PhD Dissertation

International Doctorate School in Information and Communication Technologies

DIT - University of Trento

Knowledge discovery for stochastic models of biological systems

Michele Forlin

Advisor:

Prof. Corrado Priami

Università degli Studi di Trento

December 2010

Abstract

Biology is the science of life and living organisms. Empowered by the deployment of several automated experimental frameworks, this discipline has seen a tremendous growth during the last decades. Recently, the focus towards studying biological systems holistically, has lead to biology converging with other disciplines. In particular, computer science is playing an increasingly important role in biology, because of its ability to disentangle complex system level issues. This increasing interplay between computer science and biology has lead to great progress in both fields and to the opening of new important areas for research. In this thesis we present methods and approaches to tackle the problem of knowledge discovery in computational biology from a stochastic perspective. Major bottlenecks in adopting a stochastic representation can be overcome with the use of proper methodologies by integrating statistics and computer science. In particular we focus on parameter inference for stochastic models and efficient model analysis. We show the application of these approaches on real biological case studies aiming at inferring new knowledge even when a priori (and/or experimental) information is limited.

Keywords

[systems biology, knowledge discovery, evolutionary inference,
model analysis]

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The importance of computer science is witnessed by its ability to change our life, our social relations and how we relate with the environment, as well as by its potential to affect other disciplines. In particular, developments in computer science have had a great influence on biological sciences by helping scientists understand the basic principle and dynamics of living organisms.

1.1 The Problem

Biology is the science of life and living organisms. This discipline has seen a tremendous growth during the last 20 years, empowered by the deployment of automated experimental frameworks. The increase in computational power pushed the research community to investigate more accurately, at an unprecedented finer level of granularity, how living systems behave.

New approaches to the interpretation of living systems have been proposed. In particular systems biology [36] focuses on the study of living systems as a whole, rather than following a reductionist approach that concentrates on on the action of individual subsystems. In particular, computer science is playing an increasingly important role in biology, because of its ability to disentangle complex system level issues. This increasing interplay between computer science and biology has lead to great progress in both fields and to the opening of new important areas for research.

Hitherto, developments in computer science have produced new tailored algorithms, often in association with statistical procedures, allowing us to manage and analyze the enormous amount of data that is produced in modern, high-throughput biological experiments. In its current form, bio-informatics is mainly concerned with the analysis of experimentally produced data, that is, it tries to identify patterns or infer knowledge just from a priori selected experiments.

More recently, computer science has started to play a different role, more closely linked to biological research activities. The unique algorithmic view that computer science can give [53] produces new insights on the dynamical behavior of biological systems, by offering new languages for properly modeling these complex systems.

Modeling is an essential step for fully understanding the dynamics of biological systems. Although models can be used to give a static picture of the whole system, they may also easily include the dynamical information required to study its evolution over time.

Historically, deterministic mathematical modeling has been considered the way to describe the dynamics of biological systems. Now, stochastic approaches [72] are gaining interest due to their ability to quantitatively describe in a more realistic manner the observed behavior of living systems. Among others, formal languages (e.g. Process Algebras [2]), have been designed to describe concurrent systems. Recently, they been extended to allow the abstraction of intracellular chemical reactions [54].These formalisms are called models and they are composed by a qualitative and a quantitative component.

From a purely descriptive point of view, these approaches have shown their potential, but still a considerable amount of work is required in order to automatically derive knowledge from their dynamical behavior.

A common problem in knowledge inference for biological systems is the one of parameter inference, also known as *model calibration*. Here, the goal is to properly infer model parameters in order to reproduce some experimentally observed behavior at the best.

Inference represents a challenging task. Usually the available experimental information is not sufficient to describe the entire system. This is due to several reasons: experiments are costly and time-consuming, sometimes important parameters governing cannot be measured directly and, common to every experimental activity, the available data contain noise, deriving from both experimental limitations and intrinsic biological stochasticity. Hence, a valid inference framework should be able to face with these limitations.

To derive new knowledge from a computational approach to biological systems, it is not sufficient to have an accurate model, in depth analysis is also required. At this stage we are interested in exploring the dynamical behavior of the model. Usually the focus is on those situations that are critical for biological laboratories, for technical or economical reasons.

As has been pointed out, only very small stochastic models can be analyzed analytically with tools like Ito calculus, and thus, to study their dynamic properties, we rely on stochastic simulations. A stochastic simulation of a model represents a random realization of the underlying stochastic process. Consequently, simulation runs differ from one another often making it hard to derive predictions about the system behavior. Then, in order to obtain statistically reliable conclusions, there is the need of performing a usually large number of stochastic simulations and to aggregate their outputs. Due to the complexity of certain systems that are modelled, fast and efficient methods are required to tackle the computational demands of repeated simulations.

1.2 The Contribution

The main focus of this thesis is on algorithmic systems biology, and in particular on methods and approaches to tackle the problems of knowledge discovery.

Major bottlenecks in adopting a stochastic representation can be overcome with the proper use of methodologies that incorporate statistics into computer science approaches. We followed two directions: inference, and in particular parameter inference, and efficient model analysis.

We developed a new inference scheme that uses stochastic modeling and simulation not only as a tool for describing a system, but also as an effective method for inferring biological knowledge. Within this view, we used evolutionary computational techniques to evolve stochastic models of biological systems towards solutions that match experimental data, that is, to iterate the modeling and simulation steps in an intelligent manner in order to use them as an inference tool.

This evolutionary inference framework can help us overcome the limitations of the experimental data. In fact it easily deals with problems where experimental information is sparse or even incomplete. The algorithm is powerful enough to sometime even produce hypothesis and predictions for parts of the system where experimental data was missing.

Concerning the model analysis, we focus our attention on two wel- known problems. The first is related to the analysis of model properties. Biologically relevant events can be formally characterised as temporal logic formulae that can then be automatically checked against a discrete state model. In particular we present a approximate methodology that, given a model M, a property ϕ and a desired level of statistical confidence, estimates the probability that ϕ is satisfied by M with the estimated measure meeting the desired confidence.

The second development for model analysis is related to the sensitivity of model parameters. Usually, sensitivity analysis for model parameters tries to identify those parameters which have the greatest influence on the model behavior by perturbing them (usually with small perturbations). Within this setting, but taking a broader view of the problem, we developed a workflow-schema to extend the concept of sensitivity as to efficiently generate and detect peculiar behaviors of a given model by appropriately perturbing its parameter space.

For both approaches we coupled statistical methods with high performance parallel algorithms in order to properly face with the computational demands of stochastic simulations.

In order to evaluate the proposed approaches and methodologies we considered several case studies. Every method is presented and consequently explained and tested on a particular problem. The evolutionary inference framework has been tested on a well known benchmarking problem, the thermal isomerization of α-pinene. Analysis of model properties has been tested on a cell cycle model, while the approach to identify and estimate the parameter effects on a desired model output has been tested on a predator-prey model to evaluate the relationship among parameters and oscillation frequencies.

The use of test cases let the reader fully understand the basic principles and functioning of the methodologies. Nonetheless, our aim is to provide approaches and methods able to face with real problems. Real cases are much more difficult to tackle than simulated or benchmarking ones as they result from real laboratory activities with all the constraints and limitations they have to deal with. Valid methods should be able to bypass possible obstacles and still give reliable information.

We will present two real cases derived from direct collaboration with research institutes. The former is related to the description of V. vinifera general pathway for phenolics biosynthesis leading to flavonoids, in collaboration with Instituto Agrario San Michele all'Adige (IASMA), located in Trento, Italy. This problem is challenging mainly due to the limited amount of experimental data. If, from one hand the methodologies are able to work with partial information, some assumptions still has to be considered.

The second case study is related to the development of Leishmaniasis disease, in collaboration with the Universidad La Laguna, Tenerife, Spain. In this problem the experimental information is difficult to handle due to its variability and experimental constraints, but we were able to exploit system level properties present in the experimental system to apply our theoretical methods, showing promising results.

Chapter 2

Knowledge discovery for biological systems

Recent years have witnessed unprecedent increases in the number, variety and complexity of resources available to life science researchers. An important reason lies in the general trend of moving from single molecular processes to complete cellular pathways. This shift in perspective requires the integration into system-level views of the elementary pieces of information that have been gathered by thousands research groups so far.

In order for modelling to serve 'wet' science, models have to make empirically testable predictions that can be validated. In other words, by making the process of hypothesis generation more simple, systems biology aim to promote a more rigorous analysis of the system under study.

