplication rate,
$\hat{x}(\nu) = \{x_{Teff},\ x_{IL2},\ x_{DAC}\}$.

For clarity, the notation for the movements functions of the color combinations $\langle p_x, p_y, p_z\rangle$ and $\langle q_x, q_y, q_z\rangle$, representing the location coordinates, is simplified to $\langle \mathbf{p}\rangle$ and $\langle \mathbf{q}\rangle$, respectively.

- ***TeffMovement*** simulates the movement of Teff cells from point (with coordinates represented by the color combination) $\langle \mathbf{p}\rangle$ to point $\langle \mathbf{q}\rangle$. The speed of this movement (the rate of transition TeffMovement) is inversely related to the number of EBV cells in $\langle \mathbf{p}\rangle$ and depends on the number of EBV in $\langle \mathbf{q}\rangle$ such that a greater number of EBV cells leads to a higher probability to move into that location. This is captured by the following formula

$$f_{\langle TeffMovement,\mathbf{p},\mathbf{q}\rangle}(\hat{x}(\nu),\nu) = r_{moves} * (exp(-\frac{x_{EBV_{\langle \mathbf{p}\rangle}}}{C_{EBV}})) * p_{\langle \mathbf{q}\rangle}^{Teff} * x_{Teff_{\langle \mathbf{p}\rangle}}$$

where r_{moves} is an experimental coefficient that we set equal to 0.1;
$exp(-\frac{x_{EBV_{\langle \mathbf{p}\rangle}}}{C_{EBV}})$ is a term that accounts for the fact that the velocity of the movement is inversely related to the number of EBV cells in the starting point going to 0 in a manner that is slower than $1/x_{EBV_{\langle \mathbf{p}\rangle}}$;
$p_{\langle \mathbf{q}\rangle}^{Teff} = \frac{x_{EBV_{\langle \mathbf{q}\rangle}}}{EBV_{tot}}$ represents the probability to move in the cell with coordinates $\langle \mathbf{q}\rangle$ where EBV_{tot} is the total number of EBV in the grid at time ν; and C_{EBV} is an experimental constant that we set equal to 1000. All these quantities are functions of the time ν which is omitted in the formula to keep the notation simpler.

- ***TregMovement*** represents the movements of the Treg cells from point $\langle \mathbf{p} \rangle$ to point $\langle \mathbf{q} \rangle$. Similarly to what explained for transition TeffMovement the speed is inversely related to the number of Teff cells in $\langle \mathbf{p} \rangle$ (term $exp(-\frac{x_{Teff_{\langle \mathbf{p} \rangle}}}{C_{Teff}})$) and depends on the number of Teffs in $\langle \mathbf{q} \rangle$ (term $p^{Treg}_{\langle \mathbf{q} \rangle} = \frac{x_{Teff_{\langle \mathbf{q} \rangle}}}{Teff_{tot}}$), such that a greater number of Teffs leads to a higher probability to reach that location.

$$f_{\langle TregMovement, \mathbf{p},\mathbf{q} \rangle}(\hat{x}(\nu), \nu) = r_{moves} * (exp(-\frac{x_{Teff_{\langle \mathbf{p} \rangle}}}{C_{Teff}})) * p^{Treg}_{\langle \mathbf{q} \rangle} * x_{Treg_{\langle \mathbf{p} \rangle}}$$

  Again, in our experiment we fixed $r_{moves} = 0.1$ and $C_{Teff} = 800$.

- ***EBVMovement*** simulates the EBV movements from point $\langle \mathbf{p} \rangle$ to point $\langle \mathbf{q} \rangle$. In this case we assume that the probability to move is equally distributed among all the grid cells.

$$f_{\langle EBVMovement, \mathbf{p},\mathbf{q} \rangle}(\hat{x}(\nu), \nu) = r_{moves} * p^{EBV}_{\langle \mathbf{q} \rangle} * x_{EBV_{\langle \mathbf{p} \rangle}}$$

  Also in this case, in our experiment we fixed $r_{moves} = 0.1$.

- ***DACMovement*** simulates the DAC movements from point $\langle \mathbf{p} \rangle$ to point $\langle \mathbf{q} \rangle$. This is inversely related to the number of T-cells (Treg + Teff) in $\langle \mathbf{p} \rangle$ as for the TeffMovement and TregMovement cases and directly proportional to the number of T-cells in $\langle \mathbf{q} \rangle$.

$$f_{\langle DACMovement, \mathbf{p},\mathbf{q} \rangle}(\hat{x}(\nu), \nu) = r_{moves} * (exp(-\frac{x_{Tcells_{\langle \mathbf{p} \rangle}}}{C_{Tcell}})) * p^{DAC}_{\langle \mathbf{q} \rangle} * x_{DAC_{\langle \mathbf{p} \rangle}}$$

  The quantities $r_{moves} = 0.1$ and $C_{Tcell} = 1000$ were used in this last case.

A.3.2 Third model

General transitions

Let us recall that by the definition of the ESSN formalism, transitions which do not follow the MA law are modelled as general transitions because. In details

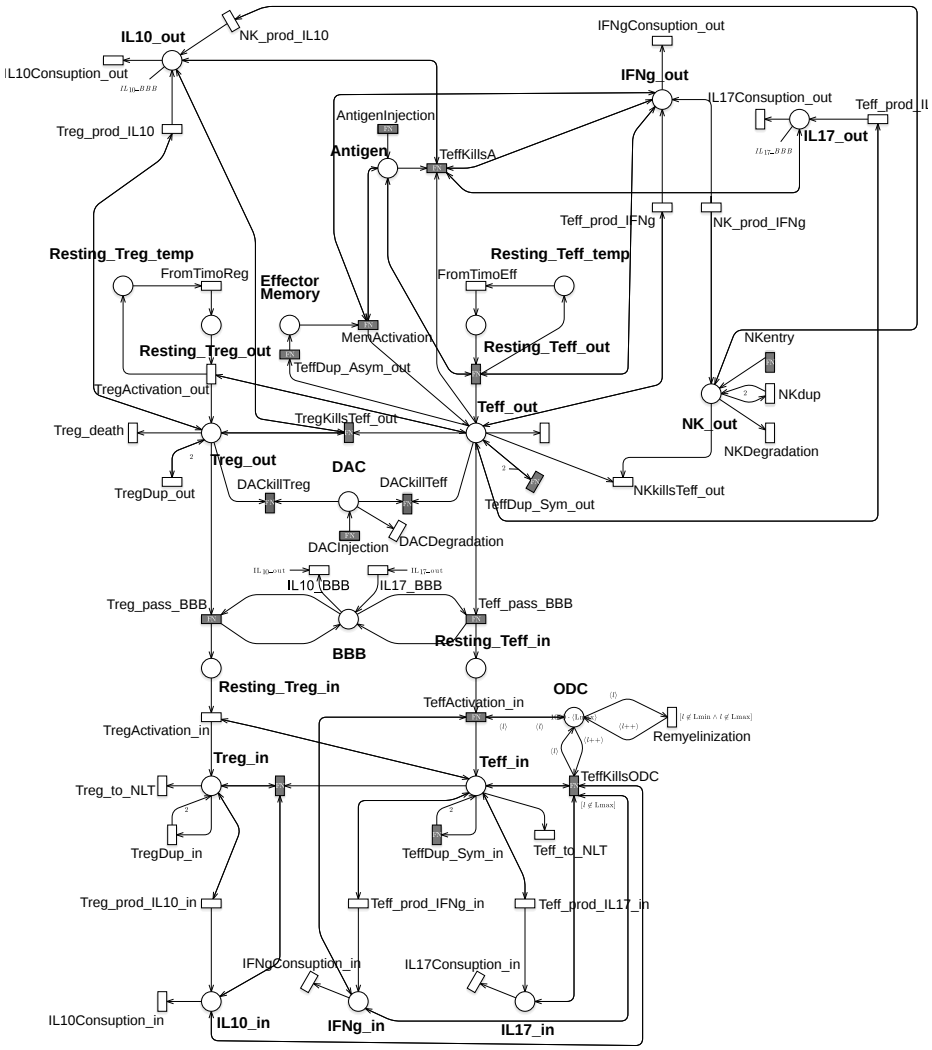


Figure A.1: RRMS model represented by exploiting the ESSN graphical formalism. Name of the places are represented in bold text.

the speed of general transition $t \in T_g$ is defined by a function $f_{\langle t,c\rangle}(\hat{x}(\nu),\nu)$, where $\hat{x}(\nu)$ represents the vector of the average number of tokens for all the transition input places at time ν and $\langle t,c\rangle$ is the instance transition. Given that the place ODC is the only place with an associated color domain, just the functions associated with transitions connected with this place will depend on the color instance c, for the other cases the function will be defined as $f_t(\hat{x}(\nu),\nu)$.
In the ESSN representing the RRMS depicted in Fig. A.1, the general transitions are 15 modeling the i) the killing of a cell, e.g., *TregKillsTeff_in* or *TeffKillsODC*, ii) the entry of NK cells into the system, *NKentry*, iii) the activation of T cells, e.g., *TeffActivation_in*, iv) the duplication of a cell, e.g., such as *TeffDup_in*, and v) the injections of the DAC and Antigen into the system, i.e., *DACInjection* and *AntigenInjection*.

Duplication. Considering the Teff duplication event we have to distinguish two possible cases: 1) the Teff symmetric duplication with probability $\rho^{dup} = 2/3$ and a Teff asymmetric duplication, implying the T Memory effector differentiation, with probability $\rho^{mem} = 1 - \rho^{dup}$. This is modeled exploiting three different transitions: *TeffDup_Sym_out*(*_in*) and *TeffDup_Asym_out*. In the CNS we model only the symmetric differentiation. The asymmetric differentiation give rise to the production of Effector memory cells. These last remain in blood circulation and are able to respond faster to antigen stimulation.The asymmetric differentiation occurs in peripheral blood circulation, then the produced effector memory cells can migrate to all other tissues (e.g. CNS).

So let define

$$r^{in}_{dup} = x_{Teff\_in}(\nu) * p_{Teff\_Dup}$$
$$r^{out}_{dup} = x_{Teff\_out}(\nu) * p_{Teff\_Dup}$$

then these two transitions are defined as:

$$f_{TeffDup\_Sym\_in}(\hat{x}(\nu),\nu) = \rho^{dup} * r^{in}_{dup}$$
$$f_{TeffDup\_Sym\_out}(\hat{x}(\nu),\nu) = \rho^{dup} * r^{out}_{dup}$$

and

$$f_{TeffDup\_Asym\_out}(\hat{x}(\nu),\nu) = \rho^{mem} * r^{out}_{dup},$$

Where p_{Teff_Dup} is the constant Teff duplication rate, see Table A.9, and $\hat{x}(\nu) = x_{Teff\_in}$ when the transition considered is the *TeffDup_Sym_in*, otherwise $\hat{x}(\nu) = x_{Teff\_out}$.

Activation. The *TeffActivation_out* and *TeffActivation_in* transitions model the activation of the Teff cells in the peripheral lymphonode/blood vessel (*_out*) and the CNS (*_in*).
The activation of the Teff cells is modeled by the following functions

$$\begin{aligned} f_{TeffActivation\_out}(\hat{x}(\nu),\nu) &= p_{TeffActivation} * x_{RestingTeff\_out}(\nu) * x_{Antigen}(\nu) \\ &\quad * (0.5 + exp(-x_{IFNg\_out}(\nu)/Cifn)); \\ f_{\langle TeffActivation\_in,l\rangle}(\hat{x}(\nu),\nu) &= p_{TeffActivation} * x_{RestingTeff\_in}(\nu) * x_{ODC\_l}(\nu) \\ &\quad * (0.5 + exp(-x_{IFNg\_in}(\nu)/Cifn)), \end{aligned}$$

where $p_{TeffActivation}$ is the constant Teff activation rate, see Table A.9, and $\hat{x}(\nu) = \{x_{RestingTeff\_out}, x_{Antigen}, x_{IFNg\_out}\}$ when the transition considered is the *TeffActivation_out*, otherwise $\hat{x}(\nu) = \{x_{RestingTeff\_in}, x_{ODC\_l}, x_{IFNg\_in}\}_{l \in cd(ODC)}$. Let us note that the term $(0.5 + exp(-x_{IFNg\_in(\_out)}(\nu)/Cifn))$ is a coefficient varying in $[0.5, 1.5]$ respectively to the concentration of IFNγ, more is present into the system slower is the velocity. In particular when there is no IFNγ then to the velocity is associated the highest value, (i.e., 1.5). otherwise it decreases until 0.5 .

Similarly, the transition *MemActivation* modeling the activation of T Memory effectors only in the peripheral lymphonode/blood vessel is defined as

$$f_{MemActivation}(\hat{x}(\nu),\nu) = \begin{cases} 0 & \nu < t_{2inj}, \\ r_{MemA} * x_{EffectorMemory}(\nu) & \nu \geq t_{2inj}, \end{cases}$$

where

$$r_{MemA} = 2 * p_{TeffActivation} * x_{Antigen}(\nu) * (0.5 + exp(-x_{IFNg\_out}(\nu)/Cifn)),$$

and t_{2inj} is the time corresponding to the second antigen injection. We are considering the velocity of this transition as zero $\forall \nu < t_{2inj}$, since we are assuming that the T Memory effectors start to react after the first virus occurrence, with twice the velocity of the Teff cells (for this reason we have $2 * p_{TeffActivation}$).

NK entry. The *NKentry* transition keeps in a constant range around 30 ([121, 87]) the number of NK_out. It is defined by the following function:

$$f_{NKentry}(\hat{x}(\nu),\nu) = \begin{cases} 0 & x_{NK\_out}(\nu) > 30, \\ 0.267 * (30 - x_{NK\_out}(\nu)) & x_{NK\_out}(\nu) \leq 30. \end{cases}$$

Killing. The general transitions modeling the killing of specific cell are the following:

- the *TregKillsTeff_out* and *TregKillsTeff_in* modeling the controlling action of the Treg over the Teff cells. These are defined as follows:

$$\begin{aligned} f_{TregKillsTeff\_out}(\hat{x}(\nu),\nu) &= p_{TregKillsTeff} * x_{Treg\_out}(\nu) * x_{Teff\_out}(\nu) \\ &\quad * (1 - exp(-x_{IL10\_out}(\nu)/cIL10)), \\ f_{TregKillsTeff\_in}(\hat{x}(\nu),\nu) &= p_{TregKillsTeff} * x_{Treg\_in}(\nu) * x_{Teff\_in}(\nu) \\ &\quad * (1 - exp(-x_{IL10\_in}(\nu)/cIL10)), \end{aligned}$$

  where the coefficient $(1 - exp(-x_{IL10\_in}(\nu)/cIL10))$ varies in [0,1] representing the Treg cell needs IL-10 for suppression of the Teff cells. Indeed, with an increasing number of IL-10 (i.e., the coefficient goes to 1) the transition velocity increases, otherwise it decreases.

- The *TeffkillsA* and *TeffKillsODC* modeling the annihilation of the pathogen by the Teff action and the ODC damage due to Activated Teff cells, respectively, are defined as follows:

$$\begin{aligned} f_{TeffkillsA}(\hat{x}(\nu),\nu) &= p_{TeffkillsA} * \Theta^{out}(\nu) * x_{Antigen}(\nu) * x_{Teff\_out}(\nu), \\ f_{\langle TeffKillsODC,l\rangle}(\hat{x}(\nu),\nu) &= p_{TeffKillsODC} * \Theta^{in}(\nu) * x_{ODC_l}(\nu) * x_{Teff\_in}(\nu), \end{aligned}$$

  with $l \in cd(ODC)$, Let define $\Theta^{in(/out)}$ as a coefficient varying in $[0.5, 1.5]$ which takes in account the pro- (IL-17, IFNγ) and anti-inflammatory (IL-10) cytokines in order to increase the velocity of the Teff action when more pro-inflammatory cytokines are present into the system, or

decrease it otherwise. These coefficients are defined as follows:

$$\Theta^{out} = 1 + 0.5 * \frac{x_{IL17\_out}(\nu) + x_{IFNg\_out}(\nu) - x_{IL10\_out}(\nu)}{x_{IL17\_out}(\nu) + x_{IFNg\_out}(\nu) - x_{IL10\_out}(\nu)},$$
$$\Theta^{in} = 1 + 0.5 * \frac{x_{IL17\_in}(\nu) + x_{IFNg\_in}(\nu) - x_{IL10\_in}(\nu)}{x_{IL17\_in}(\nu) + x_{IFNg\_in}(\nu) - x_{IL10\_in}(\nu)}.$$

- Finally the Daclizumab action to control the Treg and Teff cells spreading is modeled by the transitions DACkillTeff and DACkillTreg, whose functions are defined as follows:

$$f_{DACkillTeff}(\hat{x}(\nu), \nu) = p_{DACkill} * \frac{x_{Treg\_out}(\nu)}{x_{Treg\_out}(\nu) + x_{Teff\_out}(\nu)} * x_{DAC}(\nu)$$
$$f_{DACkillTreg}(\hat{x}(\nu), \nu) = p_{DACkill} * \frac{x_{Teff\_out}(\nu)}{x_{Treg\_out}(\nu) + x_{Teff\_out}(\nu)} * x_{DAC}(\nu)$$

  where the coefficient $\frac{1}{x_{Treg\_out}(\nu) + x_{Teff\_out}(\nu)}$ scales the velocity with respect to the number of Teff and Treg cells.

The parameters $p_{\text{Transition name}}$ are defined in Table A.9.

Injections. The *AntigenInjection* and *DACinjection* transitions inject into the system specific quantities of Antigen and Daclizumab respectively at fixed time points. These transitions are modeled by the occurrence of discrete events during the simulation which modify the model marking in specific time points (feature implemented in the `Epimod` Package).

Data

For the calibration of the model we used biological data from eight relapsing MS patients and eight healthy donors (HDs). A blood sample and a cerebrospinal fluid (CSF) sample of MS subjects were collected at the onset of the disease. Since CSF withdrawal is an invasive practice, for HDs only a blood sample was obtained. The flow cytometer cell counts have been used to set the initial marking of the net. In details, the cells were isolated from blood and *in-vitro* activated as specified in the methods section of [129]. After 18 hours cells were "tagged" with a mixture of fluorescent antibodies directed

towards proteins that identify the culture of interest. Regarding T cells, the combinations of multiple antibodies can be used to identify a specific cellular subtype of interest. In our case, the total of $CD4^+$ T cells / ml of blood or $CD4^+$ T cells / ml of CSF was counted, and, among them, the percentage of cells producing IL-17, IFNγ or IL-10 in blood and CSF was identified, see Table A.5. Referring to the Table A.5 in the main paper, we related those numbers to 1 ul and identified the values to be included in our model. In particular, we infer the numbers of T effector cells from the sum of IFNg values and IL-17 values, and the the number of regulatory T cells from IL-10 values. Differently, to attribute a value for circulating cells in the CNS in healthy subjects we identified a threshold value from the literature and then attributed random values below this threshold [41].
The individual cell counts of MS patients and healthy subjects were used to fit the model parameters.

Tables and Figures

Place	Number of cells/mmˆ 3	Reference
*R*esting_Teff_out	1689	[116]
*R*esting_Treg_out	63	[116]
*N*K_out	30	[121, 87]
*I*L-17_out	8	[129]
*I*L-10_out	13	[129]
*I*FNg_out	42	[129]
*I*L-17_in	1	[129]
*I*L-10_in	1	[129]
*I*FNg_in	1	[129]
*O*DC_Lmax	500	[116]

Table A.6: Initial marking of the model.

Patient ID	IFNγ-producing cells (blood)	IL17-producing cells (blood)	IL10-producing cells (blood)	IFNg-producing cells (CSF)	IL17-producing cells (CSF)	IL10-producing cells (CSF)
MS1	169.22	30.83	3.52	13.24	3.31	0.06
MS2	21.56	5.33	1.02	5.59	1.71	0.06
MS3	71.10	15.45	2.46	16.10	0.60	0.10
MS4	15.44	21.34	1.54	3.36	0.88	0.04
MS5	148.65	28.66	2.27	17.80	2.87	0.10
MS6	60.31	15.80	4.38	12.27	1.18	0.14
MS7	377.13	53.86	5.13	29.64	2.29	0.11
MS8	156.74	21.87	8.26	10.15	1.64	0.13
mean MS	117	25	3	12.77	1.85	0.09
HD1	42.37	21.19	26.98	0.30	0.30	0.40
HD2	2.40	13.20	16.80	0.80	0.20	1.00
HD3	8.70	7.83	8.18	0.20	0.80	1.00
HD4	69.30	3.15	11.76	1.00	1.00	1.00
HD5	16.15	2.38	8.27	0.10	0.80	0.10
HD6	18.45	1.64	6.48	2.00	1.00	2.00
HD7	63.00	7.88	10.94	0.10	0.10	0.80
HD8	97.27	13.70	12.06	0.20	1.90	1.90
mean HD	42	8	13	0.59	0.76	1.03

Table A.5: Biological data from eight relapsing MS patients and eight HD subjects. Values are expressed as number of cells/mm$^{\hat{}}$ 3.

Transition	Rate value
FromTimoREG	0.317
FromTimoEFF	0.296
NKdup	$1/24h^{-1}$
NKDegradation	$1/24h^{-1}$
Teff_death	$1/24h^{-1}$
Teff_to_NLT	$1/24h^{-1}$
Treg_death	$1/24h^{-1}$
Treg_to_NLT	$1/24h^{-1}$
Treg_prod_IL10 [2]	0.05556
Teff_prod_IL17	0.00895
Teff_prod_IFNg	0.0466
DACDegradation [3]	0.001444057

Table A.8: List of the fixed parameters.

Parameter	Transition	Healthy Value	MS Value
$p_{Teff\_Activation}$	$TeffActivation\_out$ $MemActivation$ $TeffActivation\_in\_l\_le1$ $TeffActivation\_in\_l\_le2$ $TeffActivation\_in\_l\_le3$ $TeffActivation\_in\_l\_le4$ $TeffActivation\_in\_l\_le5$	0.015	0.018
$p_{Treg\_Activation}$	$TregActivation\_in$ $TregActivation\_out$	4e-04	7e-05
p_{Treg_Dup}	$TregDup\_in$ $TregDup\_out$	0.006	–
p_{Teff_Dup}	$TeffDup\_Asym\_out$ $TeffDup\_Sym\_in$ $TeffDup\_Sym\_out$	0.04	–
$p_{TeffKillsODC}$	$TeffKillsODC\_l\_le1$ $TeffKillsODC\_l\_le2$ $TeffKillsODC\_l\_le3$ $TeffKillsODC\_l\_le4$	6e-04	–
p_{TrkTe}	$TregKillsTeff\_in$ $TregKillsTeff\_out$	0.02	–
p_{TekA}	$TeffkillsA$	6e-04	–
$p_{Pass\_BBB\_treg}$	$Treg\_pass\_BBB$	0.45	–
$p_{Pass\_BBB\_teff}$	$Teff\_pass\_BBB$	0.005	–
$p_{NKkillsTeff}$	$NKkillsTeff\_out$	0.01	–
$p_{NK\_prod\_IFNg}$	NK_prod_IFNg	0.03	–
$p_{NK\_prod\_IL10}$	NK_prod_IL10	0.045	–
p_{IL17_BBB}	$IL17\_BBB$	0.0115	–
p_{IL10_BBB}	$IL10\_BBB$	0.0765	–
$p_{Remyelinization}$	$Remyelinization\_l\_le2$ $Remyelinization\_l\_le3$ $Remyelinization\_l\_le4$	0.01	–
$p_{IL10Consuption}$	$IL10Consuption\_out$ $IL10Consuption\_in$	0.09	–
$p_{IL17Consuption}$	$IL17Consuption\_out$ $IL17Consuption\_in$	0.03	–
$p_{IFNgConsuption}$	$IFNgConsuption\_out$ $IFNgConsuption\_in$	0.05	–
$Cifn$	as $p_{Teff\_Activation}$	20	–
$CIL10$	as $p_{TregKillsTeff}$	10	–

Table A.9: List of the calibrated parameters to fit with the real data, in both the healthy and MS patient.