To build such a system-level description, the starting point is the identity of the components constituting the biological system as well as their interactions and dynamical behavior. Once the current knowledge of the system is incorporated into the model, the model can then be used to provide new insights and predictions for conditions of the systems that have not previously been explored.

More generally we can think at this knowledge discovery problem as that process that enables to obtain new information and insights about a biological system. It is depicted in Figure 2.1.

From the picture we can clearly identify two separate routes for the knowledge discovery process: an inference cycle, summarizing the need for a valid and consistent model with current knowledge and a following analysis process.

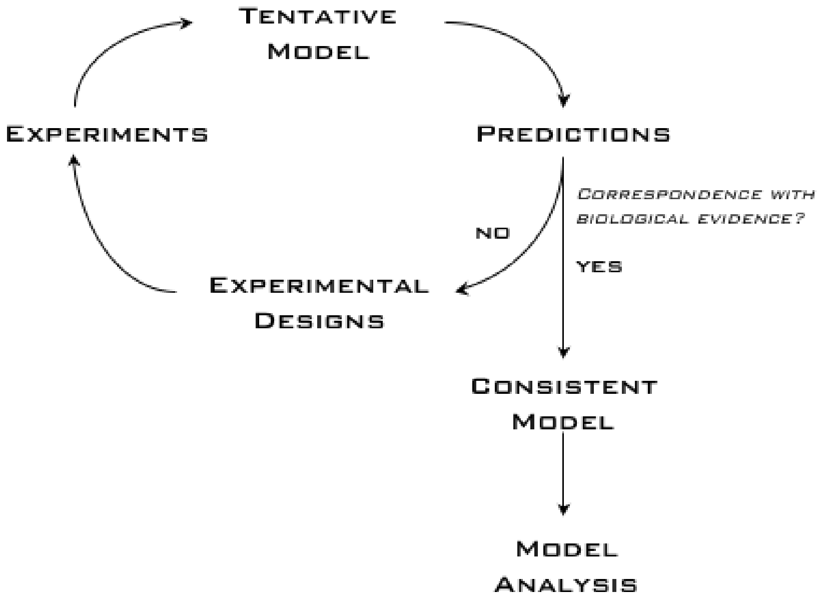


Figure 2.1: Knowledge discovery process

The goal of the inference cycle is to derive the information required to build a valid model of the biological system under study. Usually, the available knowledge about a system is only partial. One may know which are the components of the systems under study and how they interact, but not the parameters that govern their dynamical behavior. This problem, which was briefly alluded to in the introduction, is also known as parameter inference or model calibration. Another common problem when building a model from limited data, is that a full knowledge of the interactions between the components is lacking - in this case, parameter inference is not sufficient, and we have to rely on a different approach known as network inference.

In this next case study, we limit ourself to the study of parameter estimation.

Parameter estimation is a challenging task. Common situations when dealing with inference is the frequent lack of information and the limited available experimental data due cost and time reasons. Sometimes some characteristics of the system are not measurable and still, common to every experimental activity, data contain noise, deriving from both experimental conditions and intrinsic biological stochasticity.

Parameter inference for stochastic models has to be considered an iterative process. This is because usually it is not sufficient to perform a single iteration for defining a valid model. At every step, a tentative model is produced, and this model is scored based on how well it reproduces experimental data. At each step, the model that best reproduces the experiments is chosen, and this process is iterated until a model that satisfies the chosen constraints reached.

Once a valid model has been reached, the attention has to move towards model analysis. Only the full understanding of the model dynamics can let researchers to use it as a proper counterpart of real experimentation in order to derive new knowledge. In fact at this stage, the focus is on those situations that are critical for biological laboratories, for technical or economical resaons.

The ability to derive valid and efficient methods for model analysis stands in identifying the most informative aspects of the biological system in order to use the model as a perfect computational counterpart of wet-ab experimentation.

It is thus clear that the entire process of knowledge discovery is strongly influenced by both experimental choices and the simulation method. In particular, the quantity and quality of the experimental data well as the type of modeling and simulation considered will both affect the inference phase and the model analysis phase.

2.1 Theory and experimentation

Developments in experimentation has always been coupled with advances in technological devices. In the second half of the nineteen century, extensive experimentation took root in many disciplines. Among others Biology greatly benefited from this approach.

Early attempts to analyze biology at a systems level were subject to inadequate experimental activities due to several technological limitations. Only recently, with the development of high throughput technology we are starting to have the required variety of information needed to a system-level analysis of biological systems.

If technology represents a great issue in the development of experiments, a very hotly debated philosophical question arises when we analyze the role of theories into experimentation [55].

Two ways of thinking can be identified. The first sees the growth of theoretical knowledge from experimental activities considering experimentation as theory-free, while the second questions the role of theories into experiments pointing that experimentation is theory dependent and every experiment is planned and performed following some precise theoretical perspectives.

Both approaches often occur in scientific research depending upon the type of problem. Usually, most innovative or borderline researches don't have the support of fully fleshed out theories and thus experimentation has the specific task of deriving new knowledge, hopefully leading to the creation of new theories. On the contrary, problems where the focus is on specific system behavior (e.g. under peculiar conditions, mutations, etc) usually rely on a predefined theoretical knowledge. Experiments in these cases are then planned and performed with the aim of confirming or refusing the hypotheses derived from theoretical knowledge.

Besides the philosophical questions about the interplay of theories and experimentation, statistics also play a fundamental role in shaping scientific knowledge. Statistical methods for the design of experiments represent the essential complementation to properly derive trustable conclusions. Such methods should always be considered during the design of an experiment, both in the case where the experiment is aiming to prove or disprove a specific scientific question, and in the case where the aim of the experiment is to obtain new data about a subject where no relevant hypothesis exist.

A sound experimental design is a mandatory step for proper extraction of relevant information from the system under investigation.

The use of proper statistical methodology for experimentation has the advantage of appropriately dealing with the noise or intrinsic variations a system may exhibit. The reproducibility of an experiment is central. It is well known that a single experimental trial is not enough to establish a stable result and, for this reason, replicas are usually necessary and statistical aggregation techniques are used to measures are used to describe the experimental outcome.

2.2 Biological modeling

The most common interpretation a biologist can give to the term 'model' is the one of *graphical description of a mechanism underlying cellular process* [21].

More generally, the modeling activity has the goal to describe a system at a high level, sometimes reducing its complexity through the introduction of educated abstractions and simplifications, and it is mostly knowledge driven. In fact, what modelers do, is to try to summarize the knowledge about a system in a (hopefully) simple manner.

Anyway, any modeling effort should start with the definition a purpose and with the identification of the appropriate level of detail. A very detail model is inadequate for a general and qualitative inquiry and would possible lead to misleading conclusions. Even for the best studied systems the accumulated data often isn't sufficient to describe the variety of the elementary processes that occur. Consequently assumptions are always necessary.

Two major approaches to modeling are used in biology. The most popular one is the deterministic approach. This formalism uses set of Ordinary Differential Equations (ODEs) or Partial Differential Equations (PDEs) to describe the evolution of the system in time and/or space. Each equation represents the rate of change of a species concentration in the system. A derivation of the standard ODE models is the power-law formalism [61] in which the process that integrate biochemical networks are modeled using power-law expansions in the variable of the system.

By definition, a deterministic description of a biological system generates a behaviour that is completely determined by the input parameters and structure of the model. The same input will produce the same output if the model is simulated multiple times.

This deterministic approach is best suited to model and describe the behavior of systems where species are abundant and thus reactions occur frequently. In these cases species concentration are well approximated by continuous processes.

Complementary to the deterministic approach is the stochastic one. With the stochastic approach possible transformations determining the evolution of biological systems are described probabilistically. The use of various types of specification languages (such as process algebras, chemical reactions, membrane systems or Petri nets, master equations), allow encoding in a highly expressive and user-friendly way the stochastic process that dynamically reproduces the evolution of the modeled biological system over time.

With this approach every simulation results in a different realization of the stochastic process. This type of description is instead well suited when the system is composed by species that are scarce and thus the effect of every reaction event may greatly influence the overall system's behavior.

It is now generally accepted that stochastic models are necessary to properly capture the multiple sources of heterogeneity needed for modeling biosystems in a realistic way [72]. However, such models are computationally more demanding than deterministic ones, and considerably more difficult to fit to experimental data.

An intermediate approach is represented by hybrid methods. These methods include at the same time continuous representation for modeling fast reactions and stochastic representation to take into account the effect of slow reactions.

Among the different types of modeling we focused on concurrency-theory-derived methods [18] inherited from computer science, because of their capabilities in describing biological systems with an algorithmic view.

2.2.1 BlenX modeling language

BlenX is a process calculi derived programming language [19, 18] and it is specifically designed for modeling entities that can change their behavior in response to external stimuli.

A general biological molecule M with n interaction sites is depicted as a box B_M (figure 2.2. The program P_M is called process and allows to describe the behavior of B_M. In particular, P_M activates proper replies to

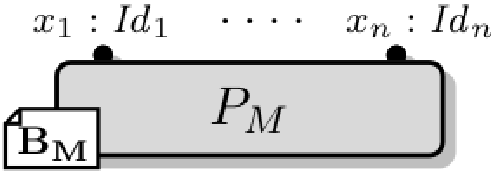


Figure 2.2: BlenX representation of a general molecule

external signals caught by interaction sites $x_i : Id_i$. Types Id_i discriminate among allowed and disallowed interactions, mimicking interaction mechanisms based on compatibility [52]. The name x_i is used by the process P_M to modify or to interact through the associated type Id_i. Process P_M is written in a process calculi style and therefore it has few primitives inspired by both π-calculus [46] and molecular biology [42].
A BlenX system consists in a set of boxes that, running in parallel, can interact and can be attached together through their interfaces forming complexes. The dynamics of a BlenX system emerges from the way in which boxes interact and change and is described in terms of an operational semantics.
A complete description of the BlenX language, followed by "on the road" explanations through examples can be found in [19]. For the sake of clarity we report here some basics on programming BlenX models with events.
Events specify transitions that are not elementary reactions. Here we consider two classes of events, namely join and split. The join event 2.1

$$when(B1, B2 :: rateParameter1)join(B) \tag{2.1}$$

is enabled when a box $B1$ and a box $B2$ are available. The left part in brackets of the event is called condition, while $join(B)$ is called verb. Event 2.1 removes boxes $B1$ and $B2$ and adds a box B. The duration of the transition is specified by the rateParameter1 value, as usual, the unique

parameter of a negative exponential distribution.
The split event in reported in 2.2

$$when(B :: rateParameter2)split(B1, B2) \tag{2.2}$$

Event 2.2 reverses event 2.1 and removes box B and adds both $B1$ and $B2$. Events add flexibility to BlenX enabling the description of biological systems with different levels of detail in the same model. Note that the BlenX language offers a richer set of primitives that the ones presented here; for a detailed description of the full language we still refer the reader to [19].

2.3 Simulation of biological models

Biological models describe the structure and dynamics of the system, but they still just give a static picture of the system behavior. For a model, to be useful, it is essential that all its relevant behavior and properties can be determined in a practical way: analytically, numerically, or by deriving the model with certain (typically random) inputs and observing the corresponding outputs. The latter process is called simulation.
Biological models, being deterministic or stochastic are often analytically not tractable. Their complexity, in terms of components and interactions, make the use of simulations mandatory for the study of the dynamical behavior.
Simulations are clearly model dependent. Deterministic simulations are the obvious result of a deterministic model. In this case the dynamics of the system is fully determined by the initial conditions and the model structure. In the discrete-stochastic setting, biochemical species are enumerable quantities representing the number of molecules of a given substance, and the evolution of the system is probabilistic, rather than deterministic, leading to Continuous Time Markov Chain (CTMC). Events occur discretely

after a random time period, with the chosen reaction and timestep both depending only on the previous state.

Within such a stochastic setting a common algorithm is used to generate exact realizations (runs) of the underlying Markov process. It is known as the Stochastic Simulation Algorithm (SSA) or more commonly, the Gillespie algorithm.

2.3.1 Stochastic Simulation Algorithm

In the stochastic realm the amount of a molecule i at time t is modeled as a discrete random variable $X_i(t)$, and its system of belonging is expressed by a vector $\mathbf{X}$ of random variables. The *Chemical Master Equation* (CME) [29] gives a description of the time evolution of a biological system in terms of a joint probability distribution $\mathcal{P}(X,t)$. In particular, the CME specifies the probability that, at time t, the system holds X_1 molecules of the first species of the vector X, X_2 molecules of the second species, and so on. Due to its complexity, $\mathcal{P}(X,t)$ the CME is often intractable both analytically and numerically, requiring researchers to resort to stochastic simulation in order to characterize its dynamics.

Several stochastic simulation algorithms are available [40], but most of them derive from the Stochastic Simulation Algorithm (SSA) [28]. SSA considers a well-stirred mix of molecular species that chemically interact through reaction channels inside some fixed volume and at a constant temperature. Based on CME, a propensity function is defined for each reaction j, which is used to calculate the probability that a reaction j will occur in the next infinitesimal interval. The algorithm then relies on standard Monte Carlo methods to stochastically select and execute a reaction ,and by iterating the process, a simulated trajectory in the discrete state-space of $\mathcal{P}(X,t)$. The simulation consists of four main steps: (i) Initialize the data structures of the system; (ii) Randomly select a reaction; (iii) Execute the selected reaction; (iv) Update the data structures.

The machinery of SSA founds on the definition of the *propensity function* of a reaction j: the likelihood that a reaction j will fire in the next infinitesimal interval is a function of the number of molecules involved in the reaction j and of a constant number specific to j, named *specific probability rate constant*. For example, let us consider the reaction: $A + B \xrightarrow{c} C + D$. Its propensity function is $c \times |A| \times |B|$, where $| \cdot |$ represents the number of molecules of A and B. Different implementations of SSA exist that use ad-hoc algorithms and data structures to improve the processes of storing and updating the propensity functions.

Chapter 3

State of the Art

Systems biology deeply relies on methodologies for knowledge discovery. Many research groups are working in this field as the enormous amount of data and intrinsic complexity of biological systems, pose several challenges. Research directions follow the entire knowledge discovery cycle described in figure 2.1. Most of the available methods and tools for both parameter inference and model analysis have been created and used for deterministic models of biological systems, while only recently works have concentrated on stochastic settings.

We will now present the current state of the art of parameter inference and model analysis.

3.1 Parameter inference

Recent literature in inferring rate coefficients of biochemical reactions reports many different methods. In a problem of parameter estimation (also known as model calibration), given a set of experimental data, the goal is to calibrate model parameters in order to reproduce the experimentally observed behavior at the best.

Technically, a parameter estimation problem is stated as the minimization of a cost function that measures the goodness of fit of the model with respect to experimental data. It represents a nonlinear programming problem and very often is multimodal (i.e. presents multiple local optima). Thus, traditional gradient-based methods may fail to identify the global solution and may converge to a local optimum.

Numerous methods have been developed for parameter estimation and they can be distinguished by the type of cost function used or by the global optimization technique implemented. Cost functions that have shown to work well in practice are: *bayesian estimators*, *maximum likelihood estimators* and *least squares estimators*. The main difference between the three methods is represented by the amount of information they require: bayesian estimators require the probability distribution of the parameters and the conditional probability distribution of the measurement for some given parameter values and maximum likelihood methods capture a substantial amount of knowledge in the definition of the likelihood function to be then maximized, whereas least squares estimators can be performed without any extrinsic information.

Beside cost functions, a great attention has been devoted to the implementation of new global optimization techniques, usually divided into deterministic and stochastic ones. Moles et al. in [47] analyzed several deterministic and stochastic global optimization methods. Results on a benchmark case study showed the better performance of stochastic methods, and pointed out the effectiveness of evolutionary strategies applied to the parameters estimation problem for continuous models.

More recent literature reports several new methods. Rodriguez and coworkers [58] presented a meta-heuristic procedure derived from operational research, showing its performances on three different biological examples modeled through ODE's. Deterministic approaches to the global optimization problem using branch-and-bound principles to identify the best set of model parameters has been presented by Polisetty et al. [51]. Chou et al. [12] developed the alternate regression (AR) method. The key feature of AR is that it dissects the nonlinear inverse problem of estimating parameter values into iterative steps of linear regression.