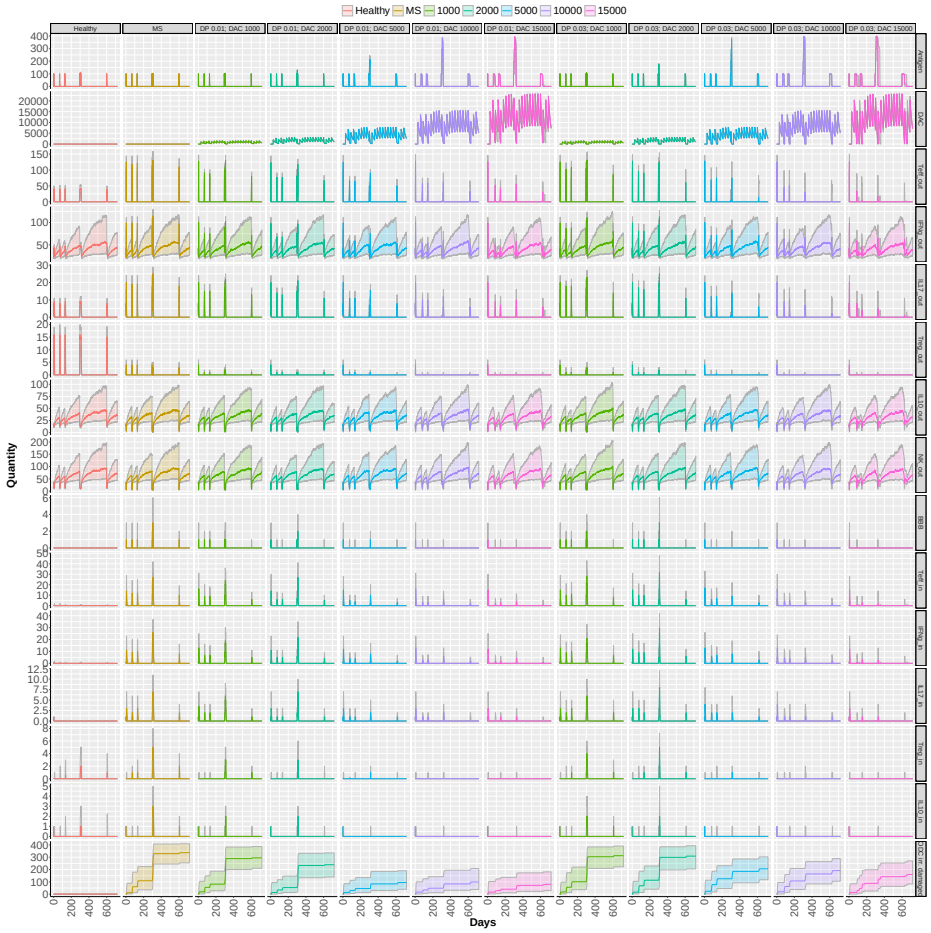


Figure A.2: Stochastic simulations considering the *early therapy*. Different colors are associated with quantity of DAC injected for each scenario, from 1000 to 15000 cells. The first two column represent the healthy and MS scenarios. Two drug potencies (called DP) are showed, i.e., 0.01 e 0.03.

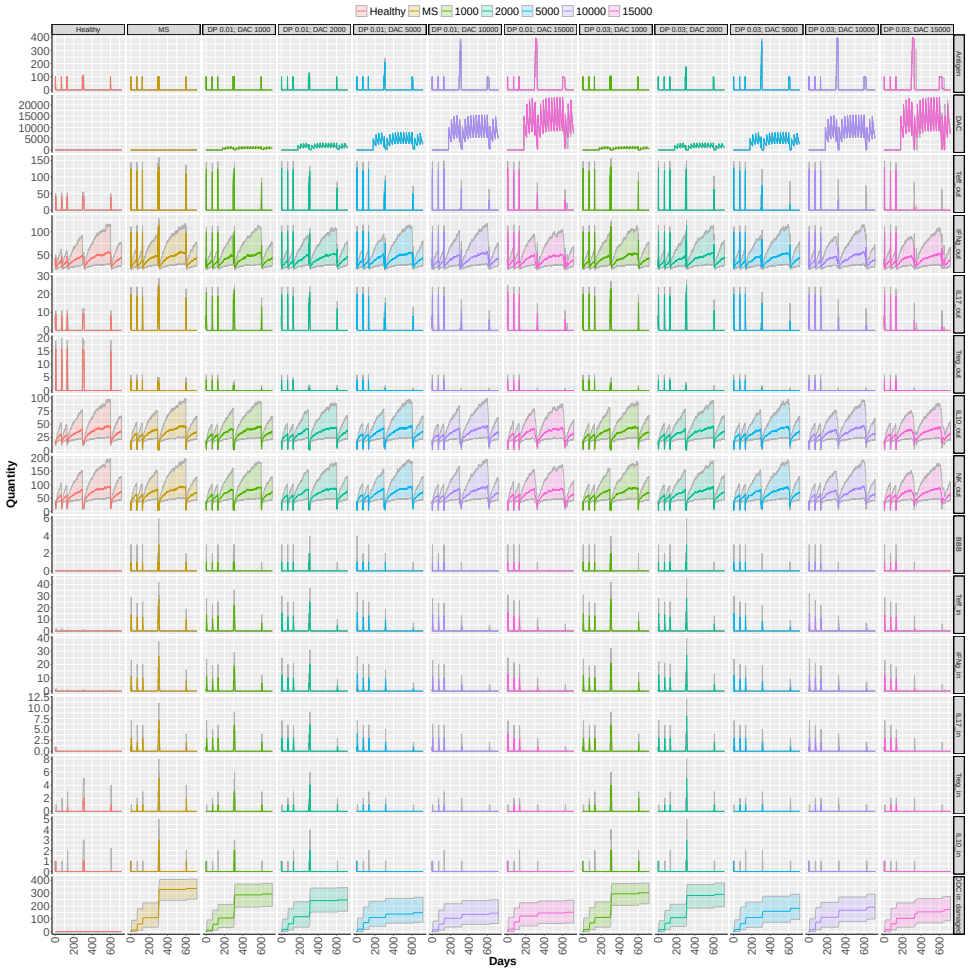


Figure A.3: Stochastic simulations considering the *late therapy*. Different colors are associated with quantity of DAC injected for each scenario, from 1000 to 15000 cells. The first two column represent the healthy and MS scenarios. Two drug potencies (called DP) are showed, i.e., 0.01 e 0.03.

A.4 Supplementary information: COVID19 model

A.4.1 Parameter

Symbol	Value	Description	
$\mathcal{A}$	$\{1, \dots, 3\}$	Set of age classes. Each age class corresponds to one of the following age ranges: 0-19, 20-69, and 70++	Assumption
$\mathcal{S}$	$\{u, q, h\}$	Set of infected categories. u: undetected infected individuals; q: quarantine infected individuals with mild symptoms; h: hospitalized infected individuals with moderate or severe symptoms	Assumption
$\mathbf{C}$	$\{Home,\ Work,\ School,\ Other\}$	Set of contact categories.	[122]
$\eta_{s,c},\ s \in \mathcal{S},\ c \in \mathbf{C}$	see Table A.12	Amount of social interaction of the infected category s	Assumption
$\lambda_s, s \in \mathcal{S}$	$0.2\, r_s$	Rates at which an exposed individual becomes an infected individual in sub-class s. It is computed as the inverse of the incubation period time (5 days) multiplied by the ratio (r_s) of new infected in category s (see Table A.11).	[88]
$\beta^c_{i,j},\ i, j \in \mathcal{A},\ c \in \mathbf{C}$		Contact rate between a susceptible of age class i and an infect of class j, considering the contact category c.	[122]
$\iota_i,\ i \in \mathcal{A}$		Age dependent probability of infection given a contact.	Estimated
ρ	0.2	The reciprocal of the mean infectious period (5 days). The time of infectiousness was derived from empirical estimates of the serial interval, the time between successive cases in a chain of transmission.	[109]
$\sigma_i,\ i \in \mathcal{A}$		Death rate for patients with sever symptoms in age class i.	Estimated
$\alpha(t, c),\ c \in \mathbf{C}$	see Table A.13	Governmental action strength at time t	Estimated
k		Intensity of the population response w.r.t the COVID-19 perceived hazard	Estimated
$\theta(i, t),\ i \in \mathcal{A}$		Age and time dependent detection rate with which an undetected infected individual is discovered becoming a quarantine infected individuals	Estimated/Assumption
μ	0.025	Mean duration of public reaction (40 days)	Assumption
ν	0.001826	The reciprocal of the mean immunization period of the recovered individuals, equal to 1.5 years. This is an average value computed considering the immunity values of SARS and MERS	[175]

Table A.10: COVID19 model notation.

Scenario	u	q	h
One-to-one	0.50	0.275	0.225

Table A.11: Ratio r_s of new infected in category s (u, q and h).

Infect category	$Home$	$Work$	$School$	$Other$
u	1	1	1	1
q	0.1	0	0	0
h	0	0	0	0.05

Table A.12: Contact restriction due to the infection.

Date	$Home$	$Work$	$School$	$Other$
February 25$^{\text{th}}$	1	1	0	1
March 8$^{\text{th}}$	1	0.75	0	0.65
March 21$^{\text{st}}$	1	0.4	0	0.3
May 4$^{\text{th}}$	1	0.5	0	0.35
May 18$^{\text{th}}$	1	0.6	0	0.4
June 1$^{\text{st}}$	1	0.9	0	0.5

Table A.13: Governmental actions strength for each contact category characterized by three restrictions (first three rows) and three releases (last three rows).

A.4.2 Contact Matrix

We exploited the Italian synthetic age-specific contact matrix estimated in [122]. In particular, this matrix is defined as a linear combination of four matrices depending on the location of contact, i.e, $Home$, $School$, $Work$, and $Other$. Since our model is defined for three age classes (0-19, 20-69 and 70++), while in [122] the age classes are defined as 5-year bands until age 70 years and a single category aged 75++ (resulting in 16 age categories), we have to scale the rates proportionally to the size and number of the new age classes. In Fig. A.7 the age-specific and location-specific contact matrices are reported. In details, the columns represent the location of contact, while the rows report the seven phases in the simulation window: the usual contact rates, namely *Normal*, the three public restrictions, and the three re-opening phases. The intense of the color indicates higher propensity of making the

contact considering a specific phase.

A.4.3 Further results

Figures A.4 and A.5 show the time evolution of infected individuals and the deaths due to Sars-CoV-2 derived by the model considering the optimal parameter values estimated in the calibration phase. In details, the stacked bar charts in Figure A.4d) shows the proportion of infected individuals in the population. Comparing this surveillance data with those obtained by our calibrated model (i.e, given by the sum of the light blue and blue-gray bars in Fig.A.4) we can clearly point out that a good level of accordance is achieved. Similar conclusion can be derived when we focus on the deaths due to Sars-CoV-2. Indeed Fig.A.5 d) shows certainly that the calibrated model is able to consistently mimic the measured death cases (i.e., red line) too, for both if we look at the aggregated information and if we compare single age classes. Furthermore, Figures A.4 from a) to c) provide a detailed overview on how the infects spread across the different age classes. Figure A.6 shows 5000 different stochastic realization of our mechanistic model with the corresponding median trend and interquartile range. It can be noted that in the initial stages of the pandemic the stochastic behaviour of our model foresees a slightly higher number of infected cases, but starting from March 21st the Surveillance data (red line) lies within the interquaritle range, depicted as a light-green shaded area.

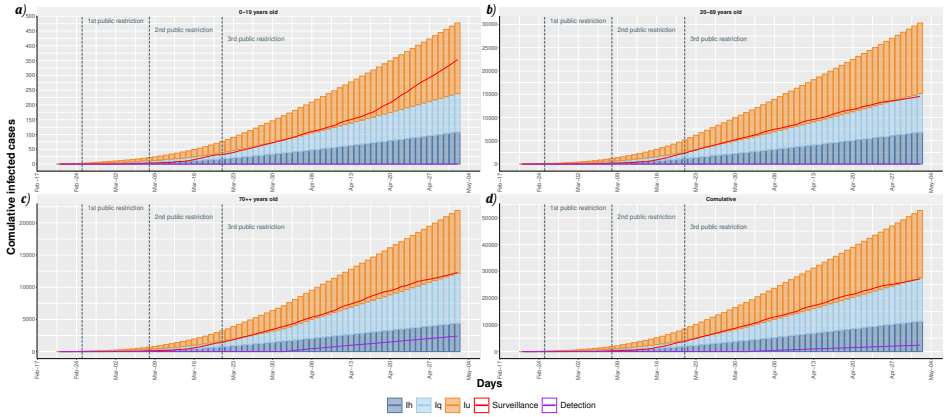


Figure A.4: Model fitting of the surveillance data. Histogram's bars show the number of infected individuals for each age class (a) Age class $0-19$; b) Age class $20-69$; c) Age class $70++$; d) Overall population) with severe symptoms I_h, under quarantine restriction in their houses I_q and undetected infects I_u. The red curve represents the surveillance data, which does not account for undetected cases.

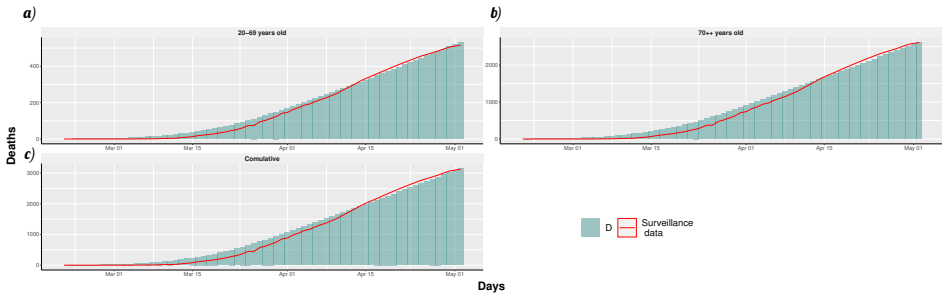


Figure A.5: Model fitting of the surveillance data considering the number of deaths for each age class (a) Age class $20-69$; b) Age class $70++$; c) Overall population). The age class $0-19$ is not showed because no deaths are reported in Piedmont region. The red curve represents the surveillance data, which does not account for undetected cases.

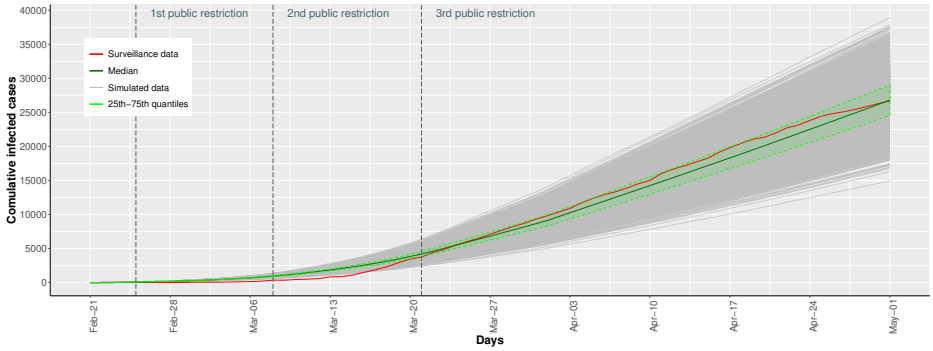


Figure A.6: Stochastic simulations of the one-to-one ratio scenario. 5000 stochastic traces spanning the time period from February 21$^{\text{th}}$ to May 1$^{\text{st}}$. The light-green shaded area shows the range containing traces within the 25$^{\text{th}}$ to 75$^{\text{th}}$ quantiles. The dark-green line is the median of all the traces and the red line is the surveillance. Vertical dashed lines mark the division between two time epoch.

Normal

	Home			Other			Work			School		
	0-19	20-70	>70	0-19	20-70	>70	0-19	20-70	>70	0-19	20-70	>70
0-19	1.94626091	1.85788621	0.05751326	3.59022198	2.65505839	0.19290783	0.01111344	0.0120437	6.56E-06	5.18483237	0.68061173	0.0001179
20-70	0.66185773	2.150229	0.06691802	0.68828975	6.14448455	0.36851142	0.21203735	3.74408731	2.60E-05	1.22937062	0.36937897	0.00087144
>70	0.43331945	1.24103351	0.45030494	0.14779459	4.41545631	0.44487167	2.76E-06	0.00022143	1.24E-05	1.90E-64	4.74E-66	6.75E-92

First restriction

	Home			Other			Work			School		
	0-19	20-70	>70	0-19	20-70	>70	0-19	20-70	>70	0-19	20-70	>70
0-19	1.94626091	1.85788621	0.05751326	3.59022198	2.65505839	0.19290783	0.00111134	0.00120437	6.56E-07	0	0	0
20-70	0.66185773	2.150229	0.06691802	0.68828975	6.14448455	0.36851142	0.19083362	3.36967858	2.34E-05	0	0	0
>70	0.43331945	1.24103351	0.45030494	0.14779459	4.41545631	0.44487167	2.48E-06	0.00019929	1.11E-05	0	0	0

Second restriction

	Home			Other			Work			School		
	0-19	20-70	>70	0-19	20-70	>70	0-19	20-70	>70	0-19	20-70	>70
0-19	1.94626091	1.85788621	0.05751326	2.33364429	1.72578795	0.12539009	0.00833508	0.00903278	4.92E-06	0	0	0
20-70	0.66185773	2.150229	0.06691802	0.44738834	3.99391496	0.23953242	0.15902801	2.80806548	1.95E-05	0	0	0
>70	0.43331945	1.24103351	0.45030494	0.09606649	2.8700466	0.28916658	2.07E-06	0.00016607	9.29E-06	0	0	0

Third restriction

	Home			Other			Work			School		
	0-19	20-70	>70	0-19	20-70	>70	0-19	20-70	>70	0-19	20-70	>70
0-19	1.94626091	1.85788621	0.05751326	1.07706659	0.79651752	0.05787235	0.00444537	0.00481748	2.62E-06	0	0	0
20-70	0.66185773	2.150229	0.06691802	0.20648692	1.84334537	0.11055342	0.08481494	1.49763492	1.04E-05	0	0	0
>70	0.43331945	1.24103351	0.45030494	0.04433838	1.32463689	0.1334615	1.10E-06	8.86E-05	4.95E-06	0	0	0

First reopening phase

	Home			Other			Work			School		
	0-19	20-70	>70	0-19	20-70	>70	0-19	20-70	>70	0-19	20-70	>70
0-19	1.94626091	1.85788621	0.05751326	1.25657769	0.92927044	0.06751774	0.00555672	0.00602185	3.28E-06	0	0	0
20-70	0.66185773	2.150229	0.06691802	0.24090141	2.15056959	0.128979	0.10601868	1.87204365	1.30E-05	0	0	0
>70	0.43331945	1.24103351	0.45030494	0.05172811	1.54540971	0.15570508	1.38E-06	0.00011072	6.19E-06	0	0	0

Second reopening phase

	Home			Other			Work			School		
	0-19	20-70	>70	0-19	20-70	>70	0-19	20-70	>70	0-19	20-70	>70
0-19	1.94626091	1.85788621	0.05751326	1.43608879	1.06202336	0.07716313	0.00666806	0.00722622	3.94E-06	0	0	0
20-70	0.66185773	2.150229	0.06691802	0.2753159	2.45779382	0.14740457	0.12722241	2.24645238	1.56E-05	0	0	0
>70	0.43331945	1.24103351	0.45030494	0.05911784	1.76618252	0.17794867	1.65E-06	0.00013286	7.43E-06	0	0	0

Third reopening phase

	Home			Other			Work			School		
	0-19	20-70	>70	0-19	20-70	>70	0-19	20-70	>70	0-19	20-70	>70
0-19	1.94626091	1.85788621	0.05751326	1.79511099	1.32752919	0.09645391	0.01000209	0.01083933	5.91E-06	0	0	0
20-70	0.66185773	2.150229	0.06691802	0.34414487	3.07224228	0.18425571	0.19083362	3.36967858	2.34E-05	0	0	0
>70	0.43331945	1.24103351	0.45030494	0.0738973	2.20772816	0.22243583	2.48E-06	0.00019929	1.11E-05	0	0	0

Figure A.7: The age-specific and location-specific (columns) contact matrices are reported for each phase (rows) in the simulation window. The intense of the color indicates higher propensity of making the contact.