Recent developments by Rodrigez-Fernandez and coworkers [59] have been focusing on hybrid methods, i.e. methods that combine global and local optimization having a number of desirable features: robustness of global optimization and rapid convergence of local method in the proximity of the optimum. Improvements in hybrid methods have been recently presented by Balsa-Canto et al [5]. They used an evolutionary strategy as a global method together with a local multiple-shooting approach. Only recently there have been some attempts to deal with the noise present in stochastic systems by developing methods based on simulated maximum likelihood (Tian, [68]) or based on a probabilistic, generative model of the variations in reactant concentration (Lecca, [39]). In both papers the authors clearly state the necessity of developing new methods able to deal with the noise of the experimental data and with the fluctuations that stochastic systems exihibit.

Effective methods for statistically estimating stochastic models by using time course data have only recently appeared in the systems biology literature. Reinker and coworkers [56] presented a method that tries to approximate the likelihood function, while another approach is to use computationally intensive Monte Carlo methods to estimate it [68]. Finally bayesian approaches [7] have been presented to develop exact bayesian inference, but still algorithms are computationally intensive and do not scale well realistic sized problems.

In general, all the presented methods for model calibration assume, as a starting point, the availability of experimental data about the time evolution of all the species involved in the model. In practice, in most circumstances, the direct measurements of some species may not be practicable both for economical and experimental reasons, resulting thus in incomplete data. Inference in systems where experimental data are not complete currently represents an open and challenging problem.

3.2 Model analysis

For what concerns model analysis, lots of methods have been developed for deterministic models. Only few are instead available for stochastic models. This is mostly due to the challenges that such a representation brings.
When dealing with techniques for the verification of temporal logic property against probabilistic/stochastic models, we can identify either exact or approximate methods.
Exact approaches work by constructing a complete representation of a finite state space model and, because of this, their application to complex systems is unfeasible. PRISM [37] and MRMC [34] are two popular probabilistic model checking tools that support both exact and approximated CSL verification. Approximated verification can be one of two different types. If the considered problem is to establishing whether the likelihood p of a formula is $p \trianglelefteq b$ where $b \in [0, 1]$ is a threshold and $\trianglelefteq \{<, \leq, \geq, >\}$ (i.e. model checking problem) then the outcome of verification is boolean and is determined based on Hypothesis testing. On the other hand if the problem is one of determining an estimate for p then this is achieved through confidence interval based techniques. PRISM approximated verification belong to the latter type: the size of the sample is determined statically as a function of the chosen level of confidence and the desired approximation, rather than being calculated iteratively as function of intermediate estimates, as is the case with our method, whereas paths generation is controlled by on-the-fly checking of the considered formula. Furthermore although PRISM has been recently added with support for (exact) probabilistic LTL model

checking, at the best of our knowledge, it currently supports statistical verification only for CSL (and not for LTL). The YMER [74] and MRMC tool, on other hand, features approximated (hypothesis testing based) model checking which uses on-the-fly verification of sampled path in order to decide whether the probability of formula is below/above a threshold. The Monte Carlo Model Checker MC2(PLTLc) [23] computes a point estimate of a Probabilistic LTL logic (with numerical constraints) formula to hold of model. MC2(PLTLc) does not include any simulation engine but works offline by taking a set of sampled trajectories generated by any simulation or ODE solver software. Besides MC2(PLTLc) calculates also the probabilistic domain of satisfaction for any free variable of PLTLc formula. Finally the APMC tool [31] features confidence interval based estimates of the probability of Probabilistic LTL and PCTL formulae to hold of either DTMC and CTMC models.

The analysis of the parameters space of biological models have been previously studied in the context of uncertainty analysis and, in particular, sensitivity analysis. Sensitivity analysis provides a series of approaches and tools to investigate how model parameters affect system output. Usually these approaches apply on deterministic models. A general review of uncertainty and sensitivity analysis methods can be found in [60]. Recently, Marino et al. [44] have compared and then proposed novel techniques to perform sensitivity analysis in a stochastic settings. Still within the stochastic setting, an extension of sensitivity analysis for the study of bistable stochastic models has been proposed by Degasperi and Gilmore [16]. These methods have been also included in computational tools. Hoare and coworkers [32] realized SaSat, a matlab toolbox for sampling and sensitivity analysis, containing the most important sampling and analysis methods. It can be used as a black box and thus applies also in the biological context, even if it is not specific. Instead, SimLab [27] is a standalone software that performs uncertainty and sensitivity analysis.

Chapter 4

Approaches

The development of new approaches always takes as its starting point the specific needs of the community. We have seen that in computational systems biology, the use of stochastic thinking allow us to accurately describe the variability at the individual level typical of biological systems. At the same time, however, it poses remarkable challenges in handling and analyzing models as well as to interface these models with the experimental evidence.

The first contribution is an evolutionary inference framework able to connect stochastic models and experimental data in order to derive systems kinetics. We will then present methods to efficiently analyze stochastic models making extensive use of statistical methodologies coupled with parallel computation.

4.1 Evolutionary parameter inference

We developed a new inference scheme which uses stochastic modeling and simulation not only as a tool for describing a system, but also as an effective method for inferring biological knowledge. Within this view, we used evolutionary computation techniques to evolve stochastic models of biological systems, evaluated through simulation, towards solutions that

match experimental data, that is, to iterate the modeling and simulation steps in an intelligent manner in order to use them as an inference tool.
The inference framework is composed by different parts that work in a coordinate manner. As depicted in Figure 4.1, the flow of information starts with some initial proposed solution (an initial random guess or derived from the experimental data) for the problem (1). It is then coded to form a model and a stochastic simulation is performed (2). By doing so, we obtain a first simulated evolution for the system, that is compared with the experimental information in order to evaluate its goodness (3). If the goodness is satisfactory the process ends with a proposed solution for the inference problem (4), while, if the results are not satisfactory the evolutionary algorithm evolves the solutions according to its dynamics (5) producing thus new solutions and models (6). The process is iterated until some goodness conditions or other convergence criteria are met.

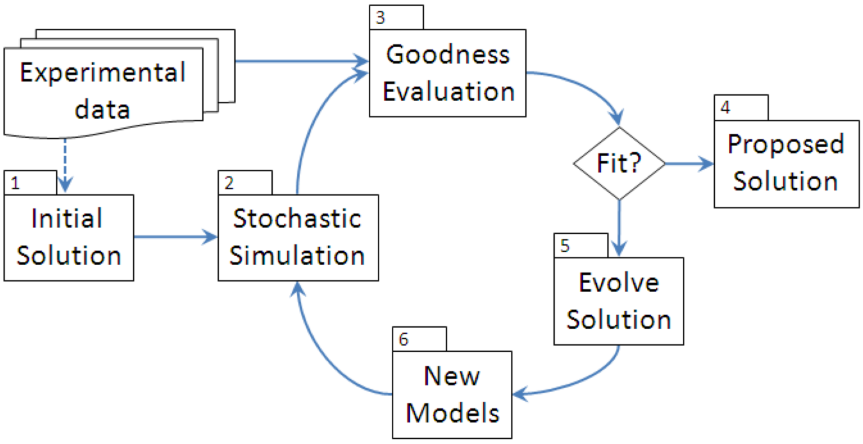


Figure 4.1: The inference scheme.

A number of different methods can be adopted (5). We have developed two different evolutionary strategies that come from two different subdisciplines: evolutionary computation and computational intelligence, Genetic Algorithm (GA) [30] and Particle Swarm Optimization (PSO) [35] respectively. Both algorithms are widely studied and present some advantages and drawbacks in solving optimization problems. In particular, GAs represent a class of optimization algorithms that emulate the natural evolution. They work by evolving populations of individuals using the computational counterparts of genetic operators (i.e. selection, recombination, mutation). PSO, instead, is a population based stochastic optimization algorithm inspired by social behaviour of bird flocking or fish schooling. Both of them are well suited to deal with the stochastic evolutionary inference.

Although recent works have shown the ability of evolutionary approaches, and in particular genetic algorithms, to tackle high dimensional biochemical design of experiments for the discovery of new compounds [66, 26], results from a simulation study of PSO algorithm in the same context [24] have shown that PSO outperforms GA, leading us to concentrate only on the use of PSO.

The use of evolutionary algorithms is particular suited to deal with common problems of parameter estimation like incomplete and noisy data. If some of the species time evolution data are not available, as it happens frequently due to cost or practical unfeasibility restrictions, the algorithm will evolve the solutions through just what is known, guessing hypotheses for what is unknown.

For what concerns the type of computational model to be used, our choice has been to use process calculi derived language BlenX, as it helps in describe biological systems in a modular and systematic way.

4.1.1 Particle Swarm Optimization for Parameter Inference

Particle Swarm Optimization (PSO) is a swarm intelligence algorithm, first developed and introduced by Eberhart and Kennedy [35] as a stochastic optimization algorithm. It is a heuristic technique inspired by the choreography of a bird flock.

Particle Swarm approaches have been successfully applied to various domains, ranging from flowshop scheduling [41] to data mining [63] and, in design of experiments, for composite box-beam design [64].

PSO is a population-based algorithm, in which the population is called *swarm*, while the search points are called particles. Each particle moves in the search space with an adaptable velocity, recording the best position it has ever visited in the search space, i.e., (in minimizing objective problems) the position with the lowest function value. The adaptation of the velocity is based on information coming from the particle itself, as well as from the rest of the particles. More specifically, each particle has a neighbourhood that consists of some pre-specified particles and the best position ever attained by any member of the neighbourhood is communicated to the particle and influences its movement.

Technically, a swarm $S = (X_1, X_2, \dots, X_N)$ consists of N particles. Every particle $X_i = (x_{i1}, x_{i2}, \dots, x_{in})^T$ is composed by n components

The velocity, V_i, of the i^{th} particle, as well as its best position, P_i, at each iteration step t, are also n-dimensional vectors, $V_i = (v_{i1}, v_{i2}, \dots, v_{in})^T$ and $P_i = (p_{i1}, p_{i2}, \dots, p_{in})^T$.

The best position of neighbourhood particles is $P_{gi} = (p_{g1}, p_{g2}, \dots, p_{gn})^T$.

Let t be the iteration counter. Then, the velocity and position of X_i are updated according to the equations:

$$V_i(t+1) = \omega V_i(t) + c_1 r_1 (P_i(t) - X_i(t)) + c_2 r_2 (P_{gi}(t) - X_i(t)) \qquad (4.1)$$

$$X_i(t+1) = X_i(t) + V_i(t+1) \qquad (4.2)$$

where ω is the inertia factor, c_1, c_2 are positive acceleration parameters called cognitive and social parameter respectively, and r_1, r_2 are vectors with components uniformly distributed in the range $[0, 1]$. All vectors operations in equations (4.1) and (4.2) are performed componentwise.

In the context of parameter estimation for stochastic models of biological systems, particles are composed by the set of parameters to be estimated, generating thus a real-valued vector.

The implemented PSO algorithm initially selects a random initial population of particles, that is, the initial position of particles is randomly selected within a predefined interval.

Each subsequent algorithm iteration modifies particles positions according to equations (4.1) and (4.2) with inertia factor ω consisting in a linearly decreasing function in the number of iteration as to enhance exploration for the initial stages of the searching problem, while then preferring exploitation properties. Cognitive and social parameters have been taken as fixed $c_1 = c_2 = 2$ meaning that each particle trusts its best position as much as the position of the best neighbouring particle.

The type of neighborhood used is a ring topology of radius 3, based on particle's index inside the population. It means that the neighborhood of a particle is composed by the previous particle in the population, itself and the successive one.

Finally, we decided to fix the number of particle in the population $n = 30$.

4.1.2 Requirements

In order to use the entire evolutionary framework for parameter inference there is the necessity of some input information and settings.

The starting point is the BlenX computational model for the system under investigation and related experimental data. Experimental data often has to be analyzed and transformed to meet stochastic modeling representation, that is to translate them from concentration measurements to absolute values. This step is necessary considering that the simulation algorithm (i.e. Gillespie's SSA) handles species copy numbers. Moreover, a model should be thought and designed to meet the study goal but also to properly work with the available experimental information, e.g., models that are too complex should be simplified if the experimental data is limited.

A further requirement is given by the set of intervals (lower and upper bounds) for every parameter in the system to be inferred. These intervals do not represent a fixed search space for parameters, but only a first set of values from which sampling the initial conditions of the inference framework. For every further iteration of the procedure these intervals do not constitute any limit for parameter values. Clearly an initial interval containing or close to hypothetic good values helps in reaching a faster convergence state.

The general inference framework does require other input settings. More precisely it should be defined a maximum number of iterations, as well as a halting criteria represented by a value for the cost function used. If the average value of the cost function among the best 10 positions of the particles in the cumulative history of the procedure is less than the given value, the entire procedure stops. This value is problem dependent as it is related to the used cost function. To properly identify a valid stopping value for the cost function, some preliminary analyses are usually necessary.

Summing up, to start the evolutionary inference framework we need:

- BlenX model of the system;
- Experimental data to match;
- Values intervals for model parameters to be estimated (from which initial guesses will be drawn);
- Maximum number of iterations;
- Stopping value for cost function (optional).

4.1.3 Implementation

The general framework for parameter inference of stochastic biological models has been developed by merging different tools and approaches. It consists in repeated interactions among Python [69] programming language code, BlenX model simulator and statistical procedures and methods implemented in R [65].
The manager of the information is the Python code which handles information passing between the other two programs. BlenX simulator is in charge to perform a single stochastic simulation for a given model parameters configuration, while the entire PSO procedure has been implemented in R.
The first step selects the initial position for any particle of the population. This is done by an R procedure which takes as input the set of value intervals for model parameters in input and returns a random sample from it. Then, the python code creates, for every particle initial position, a corresponding BlenX model and run the BlenX simulator to obtain a stochastic simulation from that model.
Once all the models constituted by particle positions have been simulated, R code executes the PSO algorithm described in the previous section, evaluating the goodness of each particle position through a cost function. This cost function is usually a least squares estimator based on the difference from the observed simulated behavior and experimental data. If halting criteria are not met then, based on the value of the cost function and following equations (4.1) and (4.2), new particles positions are generated.
The process is repeated until the maximum number of iteration or the cost function threshold have been reached.

4.1.4 Test: thermal isomerization of α-pinene

To fully understand how the evolutionary inference framework works, we used a test example to evaluate it. The system is the one of thermal isomerization of α-pinene in which we want to estimate 5 rate constants (p_1,...,p_5) of a complex biochemical reaction. This pathway has been originally studied by Box and coworkers [6], and it is also part of COPS (Collection of large-scale Constrained Optimization ProblemS) [22].

The system is depicted in figure 4.2. α-pinene (Y_1) is converted into dipentene (Y_2) and alloocimen (Y_3) which in turn yields α- and β-pyronene (Y_4) and a dimer (Y_5). Experiments on this process have been conducted and reported by Hunt and Hawkins [33], reporting the concentrations of the reactant and the four products at eight time intervals.

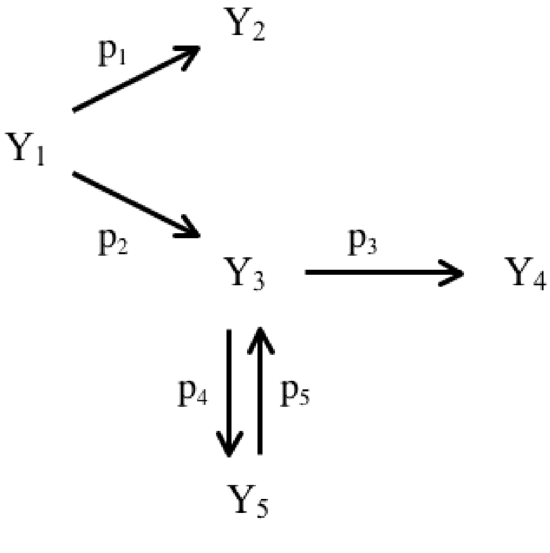


Figure 4.2: Mechanism for thermal isomerization of α-pinene

We built a stochastic model of the pathway using the BlenX language. Nonetheless, to infer the model parameters we did not convert concentration measurements into absolute species abundances. In fact, as the goal was just to evaluate the methodology, and thus not to describe the stochas-

tic kinetics of the system, we only scaled experimental data and performed the estimates on the concentrations variations. By doing so, we described the reaction rates as they would represent concentration variations.
Finally estimated model parameters were compared with those already available.