Abbreviations

$\widehat{ODE}$ Symbolic ODE. 24–27, 29, 65, 69, 72, 75–78, 80, 81, 158–160, 175

AIC Akaike Information Criterion. 122

aP acellular Pertussis. 111

BBB Blood-Brain Barrier. 133, 135, 161–163, 166, 167, 170, 175

BDF Backward Differentiation Formula. 88, 106, 121

CDF Cumulative Distribution Function. 96, 97

CK Chapman-Kolmogorov. 12, 13

CME Chemical Master Equation. 12–14, 31, 32

CNS Central Nervous System. 134, 135, 156, 161–163, 167, 170, 175, 176, 222, 223, 226

CSF cerebrospinal fluid. 225–227

CTMC Continuous Time Markov Chain. 5, 9, 12–14, 17, 23, 35, 36

DAC Daclizumab. 134–139, 144, 147–149, 152, 154, 155, 157–163, 167, 170–176, 216, 218, 220, 222

DFBA Dynamic Flux Balance Analysis. 38, 42, 43, 87

DOA Dynamic Optimization Approach. 42

EBV Epstein-Barr Virus. 135–142, 144, 145, 147, 148, 152, 153, 156–161, 167, 216–220

ECDC European Centre for Disease Prevention and Control. 111

ECDF Empirical Cumulative Distribution Function. 97, 129–131

ESPN Extended Stochastic Petri Net. 5, 6, 45–49, 54–56, 81–83, 86, 88–96, 98, 99, 102, 103, 105, 195–197

ESSN Extended Stochastic Symmetric Net. 5, 6, 47–49, 54–57, 60, 79, 80, 92, 102, 103, 105, 112, 114, 117, 118, 133, 136, 155, 156, 162, 164, 173, 174, 176, 181, 183, 195–197, 220–222

FBA Flux Balance Analysis. 3, 6, 38, 40–43, 56, 81–84, 86–92, 196

FOI force of infection. 181, 184

FVA Flux Variability Analysis. 41

HD Healthy Donor. 165–167, 225, 227

HERVs Human Endogenous RetroViruses. 161

HHV-6 Human Herpes Virus 6. 161

IFNγ Interferon gamma. 133, 135, 161–163, 165, 167, 223, 224, 226

IL-10 Interleukin-10. 133, 161, 163, 167, 224, 226

IL-17 Interleukin-17. 133, 135, 161–163, 165, 167, 224, 226

IL-2 Interleukin-2. 136–140, 154, 157, 171, 175, 218

IL-2R Interleukin-2 receptor. 135, 170

LHS Latin Hypercube Sampling. 141, 142, 144

LPP Linear Programming Problem. 40–43, 82–84, 86–89

MA Mass Action. 13, 14, 18, 45, 46, 48, 52, 56, 60, 62, 70, 71, 74, 76, 79, 81, 82, 84, 88, 92–96, 99, 113, 138, 195, 216, 220

MCS Monte Carlo Sampling. 106

MS Multiple Sclerosis. 6, 103, 108, 133–136, 139–141, 150, 156, 161, 162, 165–168, 171–176, 225–227

NK Natural Killer. 136–140, 147, 157, 158, 161–163, 171, 175, 222

ODC Oligodentrocytes. 135, 137–145, 147–150, 157, 158, 160, 161, 163, 165–167, 170, 171, 174, 175, 224

ODE Ordinary Differential Equation. 5, 6, 9, 13, 14, 20, 23–26, 32, 36, 38, 43, 46, 47, 49, 51, 56, 57, 62, 64, 69, 71, 81, 82, 84, 86–89, 91, 95, 105, 106, 118, 119, 121, 132, 140, 155, 158–160, 165, 175, 185, 196

OP Optimization Problem. 107

PDA Pancreatic Ductal Adenocarcinoma. 88, 90

PDE Partial Differential Equation. 111

PDF Probability Density Function. 96, 97

PN Petri Net. 4–6, 9–12, 15, 32, 37, 45, 48, 89, 103, 105, 195

PRCC Partial Rank Correlation Coefficient. 106, 107, 119, 121, 140, 141

RBP Reproducible Bioinformatics Project. 104, 109

REA Reaction Essentiality Analysis. 41

RG Reachability Graph. 11, 12, 23

RRMS Relapsing Remitting Multiple Sclerosis. 133–138, 144, 147, 149, 151, 155, 156, 158, 161, 162, 164, 167, 173, 174, 221, 222

RS Reachability Set. 11, 13

SEIRS Susceptible-Exposed-Infected-Recovered-Susceptible. 176, 178–180, 193

SIR Susceptible-Infected-Recovered. 12, 21, 22, 29, 45, 47, 48

SIRS Susceptible-Infected-Recovered-Susceptible. 57, 62, 77, 78, 112

SM Symbolic Marking. 21–24, 80

SOA Static Optimization Approach. 42

SPN Stochastic Petri Net. 9, 12–14, 18, 20, 32, 35, 36, 45, 51, 52, 54, 105

SRG Symbolic Reachability Graph. 23

SSA Stochastic Simulation Algorithm. 5, 12, 13, 31–35, 92, 94, 98, 99, 106, 134, 166, 196

SSN Stochastic Symmetric Net. 9, 15–23, 27–29, 32, 45, 48, 52–54, 56, 57, 59, 60, 65–67, 69–71, 74, 76–78, 105, 196

Teff T Effector. 137–142, 145, 149–155, 157, 158, 161–163, 167, 170, 171, 175, 216–220, 222–225

Treg T Regulatory. 135–141, 145, 149–154, 157, 158, 161, 163, 166, 167, 170, 171, 175, 217, 218, 220, 224, 225

VZV Varicella Zoster Virus. 161

wP whole-cell Pertussis. 119

Bibliography

[1] S. I. Ahmed, K. Aziz, A. Gul, S. S. Samar, and S. B. Bareeqa. Risk of multiple sclerosis in epstein–barr virus infection. *Cureus*, 11(9), 2019.

[2] T. Aittokallio and B. Schwikowski. Graph-based methods for analysing networks in cell biology. *Briefings in bioinformatic*, 7, 2006.

[3] H. Akaike. Information theory and an extension of the maximum likelihood principle. In *Selected papers of hirotugu akaike*, pages 199–213. Springer, 1998.

[4] A. Al-Mawali, A. D. Pinto, R. Al-Busaidi, R. H Al-Lawati, and M. Morsi. Comprehensive haematological indices reference intervals for a healthy omani population: First comprehensive study in gulf cooperation council (gcc) and middle eastern countries based on age, gender and abo blood group comparison. *PloS one*, 13(4):e0194497, 2018.

[5] E. G. Amparore, G. Balbo, M. Beccuti, S. Donatelli, and G. Franceschinis. 30 years of GreatSPN. In *Principles of Performance and Reliability Modeling and Evaluation*, pages 227–254. Springer, 2016.

[6] J. Babar, M. Beccuti, S. Donatelli, and A. S. Miner. GreatSPN enhanced with decision diagram data structures. In *Application and Theory of Petri Nets.PETRI NETS 2010*, volume 6128 of *LNCS*, pages 308–317. 2010.

[7] H. H Balfour Jr, C. J Holman, K. M Hokanson, M. M Lelonek, J. E Giesbrecht, D. R White, D. O Schmeling, C. Webb, W. Cavert, D. H Wang, and et al. A prospective clinical study of epstein-barr virus and host interactions during acute infectious mononucleosis. *Journal of Infectious Diseases*, 192(9), 2005.

[8] A. Bar-Or, M. P Pender, R. Khanna, L. Steinman, H.-P. Hartung, T. Maniar, Ed Croze, B. T Aftab, G. Giovannoni, and M. A Joshi. Epstein–barr virus in multiple sclerosis: theory and emerging immunotherapies. *Trends in molecular medicine*, 26(3):296–310, 2020.

[9] M. Beccuti, E. Bibbona, A. Horvath, R. Sirovich, A. Angius, and G. Balbo. Analysis of petri net models through stochastic differential equations. In *International Conference on Applications and Theory of Petri Nets and Concurrency*, pages 273–293. Springer, 2014.

[10] M. Beccuti, C. Fornari, G. Franceschinis, S.M. Halawani, O. Ba-Rukab, A.R. Ahmad, and G. Balbo. From symmetric nets to differential equations exploiting model symmetries. *Computer Journal*, 58(1):23–39, 2015.

[11] M. Beccuti, L. Capra, M. De Pierro, G. Franceschinis, and S. Pernice. Deriving symbolic ordinary differential equations from stochastic symmetric nets without unfolding. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11178 LNCS:30–45, 2018. doi: 10.1007/978-3-030-02227-3_3.

[12] M. Beccuti, L. Capra, M. De Pierro, G. Franceschinis, L. Follia, and S. Pernice. A tool for the automatic derivation of symbolic ode from symmetric net models. volume 2019-October, pages 36–48, 2019. doi: 10.1109/MASCOTS.2019.00015.

[13] D. Besozzi. Reaction-based models of biochemical networks. In A. Beckmann, L. Bienvenu, and N. Jonoska, editors, *Pursuit of the Universal*, pages 24–34, Cham, 2016. Springer International Publishing. ISBN 978-3-319-40189-8.

[14] L. R Bisset, T. L Lung, M. Kaelin, E. Ludwig, and R.W Dubs. Reference values for peripheral blood lymphocyte phenotypes applicable to the healthy adult population in switzerland. *European journal of haematology*, 72(3):203–212, 2004.

[15] J. C Blackwood, D. AT Cummings, H. Broutin, S. Iamsirithaworn, and P. Rohani. Deciphering the impacts of vaccination and immunity on

pertussis epidemiology in thailand. *Proceedings of the National Academy of Sciences*, 110(23):9595–9600, 2013.

[16] A. Bobbio, M. Gribaudo, and M. Telek. Analysis of large scale interacting systems by mean field method. In *2008 Fifth International Conference on Quantitative Evaluation of Systems*, pages 215–224, 2008. doi: 10.1109/QEST.2008.47.

[17] A. Bobbio, D. Cerotti, M. Gribaudo, M. Iacono, and D. Manini. *Markovian Agent Models: A Dynamic Population of Interdependent Markovian Agents*, pages 185–203. Springer International Publishing, Cham, 2016. ISBN 978-3-319-33786-9. doi: 10.1007/978-3-319-33786-9_13. URL `https://doi.org/10.1007/978-3-319-33786-9_13`.

[18] A. Burden, R. Burden, and J. Faires. *Numerical Analysis, 10th ed.* 01 2016. ISBN 1305253663. doi: 10.13140/2.1.4830.2406.

[19] Patricia Therese Campbell, James M. McCaw, and Jodie McVernon. Pertussis models to inform vaccine policy. In *Human vaccines & immunotherapeutics*, 2015.

[20] Y. Cao and L. Petzold. Accuracy limitations and the measurement of errors in the stochastic simulation of chemically reacting systems. *Journal of Computational Physics*, 212(1):6–24, 2006.

[21] Y. Cao and D. C Samuels. Discrete stochastic simulation methods for chemically reacting systems. *Methods in enzymology*, 454:115–140, 2009.

[22] Y. Cao, H. Li, and L. Petzold. Efficient formulation of the stochastic simulation algorithm for chemically reacting systems. *The journal of chemical physics*, 121(9):4059–4067, 2004.

[23] Y.g Cao, D. Gillespie, and L. Petzold. Avoiding negative populations in explicit poisson tau-leaping. *The Journal of chemical physics*, 123(5): 054104, 2005.

[24] L. Capra, M. De Pierro, and G. Franceschinis. A high level language for structural relations in well-formed nets. In G. Ciardo and P. Darondeau, editors, *Applications and Theory of Petri Nets 2005*, pages 168–187, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg. ISBN 978-3-540-31559-9.

[25] L. Capra, M. De Pierro, and G. Franceschinis. Computing structural properties of symmetric nets. In *Proceedings of the 15th International Conference on Quantitative Evaluation of Systems (QEST 2015)*, QEST '15, Madrid, ES, 2015. IEEE Computer Society.

[26] F. Carinci. Covid-19: Preparedness, Decentralisation, and the Hunt for Patient Zero. *British Medical Journal*, 368:1–2, 2020.

[27] P. Castagno, S. Pernice, G. Ghetti, M. Povero, L. Pradelli, D. Paolotti, G. Balbo, M. Sereno, and M. Beccuti. A computational framework for modeling and studying pertussis epidemiology and vaccination. *BMC bioinformatics*, 21:344, 2020. doi: 10.1186/s12859-020-03648-6. URL `https://www.scopus.com/inward/record.uri?eid=2-s2.0-85091129070&doi=10.1186%2fs12859-020-03648-6&partnerID=40&md5=4170c151da6019f5324230e4d0553f2f`.

[28] P. Cazzaniga, C. Damiani, D. Besozzi, R. Colombo, S.M. Nobile, D. Gaglio, D. Pescini, S. Molinari, G. Mauri, L. Alberghina, and M. Vanoni. Computational strategies for a system-level understanding of metabolism. *Metabolites*, 4(4):1034–1087, 2014.

[29] S.P. Chapman, C. M. Paget, G. N. Johnson, and J.-M. Schwartz. Flux balance analysis reveals acetate metabolism modulates cyclic electron flow and alternative glycolytic pathways in chlamydomonas reinhardtii. *Frontiers in plant science*, 6:474, 2015.