Input settings

BlenX stochastic model of the hermal isomerization of α-pinene has been built using an event based approach, in which every species is declared and every action the system may take is entirely governed by these events. Associated with each event there is a rate constant representing the parameter set to be estimated.
Experimental data, has been multiplied by a factor 10, as to avoid possible model deviations due to stochastic effects.
Once defined model and experimental data to match we proceed defining procedure's parameters. In particular we need to define a proper fitness function, halting criteria and initial parameters ranges.
The fitness function, representing the goodness of fit of each simulation with respect to the experimental data is based on sum of squares errors.
Halting criteria are twofold. The first one, based on number of iterations, has been arbitrarily fixed to 100, while the second, based on a fitness function threshold, defining the goodness of fit, has been fixed to an average 5% distance from experimental data (i.e. equal to 380).
The last input to be set is the set of initial ranges for parameters. We recall here that they are not the ranges in which the procedure will search for good estimates, but just an initial interval from which randomly guessing a first population of solutions. The procedure then may look for good estimates out of that intervals at later steps.
We used the method for generating initial parameter ranges described

Parameter	Lower limit	Upper limit
p1	0	0,000044
p2	0	0,000044
p3	0	0,000025
p4	0	0,000025
p5	0	0,000046

Table 4.1: Initial parameter ranges

Parameter	Best known value	Estimate	Standard error
p1	5.93e - 5	6.14e-05	5.40e-07
p2	2.96e - 5	2.92e-05	5.07e-07
p3	2.05e - 5	2.11e-05	2.72e-07
p4	27.5e - 5	30.0e-05	1.84e-05
p5	4.00e - 5	3.89e-05	2.81e-06

Table 4.2: Estimated parameters with standard errors and best known value

above ending with intervals for the parameters reported in table 4.1.

Results

We run three times the evolutionary inference framework. In all three cases the procedure stopped after reaching the fitness threshold. The average number of iterations performed in the three runs was 14 with an average time to convergence of 90 seconds. Given the good and fast results, we decided to scale down the threshold lowering it to a 2.5% of average difference with respect to experimental data. The procedure took 54 iterations to converge, with a computation time of 5 minutes on a standard Apple MacBook laptop, 2.0GHz processor and 1Gb RAM.
Estimated parameters are reported in table 4.2 together with the true parameters from the literature and standard error for the estimates.
As can be seen by glancing at the above table, the parameter estimates

are very close to the best known values. To confirm the goodness of fit we report in figure 4.3 the behavior of the best solutions identified together with experimental points. Traces on the plot represent a single stochastic simulation which has shown a good fitting. The procedure identified 5 solutions with a fitness function value lower than the predefined threshold.

The same system has been used in [58] to test a new inference procedure based on an evolutionary technique and using a deterministic modeling approach. The presented approach is able to efficiently tackle the problem in a finite amount of time even considering wider parameter intervals value with respect to previous approaches, which were unable to obtain

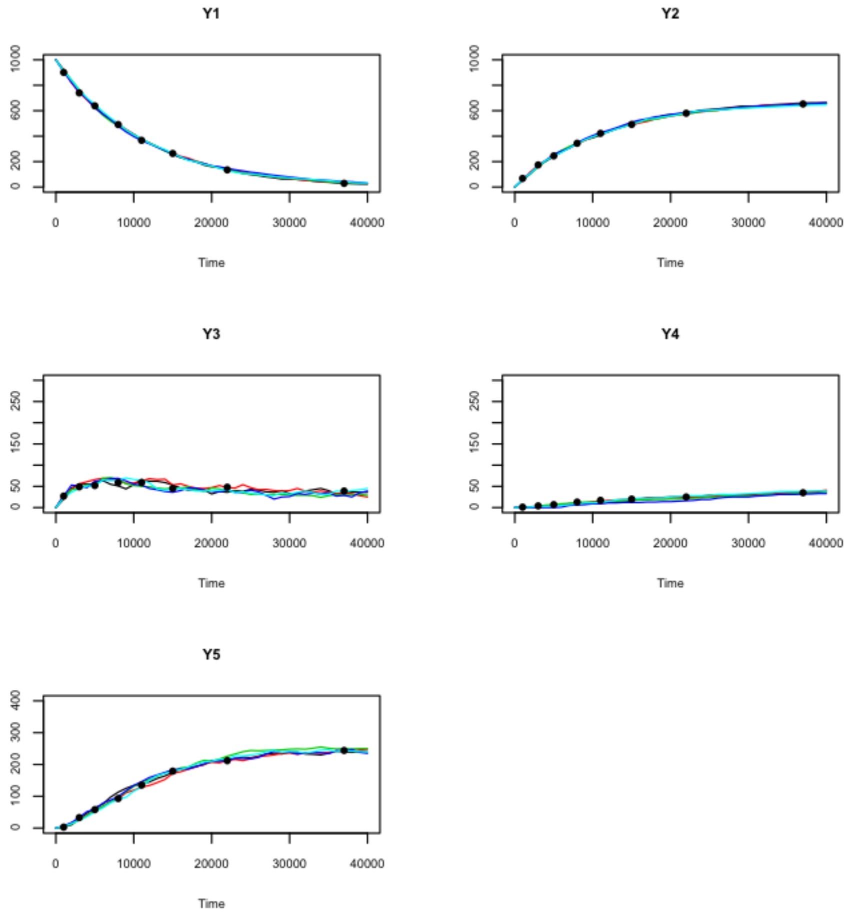


Figure 4.3: Best solutions identified and experimental points

solutions.

Even if the evolutionary procedure presented here does not give the same results when ranges are increased to the paper's original values, we showed that an appropriate choice of that intervals is able to drastically reduce computational time to achieve optimal solutions.

Concluding, with this example we showed the potentialities of the evolutionary framework to solve parameter estimation problems in a stochastic setting.However, it is worth highlighting, that most of the times estimation tasks are more complex, as usually experimental data is not complete (i.e. they are not available for every species) and they may be very sparse.

When we will tackle real cases in the proceeding of the thesis, these aspects will clearly arise, showing how the approach is well suited to deal with real problems.

4.2 Model Analysis

The major advantage of working with computational models of a biological system lies in the ability to use the model as a perfect counterpart of the real system. With a model we can reproduce experimental evidence but also derive new information by deeply analyzing it without requiring real experimentation.

Such a use of models although desirable, is not straightforward. Once we have defined an acceptable model in terms of abstraction and correspondence to available experimental data, its analysis poses several challenges. These challenges are even more pronounced when we are dealing with stochastic models. As we have pointed out, only very small stochastic models can be analyzed analytically and thus the unique way to observe model behaviors is to heavily rely on stochastic simulations.

A stochastic simulation of a model represents a random realization of the underlying stochastic process. Consequently, simulation runs differ from one another often making it hard to derive predictions about the system behavior. In fact, in order to obtain statistically reliable conclusions, a large number of stochastic simulations have to be run, and those runs have to then be pooled together in a stastically sound way. This highlights the clear need for fast and efficient methods to tackle the computational demands of repeated simulations.

4.3 Statistical approximate model checking

A model of a biological system essentially describes the dynamics of a population of n interacting biochemical species $S_1, S_2, \ldots S_n$. Analyzing the behaviour of such models entails looking for the occurrence of biologically relevant events during the evolution of the system.

When dealing with discrete stochastic modeling of biological systems, biochemical species are enumerable quantities representing the number of molecules of a given substance, and the evolution of the system is probabilistic, rather than deterministic, leading to Continuous Time Markov Chain (CTMC) models. Unfortunately finding numerical solutions of CTMC models is unfeasible for most realistic case studies.

A common way to analyze stochastic models is given by query based verification methods (model checking [13]). Biologically relevant events can be formally characterised as temporal logic formulae that can then be automatically checked against a discrete state model. But still, these methods suffer of the state-space explosion problem which limits their accessibility especially in systems biology where very large models are common.

Approximate methods, based on statistical measures, have been proposed as an alternative to exact probabilistic model checking that allows to get an estimate of the likelihood of a condition to hold of a CTMC model. This paradigm essentially comprises three ingredients: (*i*) a stochastic engine that generates trajectories of the underlying state space; (*ii*) a model checking algorithm capable of analyzing a single trajectory; (*iii*) a statistical support for estimating the accuracy of the answer. The advantage is that, contrary to exact model checking, it does not need to build (i.e. to store) the state space of the model as it only explores a limited number of (finite) trajectories. The cost paid for such space saving is in terms of precision of the calculated measure.

We realized a methodology that given a CTMC model M, a property ϕ and a desired *level of confidence* estimates the probability of ϕ to be satisfied by M with the estimated measure meeting the desired confidence. The method we propose is based on three key aspects: (*i*) the trajectory generation is controlled by on-the-fly verification of the considered formula which means that simulation halts as soon as a state which verifies (falsifies) the formula is reached. (*ii*) we use an efficient statistical method (i.e. a variant of the *Wilson score interval method*) which results in smaller samples (i.e fewer simulation runs) in order to meet the desired confidence. (*iii*) the whole simulation/verification framework has been designed and tested on a parallel prototype, which is based on independent simulation/verification engines generation, and a MPI client/server parallel computation architecture.

4.3.1 On-the-fly Bounded Linear-time Temporal Logic with numerical constraints verification

Statistical verification of a CTMC model M is based on the simple principle of collecting N sample realisations σ_i ($i \in \{1, \dots, N\}$) of M and verifying each of them against a given property ϕ. The estimate of the likelihood of ϕ to hold true of M is obtained as the frequency $\hat{p}_\phi = \frac{po}{N}$ of positive outcomes (po) of the verification of ϕ versus σ_i.

In order to state properties of simulated trajectories, we defined a temporal logic, namely BLTLc logic (Bounded Linear-time Temporal Logic with numerical constraints) which combines Constraint LTL (LTLc) and Bounded-LTL (BLTL). Both LTLc and BLTL are based on classical LTL [50] temporal operators. However while LTLc allows for using (complex) arithmetical conditions between state variables, it does not allow for expressing time bounded conditions. On the other hand with BLTL time bounded LTL expressions can be formed but based on simple non-arithmetical conditions rather than on a grammar for arithmetic expressions as it is the case with LTLc.

BLTLc formulae are evaluated against timed-paths resulting from simulation of a CTMC model. The formal semantics of BLTLc formulae, expressed in terms of the $\models$ relation, is given below, where σ is a timed-path of a CTMC model.

As an example of the expressiveness of the BLTLc logic consider the following formula $\phi \equiv [(X_1 < Sqrt(X_2))\ U\ (X_2 \geq 10 + X_3)]$ which states that the concentration of X_1 shall be less than the square root of that of X_2 until that of X_2 exceeds X_3 by at least 10.

The verification of a formula ϕ on a CTMC model M is performed *on-the-fly* meaning that simulation proceeds with the generation of the next state only if ϕ is neither satisfied nor falsified in the current one (and if the simulation time limit has not been reached). Verification of temporal formulae may result in passing of the already-generated trace (available in a buffered trace σ_{buff} that results from verification of a sub-formula) from the inner-most sub-formulae to the outer-most ones. For time-unbounded formulae, we adopt a *pessimistic* approach: if the simulation max time t_{max} is reached and the formula is neither verified nor falsified then the algorithm returns *false*. Thus the exact probability of time-unbounded formulae is an upper bound of the estimated one.

4.3.2 Estimating the probability of a property

Checking of a BLTLc property on a simulated trace corresponds to a Bernoulli experiment, where the outcome can be either positive or negative. Thus the number of successes that results from reiterated checking of the same property on n independent simulations, represents a random variable X with a Binomial distribution. The point estimation of the unknown probability of success p out of n independent trials, is given by the well known maximum likelihood estimator $\hat{p} = po/n$, where po represents the number of successes. Clearly the reliability of such estimate is highly affected by the number of simulations performed, i.e. by the sample size n. As a consequence the point estimate $\hat{p}$ is usually associated with a confidence interval, expressed in terms of a real value $\alpha \in (0, 1)$, which represents the range within which the actual value of the unknown parameter θ (i.e. the actual probability of the considered formula to hold against the simulated model) shall fall $(1 - \alpha)\%$ times[1].

The standard approach to compute the confidence interval for the probability of success of a binomial distribution uses the normal approximation, producing the so-called Wald interval. Brown *et al.* [8],[9], have studied the coverage characteristics of different types of binomial proportion confidence intervals, and they showed that the Wald interval present unstable coverage characteristics also for large n, suggesting thus the use of other types of confidence intervals. Among the discussed intervals, the Wilson score interval [73] has shown good coverage characteristics also for small

[1] There exists a strong connection between confidence intervals and hypothesis testing: all the values θ_0 for the unknown parameter θ external to a $1 - \alpha$ confidence interval would end in the rejection of the two sided hypothesis testing (i.e. Null Hypothesis $H_0 : \theta = \theta_0$) at the α level.

n and extreme probabilities. Wilson score confidence interval is calculated by means of $Wilson\_interval(\hat{p}, n, \alpha) = [L, U]$ with,

$$[L, U] = \frac{\hat{p} + \frac{1}{2n} z^2_{1-\alpha/2} \mp z_{1-\alpha/2}\sqrt{\frac{\hat{p}(1-\hat{p})}{n} + \frac{z^2_{1-\alpha/2}}{4n^2}}}{1 + \frac{1}{n} z^2_{1-\alpha/2}} \tag{4.3}$$

where $\hat{p}$ is the estimated probability from the statistical sample, α is the confidence level, $z_{1-\alpha/2}$ is the $1 - \alpha/2$ percentile of a standard normal distribution, and n is the sample size.

We can now use this equation to ask which is the proper sample size to obtain a confidence interval of a given width at a specific confidence level α. In particular, the sample size required for the Wilson interval of width 2ϵ at $1 - \alpha$ confidence level can be obtained simply by solving for n the Wilson score limits in equation (4.3) [49]: formally this is given by function $Wilson\_sample(p, \epsilon, \alpha) = N$ with,

$$N \geq z^2_{1-\alpha/2} \frac{\hat{p}(1-\hat{p}) - 2\epsilon^2 + \sqrt{\hat{p}^2(1-\hat{p})^2 + 4\epsilon^2(\hat{p} - 0.5)^2}}{2\epsilon^2} \tag{4.4}$$

where $\hat{p}$ is the frequency of positive outcomes of a re-iterated Bernoulli experiment. As we usually do not have a guess of the probability $\hat{p}$ to be used in (4.4), the standard approach is to take a conservative estimate, by considering $\hat{p} = 0.5$ which is the estimate with maximum variance, and, as such, produces the highest sample size.

The method we developed consists in adopting a different approach in determining the sample size. By iterating (4.4) with successive estimates of $\hat{p}$ we are able to drastically reduce the number of samples required when the true probability p is far from 0.5. More specifically, given a confidence interval width 2ϵ and a confidence level $1 - \alpha$, the algorithm starts by calculating the sample size required for an initial estimate $\hat{p} = 1$ (or equivalently $\hat{p} = 0$) and returns the minimum number N of simulations to be performed. After computing the proportion of successes the new estimate $\hat{p}$

is rounded by adding or subtracting the quantity ϵ if $\hat{p} \leq 0.5$ or $\hat{p} > 0.5$ respectively. The rounded estimate p' is then used to recalculate the sample size resulting in N'. If we have already performed a cumulative $N_{tot} \geq N'$ simulations the algorithm stops. Conversely, we iterate the process again by launching $N' - N_{tot}$ simulations.

The $\hat{p}$ rounding step is crucial and ensures that a successive sample size calculation would avoid undersized samples due to erratic estimates. If the current estimated probability of success drifts from the true unknown one, towards extreme probabilities, of more than ϵ, this would produce an undersized sample which would produce a confidence interval not covering the parameter at $1 - \alpha$ level.