[30] G. Chiola, C. Dutheillet, G. Franceschinis, and S. Haddad. Stochastic well-formed coloured nets for symmetric modelling applications. *IEEE Tran. Comput.*, 42(11):1343–1360, 1993.

[31] J. Choi, S. J. Lee, Y. A Lee, H.G. Maeng, J.K. Lee, and Y. W. Kang. Reference values for peripheral blood lymphocyte subsets in a healthy korean population. *Immune network*, 14(6):289–295, 2014.

[32] J. Colom and M. Silva. Convex geometry and semiflows in p/t nets. a comparative study of algorithms for computation of minimal p-semiflows. pages 79–112, 06 1989.

[33] A Compston and A Coles. Multiple sclerosis. lancet (lond, engl) 372: 1502–1517, 2008.

[34] F. Cordero, M. Beccuti, C. Fornari, S. Lanzardo, L. Conti, F. Cavallo, G. Balbo, and R. Calogero. Multi-level model for the investigation of oncoantigen-driven vaccination effect. *BMC Bioinformatics*, 14 (SUPPL6), 2013.

[35] C. Cosner, D. L DeAngelis, J. S Ault, and D. B Olson. Effects of spatial grouping on the functional response of predators. *Theoretical population biology*, 56(1):65–75, 1999.

[36] M. W Covert and B. Ø Palsson. Transcriptional regulation in constraints-based metabolic models of escherichia coli. *Journal of Biological Chemistry*, 277(31):28058–28064, 2002.

[37] M.W. Covert, N. Xiao, T.J. Chen, and J.R. Karr. Integrating metabolic, transcriptional regulatory and signal transduction models in escherichia coli. *Bioinformatics*, 24(18):2044–2050, 2008.

[38] F. da Veiga Leprevost, B. A Grüning, S. Alves Aflitos, H. L Röst, J. Uszkoreit, H. Barsnes, M. Vaudel, P. Moreno, L. Gatto, J. Weber, M. Bai, R. C Jimenez, T. Sachsenberg, J. Pfeuffer, R. Vera Alvarez, J. Griss, A. I Nesvizhskii, and Y. Perez-Riverol. BioContainers: an open-source and community-driven framework for software standardization. *Bioinformatics*, 33(16):2580–2582, 03 2017. ISSN 1367-4803.

[39] B. W Dale, L. G Adams, and R T. Bowyer. Functional response of wolves preying on barren-ground caribou in a multiple-prey ecosystem. *Journal of Animal Ecology*, pages 644–652, 1994.

[40] M. D. de Cellès, F.MG Magpantay, A. A King, and P. Rohani. The impact of past vaccination coverage and immunity on pertussis resurgence. *Science translational medicine*, 10(434), 2018.

[41] F. Deisenhammer, H. Zetterberg, B. Fitzner, and U. K Zettl. The cerebrospinal fluid in multiple sclerosis. *Frontiers in immunology*, 10:726, 2019.

[42] C.A Dendrou, L. Fugger, and M. A Friese. Immunopathology of multiple sclerosis. *Nature Reviews Immunology*, 15(9):545–558, 2015.

[43] Italy Department of Civil Protection. Covid-19 italia. http://opendatadpc.maps.arcgis.com/apps/opsdashboard/index.html#/b0c68bce2cce478eaac82fe38d4138b1. Accessed March 23, 2020.

[44] C. Di Pietrantonj. Epidemia COVID-19 regione piemonte: situazione al 16.04.2020 ore 10:00. https://www.seremi.it/, 2020. Version: 2020-04-16.

[45] C. Di Pietrantonj. Epidemia COVID-19 regione piemonte: situazione al 26.04.2020 ore 10:00. https://www.seremi.it/, 2020. Version: 2020-04-26.

[46] C. Di Pietrantonj and et al. Epidemia COVID-19 regione piemonte: Report del 17 marzo 2020. https://www.seremi.it/, 2020. Version: 2020-03-17.

[47] C. Di Pietrantonj and et al. Epidemia COVID-19 regione piemonte: Report del 22 marzo 2020. https://www.seremi.it/, 2020. Version: 2020-03-22.

[48] L. Durelli, L. Conti, M. Clerico, D. Boselli, G. Contessa, P. Ripellino, B. Ferrero, P. Eid, and F. Novelli. T-helper 17 cells expand in multiple sclerosis and are inhibited by interferon-β. *Annals of Neurology: Official Journal of the American Neurological Association and the Child Neurology Society*, 65(5):499–509, 2009.

[49] R. Dutta and B. D Trapp. Mechanisms of Neuronal Dysfunction and Degeneration in Multiple Sclerosis. *Prog. Neurobiol.*, 93(1):1–12, jan 2011. ISSN 0301-0082. URL http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3030928/.

[50] P. Érdi and J. Tóth. *Mathematical models of chemical reactions: theory and applications of deterministic and stochastic models*. Manchester University Press, 1989.

[51] European Centre for Disease Prevention and Control. Pertussis - annual epidemiological report for 2017. Technical report, 2018. URL https://ecdc.europa.eu/sites/portal/files/documents/AER_for_2017-pertussis.pdf.

[52] W. Feller. *An Introduction to Probability Theory and Its Applications*, volume 1. Wiley, January 1968. ISBN 0471257087.

[53] NM Ferguson, D Laydon, G Nedjati-Gilani, N Imai, K Ainslie, M Baguelin, S Bhatia, A Boonyasiri, Z Cucunubá, G Cuomo-Dannenburg, A Dighe, I Dorigatti, and et al. Impact of non-pharmaceutical interventions (NPIs) to reduce COVID-19 mortality and healthcare demand. *Imperial College COVID-19 Response Team*, pages 1–20, 2020.

[54] G. Franceschinis, L. Capra, and M. De Pierro. A tool for symbolic manipulation of arc functions in symmetric net models. ICST, 1 2014. doi: 10.4108/icst.valuetools.2013.254407.

[55] T. Gharibi, Z. Babaloo, A. Hosseini, F. Marofi, A. Ebrahimi-kalan, S. Jahandideh, and B. Baradaran. The role of b cells in the immunopathogenesis of multiple sclerosis. *Immunology*, 2020.

[56] M. A Gibson and J. Bruck. Efficient exact stochastic simulation of chemical systems with many species and many channels. *The journal of physical chemistry A*, 104(9):1876–1889, 2000.

[57] D. Gillespie. Exact stochastic simulation of coupled chemical reactions. *The journal of physical chemistry*, 81(25):2340–2361, 1977.

[58] D. Gillespie. Approximate accelerated stochastic simulation of chemically reacting systems. *The Journal of Chemical Physics*, 115(4):1716–1733, 2001.

[59] D. Gillespie and L. Petzold. Improved leap-size selection for accelerated stochastic simulation. *The Journal of Chemical Physics*, 119(16):8229–8234, 2003.

[60] G Giordano, F Blanchini, R Bruno, and et al. Modelling the COVID-19 epidemic and implementation of population-wide interventions in Italy. *Nature Medicine*, pages 1–6, 2020.

[61] R. Gold, E. Radue, G. Giovannoni, K. Selmaj, E. K. Havrdova, X. Montalban, D. Stefoski, T. Sprenger, R. R Robinson, and et al. Long-term safety and efficacy of daclizumab beta in relapsing–remitting multiple

sclerosis: 6-year results from the selected open-label extension study. *Journal of Neurology*, 2020.

[62] M V Gonfiantini, E Carloni, F Gesualdo, E Pandolfi, E Agricola, E Rizzuto, S Iannazzo, M L Ciofi Degli Atti, A Villani, and A E Tozzi. Epidemiology of pertussis in italy: Disease trends over the last century. *Eurosurveillance*, 19(40): 20921, 2014. doi: https://doi.org/10.2807/1560-7917.ES2014.19.40.20921. URL `https://www.eurosurveillance.org/content/10.2807/1560-7917.ES2014.19.40.20921`.

[63] M. Gribaudo and A. Horváth. Modeling hybrid positive systems with hybrid petri nets. In *Positive Systems*, pages 71–78. Springer, 2003.

[64] M. Gribaudo, A. Horváth, A. Bobbio, E. Tronci, E. Ciancamerla, and M. Minichino. Fluid petri nets and hybrid model-checking: A comparative case study. *Reliability Engineering & System Safety*, 81(3):239–257, 2003.

[65] M. Gribaudo, M. Iacono, and D. Manini. Covid-19 spatial diffusion: A markovian agent-based model. *Mathematics*, 9(5):485, 2021.

[66] E. L Haseltine and J. B Rawlings. Approximate simulation of coupled fast and slow reactions for stochastic chemical kinetics. *The Journal of chemical physics*, 117(15):6959–6969, 2002.

[67] L. Heirendt and et al. Creation and analysis of biochemical constraint-based models: the cobra toolbox v3.0. *Nature protocols*, 2018.

[68] M. Herajy, L. Fei, C. Rohr, and M. Heiner. Coloured hybrid petri nets: An adaptable modelling approach for multi-scale biological networks. *Computational Biology and Chemistry*, 76:87–100, 2018.

[69] H. W Hethcote. The mathematics of infectious diseases. *SIAM review*, 42(4):599–653, 2000.

[70] H. W Hethcote, P. Horby, and P. McIntyre. Using computer simulations to compare pertussis vaccination strategies in australia. *Vaccine*, 22(17-18):2181–2191, 2004.

[71] R. Hofestädt. A Petri net application of metabolic processes. *Journal of System Analysis, Modeling and Simulation*, 16:113–122, 1994.

[72] CS Holling. The components of predation as revealed by a study of small-mammal predation of the european pine sawfly. 1959.

[73] I Holmdahl and C Buckee. Wrong but Useful - What Covid-19 Epidemiologic Models Can and Cannot Tell Us. *New England Journal of Medicine*, 10:1–7, 2020.

[74] S. A Juliano. Nonlinear curve fitting: predation and functional response curves. *Design and analysis of ecological experiments*, 2:178–196, 2001.

[75] M. K. Molloy. Performance analysis using stochastic Petri nets. *IEEE Transaction on Computers*, 31(9):913–917, September 1982.

[76] H. Kebir, K. Kreymborg, I. Ifergan, A. Dodelet-Devillers, R. Cayrol, M. Bernard, F. Giuliani, N. Arbour, B. Becher, and A. Prat. Human th 17 lymphocytes promote blood-brain barrier disruption and central nervous system inflammation. *Nature medicine*, 13(10):1173–1175, 2007.

[77] R.J Keizer, A. DR Huitema, J. HM Schellens, and J. H Beijnen. Clinical pharmacokinetics of therapeutic monoclonal antibodies. *Clinical pharmacokinetics*, 49(8):493–507, 2010.

[78] B. Kendall, C. Briggs, W. Murdoch, P. Turchin, S. Ellner, E. McCauley, R. Nisbet, and S. Wood. Why do populations cycle? a synthesis of statistical and mechanistic modeling approaches. *Ecology*, 80(6):1789–1805, 1999.

[79] A. P Kim and D. E Baker. Daclizumab. *Hospital pharmacy*, 51(11): 928–939, 2016.

[80] SM Kissler, C Tedijanto, E Goldstein, YH Grad, and M. Lipsitch. Projecting the transmission dynamics of SARS-CoV-2 through the postpandemic period. *Science*, pages 1–9, 2020.

[81] N. P Klein, J. Bartlett, A. Rowhani-Rahbar, B. Fireman, and R. Baxter. Waning protection after fifth dose of acellular pertussis vaccine in children. *New England Journal of Medicine*, 367(11):1012–1019, 2012.

[82] N. Kulkarni, L. Alessandrí, R. Panero, M. Arigoni, M. Olivero, G. Ferrero, F. Cordero, M. Beccuti, and R.A. Calogero. Reproducible bioinformatics project: A community for reproducible bioinformatics analysis pipelines. *BMC Bioinformatics*, 19, 2018. doi: 10.1186/s12859-018-2296-x.

[83] T. Kurtz. Solutions of ordinary differential equations as limits of pure jump Markov processes. *J. Appl. Probab.*, 1(7):49–58, 1970.

[84] J. L. Peterson. *Petri Net Theory and the Modeling of Systems*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1981. ISBN 0136619835.

[85] K. Lange. *Optimization*. Springer, New York, NY, second edition, 2013.

[86] C. L Langrish, Y. Chen, W. M Blumenschein, J. Mattson, B. Basham, J.D Sedgwick, T. McClanahan, R.A Kastelein, and D. J Cua. Il-23 drives a pathogenic t cell population that induces autoimmune inflammation. *The Journal of experimental medicine*, 201(2):233–240, 2005.

[87] A. Laroni and A. Uccelli. Cd56bright natural killer cells: A possible biomarker of different treatments in multiple sclerosis. *Journal of Clinical Medicine*, 9(5):1450, 2020.

[88] S. A Lauer, K. H Grantz, Q. Bi, F. K Jones, Qulu Z., H. R Meredith, A. S Azman, N. G Reich, and J. Lessler. The Incubation Period of Coronavirus Disease 2019 (COVID-19) From Publicly Reported Confirmed Cases: Estimation and Application. *Annals of Internal Medicine*, 2020.

[89] E. Lavezzo, E. Franchin, C. Ciavarella, G. Cuomo-Dannenburg, L. Barzon, C. Del Vecchio, L. Rossi, R. Manganelli, A. Loregian, N. Navarin, I. Dorigatti, and et al. Suppression of covid-19 outbreak in the municipality of vo', italy. *MedRxiv*, 2020.

[90] J. E Libbey, L. L McCoy, and R.S Fujinami. Molecular mimicry in multiple sclerosis. *International review of neurobiology*, 79:127–147, 2007.

[91] Q. Lin, S. Zhao, D. Gao, Y. Lou, S. Yang, S. S Musa, M. H Wang, Y. Cai, W. Wang, L. Yang, and et al. A conceptual model for the outbreak of Coronavirus disease 2019 (COVID-19) in Wuhan, China with individual reaction and governmental action. *International Journal of Infectious Diseases*, 93:211–216, 2020.

[92] F. Liu, M. Heiner, and M. Yang. An efficient method for unfolding colored Petri nets. In *Winter Simulation Conference*, pages 1–12, Berlin, Germany, Dec. 2012. IEEE Computer Society.

[93] A. J Lotka. *Analytical theory of biological populations*. Springer Science & Business Media, 1998.

[94] FMG Magpantay, M D. de Cellès, P Rohani, and AA King. Pertussis immunity and epidemiology: mode and duration of vaccine-induced immunity. *Parasitology*, 143(7):835–849, 2016.

[95] R Mahadevan and CH Schilling. The effects of alternate optimal solutions in constraint-based genome-scale metabolic models. *Metabolic engineering*, 5(4):264–276, 2003.

[96] R. Mahadevan, J. S Edwards, and F. J Doyle III. Dynamic flux balance analysis of diauxic growth in escherichia coli. *Biophysical journal*, 83(3): 1331–1340, 2002.

[97] D.C. Maranas and A.R. Zomorrodi. *Optimization Methods in Metabolic Networks*. Wiley, 2016. ISBN 1119028493.

[98] S. Marino, I.B. Hogue, C.J. Ray, and D. E. Kirschner. A methodology for performing global uncertainty and sensitivity analysis in systems biology. *Journal of Theoretical Biology*, 254(1):178 – 196, 2008. ISSN 0022-5193. URL `http://www.sciencedirect.com/science/article/pii/S0022519308001896`.

[99] A. Marsan, G. Balbo, G. Conte, S. Donatelli, and G. Franceschinis. *Modelling with Generalized Stochastic Petri Nets*. J. Wiley, New York, NY, USA, 1995.

[100] P. McCombe. The short and long-term effects of pregnancy on multiple sclerosis and experimental autoimmune encephalomyelitis. *Journal of clinical medicine*, 7(12):494, 2018.

[101] L. Michaelis and M. L. Menten. *Die kinetik der invertinwirkung*. Universitätsbibliothek Johann Christian Senckenberg, 2007.

[102] J. Min Lee, E. P. Gianchandani, J.A. Eddy, and J.A. Papin. Dynamic analysis of integrated signaling, metabolic, and regulatory networks. *PLOS Computational Biology*, 4(5):1–20, 05 2008.

[103] Ministero della Salute. Tavola storica 4.15: Casi denunciati di alcune malattie soggette a denuncia obbligatoria, anni 1925-2009. http://seriestoriche.istat.it/fileadmin/documenti/Tavola_4.15.xls, last accessed on 2020-05-08.

[104] L. K Misegades, K. Winter, K. Harriman, J. Talarico, N. E Messonnier, T. A Clark, and S. W Martin. Association of childhood pertussis with receipt of 5 doses of pertussis vaccine by time since last vaccine dose, california, 2010. *Jama*, 308(20):2126–2132, 2012.

[105] G. Moirano, L. Richiardi, C. Novara, and M. Maule. Approaches to daily monitoring of the SARS-CoV-2 outbreak in Northern Italy. *Fronteirs Public Health*, in press:1–10, 2020.

[106] X. Montalban, R. Gold, A. J Thompson, S. Otero-Romero, M. P. Amato, D. Chandraratna, M. Clanet, G. Comi, T. Derfuss, F. Fazekas, and et al. Ectrims/ean guideline on the pharmacological treatment of people with multiple sclerosis. *Multiple Sclerosis Journal*, 24(2):96–120, 2018.

[107] J. Mossong, N. Hens, M. Jit, P. Beutels, K. Auranen, R. Mikolajczyk, M. Massari, S. Salmaso, G. S. Tomba, J. Wallinga, and et al. Social contacts and mixing patterns relevant to the spread of infectious diseases. *PLOS Medicine*, 5(3):1–1, 03 2008. doi: 10.1371/journal.pmed.0050074. URL https://doi.org/10.1371/journal.pmed.0050074.

[108] Japan National Institute of Infectious Diseases. Field briefing: Diamond princess covid-19 cases. https://www.niid.go.jp/niid/en/2019-ncov-e/9407-covid-dp-fe-01.html. Published: 2020-02-19.

[109] H. Nishiura, N. M Linton, and A. R. Akhmetzhanov. Serial interval of novel coronavirus (2019-nCoV) infections. *medRxiv*, pages 1–6, 2020. doi: 10.1101/2020.02.03.20019497.

[110] J. Oskari Virtanen and S. Jacobson. Viruses and multiple sclerosis. *CNS & Neurological Disorders-Drug Targets (Formerly Current Drug Targets-CNS & Neurological Disorders)*, 11(5):528–544, 2012.

[111] B. Palsson. *Systems Biology: Simulation of Dynamic Network States*. Cambridge University Press, 2011.

[112] E. T. Papoutsakis. Equations and calculations for fermentations of butyric acid bacteria. *Biotechnology and bioengineering*, 26(2):174–187, 1984.

[113] G. R. D. Passos, D. K. Sato, J. Becker, and K. Fujihara. Th17 cells pathways in multiple sclerosis and neuromyelitis optica spectrum disorders: pathophysiological and therapeutic implications. *Mediators of inflammation*, 2016, 2016.

[114] S. Pernice, M. Beccuti, P. Do, M. Pennisi, and F. Pappalardo. Estimating daclizumab effects in multiple sclerosis using stochastic symmetric nets. pages 1393–1400, 2019. doi: 10.1109/BIBM.2018.8621259. URL `https://www.scopus.com/inward/record.uri?eid=2-s2.0-85062527310&doi=10.1109%2fBIBM.2018.8621259&partnerID=40&md5=5e8c050e8714b74ef884213a8671e9a5`.

[115] S. Pernice, L. Follia, G. Balbo, G. Sartini, N. Totis, P. Lió, I. Merelli, F. Cordero, and M. Beccuti. Integrating petri nets and flux balance methods in computational biology models: a methodological and computational practice. *Fundamenta Informaticae*, 2019.

[116] S. Pernice, M. Pennisi, G. Romano, A. Maglione, S. Cutrupi, F. Pappalardo, G. Balbo, M. Beccuti, F. Cordero, and R. A. Calogero. A computational approach based on the colored petri net formalism for studying multiple sclerosis. *BMC bioinformatics*, 2019.

[117] S. Pernice, G. Romano, G. Russo, M. Beccuti, M. Pennisi, and F. Pappalardo. Exploiting stochastic petri net formalism to capture the relapsing remitting multiple sclerosis variability under daclizumab administration. pages 2168–2175, 2019. doi: 10.1109/BIBM47256.2019.8983368.

[118] S. Pernice, M. Beccuti, G. Romano, M. Pennisi, A. Maglione, S. Cutrupi, F. Pappalardo, L. Capra, G. Franceschinis, M. De Pierro, G. Balbo, F. Cordero, and R. Calogero. Multiple sclerosis disease: A computational approach for investigating its drug interactions. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 12313 LNBI:299–308, 2020. doi: 10.1007/978-3-030-63061-4_26.

[119] S. Pernice, P. Castagno, L. Marcotulli, M.M. Maule, L. Richiardi, G. Moirano, M. Sereno, F. Cordero, and M. Beccuti. Impacts of reopening strategies for covid-19 epidemic: a modeling study in piedmont region. *BMC Infectious Diseases*, 20 (1), 2020. doi: 10.1186/s12879-020-05490-w. URL `https://www.scopus.com/inward/record.uri?eid=2-s2.0-85094138913&doi=10.1186%2fs12879-020-05490-w&partnerID=40&md5=d14fa879de47ee676244b12a524d859d`.

[120] S. Pernice, L. Follia, A. Maglione, M. Pennisi, F. Pappalardo, F. Novelli, M. Clerico, M. Beccuti, F. Cordero, and S. Rolla. Computational modeling of the immune response in multiple sclerosis using epimod framework. *BMC bioinformatics*, 21(17):1–20, 2020.

[121] A. Poli, T. Michel, M. Thérésine, E. Andrès, F. Hentges, and J. Zimmer. Cd56bright natural killer (nk) cells: an important nk cell subset. *Immunology*, 126(4):458–465, 2009.

[122] K. Prem, A. R Cook, and M. Jit. Projecting social contact matrices in 152 countries using contact surveys and demographic data. *PLoS computational biology*, 13(9):e1005697, 2017.

[123] K Prem, AR Cook, and M Jit. Projecting social contact matrices in 152 countries using contact surveys and demographic data. *Plos Computational Biology*, 13(9):1–21, 2017.

[124] Presidenza del Consiglio dei Ministri - Dipartimento della Protezione Civile. Italian survelliance data. `https://github.com/pcm-dpc/COVID-19`. Accessed: 2020-03-28.

[125] Md S. Rahman and S. Chakravarty. A predator-prey model with disease in prey. *Nonlinear Analysis: Modelling and Control*, 18(2):191–209, 2013.

[126] V. Reddy, M. Mavrovouniotis, and M. Liebman. Petri net representation in metabolic pathways. In *Proc. Int. Conf. Intelligent Systems for Molecular Biology*, pages 328–336, 1993.

[127] L. F. Richardson. Ix. the approximate arithmetical solution by finite differences of physical problems involving differential equations, with an

application to the stresses in a masonry dam. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 210(459-470):307–357, 1911.

[128] P. Rohani, X. Zhong, and A. A King. Contact network structure explains the changing epidemiology of pertussis. *Science*, 330(6006):982–985, 2010.

[129] S. Rolla, V. Bardina, S. De Mercanti, P. Quaglino, R. De Palma, D. Gned, D. Brusa, L. Durelli, F. Novelli, and M. Clerico. Th22 cells are expanded in multiple sclerosis and are resistant to ifn-β. *Journal of leukocyte biology*, 96(6):1155–1164, 2014.

[130] S. Rolla, A. Maglione, S. F. De Mercanti, and M. Clerico. The meaning of immune reconstitution after alemtuzumab therapy in multiple sclerosis. *Cells*, 9(6):1396, 2020.

[131] K. Roosa, Y. Lee, R. Luo, A. Kirpich, R. Rothenberg, J. M. Hyman, and et al. Real-time forecasts of the COVID-19 epidemic in China from February 5th to February 24th, 2020. *Infect. Dis. Model*, 5:256–263, 2020.

[132] M. Roy and S. D Finley. Computational model predicts the effects of targeting cellular metabolism in pancreatic cancer. *Frontiers in physiology*, 8:217, 2017.

[133] E. Saathoff, P. Schneider, V. Kleinfeldt, S. Geis, D. Haule, L. Maboko, E. Samky, M. Souza, M. Robb, and M. Hoelscher. Laboratory reference values for healthy adults from southern tanzania. *Tropical Medicine & International Health*, 13(5):612–625, 2008.

[134] A. Saltelli, M. Ratto, S. Tarantola, and F. Campolongo. Sensitivity analysis for chemical models. *Chemical Reviews*, 105(7):2811–2828, 2005.

[135] S. Sánchez-Ramón, J. Navarroa, C. Aristimuño, M. Rodríguez-Mahou, J.M. Bellón, E. Fernández-Cruz, and C. de Andrés. Pregnancy-induced expansion of regulatory t-lymphocytes may mediate protection to multiple sclerosis activity. *Immunology letters*, 96(2), 2005.

[136] A. Santagostino, G. Garbaccio, A. Pistorio, V. Bolis, G. Camisasca, P. Pagliaro, and M. Girotto. An italian national multicenter study for

the definition of reference ranges for normal values of peripheral blood lymphocyte subsets in healthy adults. *Haematologica*, 84(6), 1999.

[137] C. H Schilling, J. S Edwards, D. Letscher, and B. Ø Palsson. Combining pathway analysis with flux balance analysis for the comprehensive study of metabolic systems. *Biotechnology and bioengineering*, 71(4):286–306, 2000.

[138] F. Schlögl. Chemical reaction models for non-equilibrium phase transitions. *Zeitschrift für physik*, 253(2):147–161, 1972.

[139] D. Segal, C. Schmitz, and P. R Hof. Spatial distribution and density of oligodendrocytes in the cingulum bundle are unaltered in schizophrenia. *Acta neuropathologica*, 117(4):385, 2009.

[140] S. L Sheridan, R. S Ware, K. Grimwood, and S. B Lambert. Number and order of whole cell pertussis vaccines in infancy and disease protection. *Jama*, 308(5):454–456, 2012.

[141] M. Shirley. Daclizumab: a review in relapsing multiple sclerosis. *Drugs*, 77(4):447–458, 2017.

[142] M. Silva, J. M. Colom, J. M. Campos, and C Gamma. Linear algebraic techniques for the analysis of petri nets. In *In: Recent Advances in Mathematical Theory of Systems, Control, Networks, and Signal Processing II*, pages 35–42. Mita Press, 1992.

[143] M. Soheilypour and M. RK Mofrad. Agent-based modeling in molecular systems biology. *BioEssays*, 40(7):1800020, 2018.

[144] D. K Sojka, Y. Huang, and D. J Fowell. Mechanisms of regulatory t-cell suppression–a diverse arsenal for a moving target. *Immunology*, 124(1): 13–22, 2008.

[145] D. A Somerset, Y. Zheng, M. D Kilby, D. M Sansom, and M. T Drayson. Normal human pregnancy is associated with an elevation in the immune suppressive cd25+ cd4+ regulatory t-cell subset. *Immunology*, 112(1), 2004.

[146] J. Son, C. A Lyssiotis, H. Ying, X. Wang, S. Hua, M. Ligorio, R. M Perera, C. R Ferrone, E. Mullarky, Ng Shyh-Chang, and et al. Glutamine

supports pancreatic cancer growth through a kras-regulated metabolic pathway. *Nature*, 496(7443):101, 2013.

[147] S. Spina, F. Marrazzo, M. Migliari, R. Stucchi, A. Sforza, and R. Fumagalli. The response of Milan's Emergency Medical System to the COVID-19 outbreak in Italy. *The Lancet*, 395:49–50, 2020.

[148] A. J Steelman. Infection as an environmental trigger of multiple sclerosis disease exacerbation. *Frontiers in immunology*, 6:520, 2015.

[149] J. Stelling. Mathematical models in microbial systems biology. *Opinion in Microbiology*, 7(S), 2004.

[150] D. T Gillespie. A rigorous derivation of the chemical master equation. physica a 188, 404-425. *Physica A: Statistical Mechanics and its Applications*, 188:404–425, 09 1992.

[151] Task force COVID-19. Sorveglianza integrata covid-19 in italia. Istituto Superiore di Sanitá - Dipartimento Malattie Infettive e Servizio di Informatica, Version: 2020-04-15, `https://www.epicentro.iss.it/coronavirus/bollettino/Infografica_15aprile%20ITA.pdf`.

[152] N. Totis, L. Follia, C. Riganti, F. Novelli, F. Cordero, and M. Beccuti. Overcoming the lack of kinetic information in biochemical reactions networks. *SIGMETRICS Perform. Eval. Rev.*, 44(4):91–102, May 2017. ISSN 0163-5999.

[153] M Trabucchi and D De Leo. Nursing homes or besieged castles: Covid-19 in northern italy. *The lancet psychiatry*, 5(5):387–388, 2020.

[154] B.D. Trapp and K.-A. Nave. Multiple Sclerosis: An Immune or Neurodegenerative Disorder? *Annu. Rev. Neurosci.*, 2008. ISSN 0147-006X.

[155] J. S Tzartos, M. A Friese, M.J Craner, J. Palace, J. Newcombe, M. M Esiri, and L. Fugger. Interleukin-17 production in central nervous system-infiltrating t cells and glial cells is associated with active disease in multiple sclerosis. *The American journal of pathology*, 172(1): 146–155, 2008.

[156] M Van Boven, HE De Melker, JFP Schellekens, and M Kretzschmar. A model based evaluation of the 1996–7 pertussis epidemic in the netherlands. *Epidemiology & Infection*, 127(1):73–85, 2001.

[157] P. Van den Driessche and J. Watmough. Reproduction numbers and sub-threshold endemic equilibria for compartmental models of disease transmission. *Mathematical biosciences*, 180(1-2):29–48, 2002.

[158] A. Varma and B.O. Palsson. Stoichiometric flux balance models quantitatively predict growth and metabolic by-product secretion in wild-type escherichia coli w3110. *Applied and environmental microbiology*, 60(10): 3724–3731, 1994.

[159] R. Verity, L. C Okell, I. Dorigatti, P. Winskill, C. Whittaker, N. Imai, G. Cuomo-Dannenburg, H. Thompson, P. GT Walker, H. Fu, and et al. Estimates of the severity of coronavirus disease 2019: a model-based analysis. *The Lancet infectious diseases*, 20(6):669–677, 2020.

[160] S. Vijayakumar, M. Conway, P. Lió, and C. Angione. Optimization of multi-omic genome-scale models: Methodologies, hands-on tutorial, and perspectives. In *Methods in Molecular Biology*, pages 389–408. Springer New York, dec 2017. doi: 10.1007/978-1-4939-7528-0_18.

[161] S. Vijayakumar, M. Conway, P. Lió, and C. Angione. Seeing the wood for the trees: a forest of methods for optimization and omic-network integration in metabolic modelling. *Briefings in Bioinformatics*, may 2017. doi: 10.1093/bib/bbx053.

[162] J. Virtanen. Viruses and Multiple Sclerosis. *CNS Neurol Disord Drug Targets*, pages 528–544, aug 2012. URL `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4758194/`.

[163] M Vollmer and et al. Using mobility to estimate the transmission intensity of COVID-19 in Italy: A subnational analysis with future scenarios. *Imperial College COVID-19 Response Team*, pages 1–17, 2020.

[164] M. W. Covert, C. H. Schilling, and B Palsson. Regulation of gene expression in flux balance models of metabolism. *Journal of Theoretical Biology*, 213(1):73 – 88, 2001. ISSN 0022-5193.

[165] M. Warny, J. Helby, B. G. Nordestgaard, H. Birgens, and S. E. Bojesen. Lymphopenia and risk of infection and infection-related death in 98,344 individuals from a prospective danish population-based study. *PLoS medicine*, 15(11):e1002685, 2018.

[166] M. R. Watson. Metabolic maps for the Apple II. *Biochemical Society Transactions*, 12(6):1093–1094, 12 1984. ISSN 0300-5127. doi: 10.1042/bst0121093. URL https://doi.org/10.1042/bst0121093.

[167] H. J Wearing and P. Rohani. Estimating the duration of pertussis immunity using epidemiological signatures. *PLoS pathogens*, 5(10):e1000647, 2009.

[168] Website. Ministero della salute. resident population on 1st january in piedmont. http://dati.istat.it/Index.aspx?QueryId=18540, 2020.

[169] Website. Atlas ecdc. italian statistics. https://atlas.ecdc.europa.eu/public/index.aspx, last accessed on 2020-05-08.

[170] Website. Ministero della salute. bollettino nazionale delle notifiche delle malattie infettive dal 1996. http://www.salute.gov.it/portale/documentazione/p6_2_8_1_1.jsp?lingua=italiano&id=3, last accessed on 2020-05-08.

[171] Website. Ministero della salute. coperture vaccinali. http://www.salute.gov.it/portale/documentazione/p6_2_8_3_1.jsp?lingua=italiano&id=20, last accessed on 2020-05-08.

[172] Website. Ministero della salute. statistics about the italian birth and death rate. http://dati.istat.it, last accessed on 2020-05-08.

[173] A. M Wendelboe, A. Van Rie, S. Salmaso, and J. A Englund. Duration of immunity against pertussis after natural infection or vaccination. *The Pediatric infectious disease journal*, 24(5):S58–S61, 2005.

[174] H. Wickham. *ggplot2: elegant graphics for data analysis*. Springer, New York, NY, USA, 2016.

[175] L Wu, N Wang, Y Chang, and et al. Duration of antibody responses after severe acute respiratory syndrome. *Emerging Infectious Diseases*, 13(10):1562–1564, 2007.

[176] Y Xiang, S. Gubian, B. Suomela, and J. Hoeng. Generalized simulated annealing for efficient global optimization: the GenSA package for R. *The R Journal*, 2012. URL http://journal.r-project.org/. Forthcoming.

[177] B. I. Yamout and R. Alroughani. Multiple Sclerosis. *Semin Neurol*, 38 (2):212–225, 04 2018.

[178] L. Yang, A. Ebrahim, C. J Lloyd, M. A Saunders, and B. O Palsson. Dynamicme: Dynamic simulation and refinement of integrated models of metabolism and protein expression. *BMC systems biology*, 13(1):2, 2019.

[179] A.L Zozulya and H. Wiendl. The role of regulatory t cells in multiple sclerosis. *Nature clinical practice Neurology*, 4(7):384–398, 2008.