By using this iterative method for the determination of sample size we drastically reduce the number of required samples with respect to the conservative approach that starts with $\hat{p} = 0.5$. Of course this gain is greater when the actual p is close to the extreme values $p = 0$ and $p = 1$, while using the same sample size for p close to 0.5. Figure 4.4 represents the sample

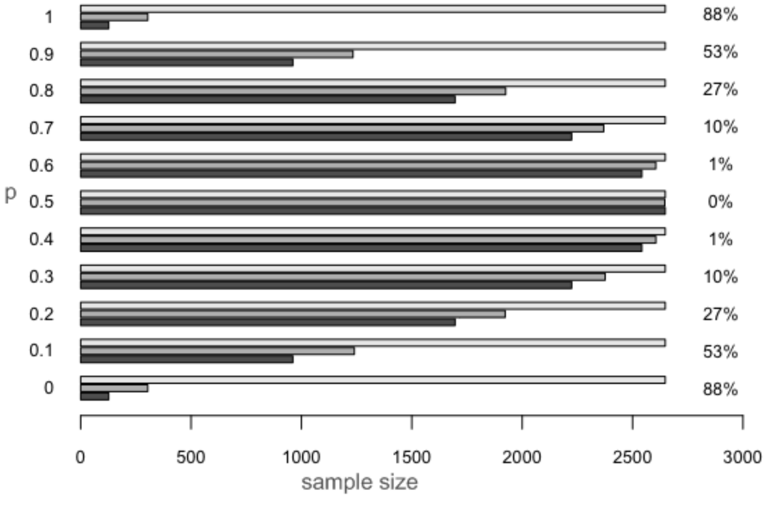


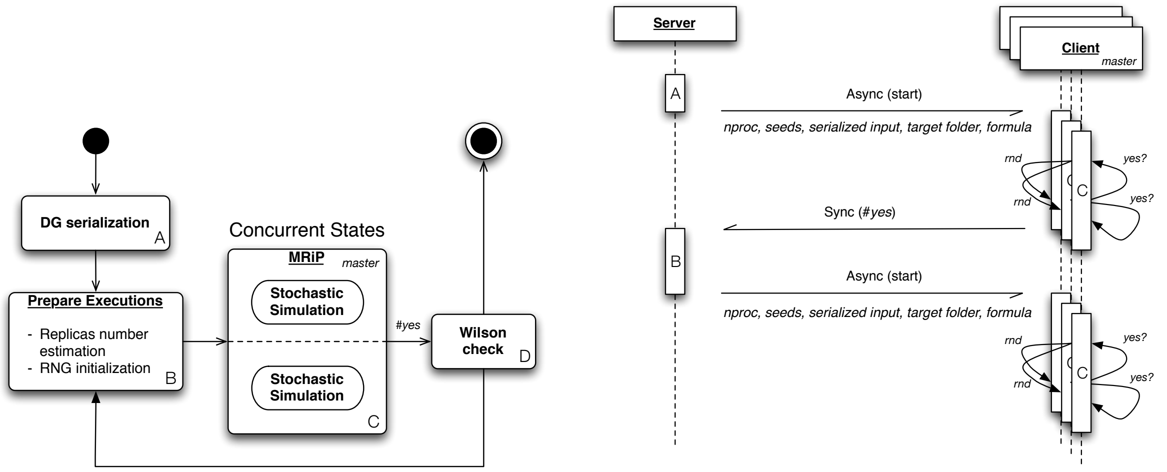
Figure 4.4: Average sample size for CI-width= 0.05 at 99% confidence lelvel. Comparison of sample size required with i) conservative approach, ii) Iterative Wilson, iii) Minimum sample size with known p

size required to obtain a confidence interval of a given width $2\epsilon = 0.05$ at a confidence level $1 - \alpha = 0.99$ for values of p ranging from 0 to 1 by 0.1. Bars from light gray to dark gray represent respectively: **I**. Conservative approach (using always $\hat{p}$=0.5); **II**. Iterative Wilson; **III**. Minimum sample size if the unknown p was known (i.e. $\hat{p} = p$). On the right side of the plot we reported the percentage of reduction in sample size by using the Iterative Wilson approach with respect to the Conservative approach. As it can be easily seen, we have the strongest reduction in sample size when the true p is close to 0 or 1. By using the Iterative Wilson method, we can reduce the sample size required by up to 88% with respect to the Conservative approach. This reduction at extreme values is explained by the fact that at those probabilities the variance of the binomial distribution is smaller, so we need less samples to obtain good estimates.

4.3.3 Implementation

What so far illustrated has been implemented within a distributed software architecture. It is actually made of two distinct modules: a graphical front-end and a remote simulation engine. The front-end part acts as a server and is in charge of drawing the computational graph relative to the loaded BlenX models.

The prototype collects any information from a BlenX model and serializes it in a proprietary, xml-based, data format along with all the simulation information manually inputted by the user (see Figure 4.5(a) - State `A`). Further information about the logical formula to be checked, the α and ϵ values are required only whenever one wants to automatically calculate the number of replicated simulations (or “replicas”) needed to reach the required confidence threshold. However, both in the case that the number of replicas is user-defined and that it is automatically computed, a random number generator is instantiated and used to make a stream of initial



(a) Wilson-driven computational loop (b) Inter-process and client-server communications

seeds, one for each simulation (see Figure 4.5(a) - State B).

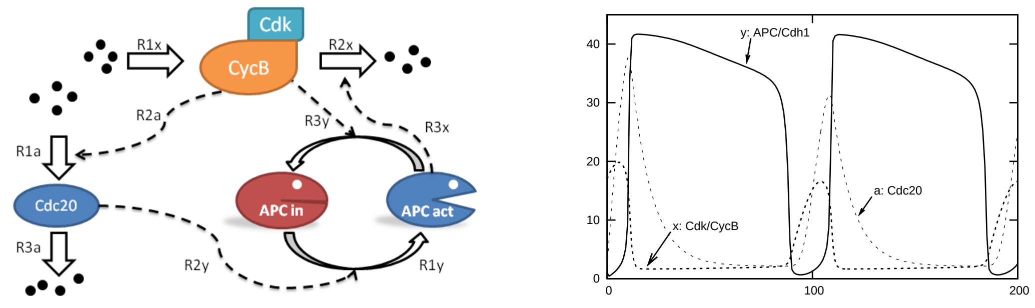
As soon as the simulation task is invoked by the user, a number of independent simulation engines is instantiated (see Figure 4.5(b) - Activity A). Among them, one is entitled to be master. The master handles both the inter-process and the client-server communications. In the former case, it takes care of scattering and dispatching the initial seeds to the slave processes (and to itself) and of gathering the results (see Figure 4.5(b) - Activities C). In the latter case, the master node is responsible for counting the computed YES and for its sending to the server (see Figure 4.5(b) - Activities C). Hence, each process simulates independently (see Figure 4.5(a) - State C and Figure 4.5(b) - Activities *C*) and evaluates on-the-fly a logical formula, giving a boolean answer. The summation of the positive answers is sent to the server, which recomputes the Wilson method and returns a new number of simulation replicas to be performed (see Figure 4.5(a) - State D and Figure 4.5(b) - Activity B). This loop halts only when no more replicas are requested

4.3.4 Test: buddying yeast cell cycle

We consider a stochastic model of (a part) of the regulatory network that controls the buddying yeast cell cycle [48]. Once identified few BLTLc formulae characterizing relevant aspects of the cell-cycle behavior, we then run the statistical verification tool to estimate the probability of the considered formulae. To assess the accuracy of the statistical procedure we compare the estimates obtained through our statistical model checker with the exact values calculated through numerical model checking, namely by means of the PRISM model checker [37]. As the state-space dimension corresponding to the original cell-cycle model is too large to be handled through numerical model checkers we consider a "scaled-down" version of the model for validating the statistical model checking approach against the numerical one.

Input settings

To run the statistical verification tool, we clearly need to analyze the stochastic model first. We consider here a simplified cell-cycle model sketched in Figure 4.5(c). It consists of three species, x (Cdk/CycB complex), y (activated APC/Cdh1 complex) and a (activated Cdc20), and nine molecular reactions listed in Table 4.3 and with parameters in Table 4.4. Complex x is synthesized and degraded by reactions R_{1x} and R_{2x}, respectively. Complex y speeds up x production by means of reaction R_{3x}. At the same time, x deactivates y by R_{3y}. Also, y turns active by itself with R_{1y} and with the help of a in reaction R_{2y}. Finally, a is produced and consumed by R_{1a} and R_{2a} and regulated by x in reaction R_{2a}. Such system behaves as a bistable switch with two stable states: G1 with low x and high y, and $S/G2/M$ with high x and low y, as shown in the plot of Figure 4.5(d).



(c)

(d)

Figure 4.5: Budding yeast cell-cycle cartoon and simulation

Cdk/CycB				APC/Cdh1				Cdc20			
R_{1x}	$\emptyset$	$\xrightarrow{k_1 \alpha}$	x	R_{1y}	y_{in}	$\xrightarrow{k_3^*}$	y	R_{1a}	$\emptyset$	$\xrightarrow{k_5' \alpha}$	a
R_{2x}	x	$\xrightarrow{k_2'}$	$\emptyset$	R_{2y}	$y_{in} + a$	$\xrightarrow{k_3'''}$	$y + a$	R_{2a}	x	$\xrightarrow{k_5^*}$	$x + a$
R_{3x}	$x + y$	$\xrightarrow{k_2'' \alpha}$	y	R_{3y}	$x + y$	$\xrightarrow{k_4^*}$	$x + y_{in}$	R_{3a}	a	$\xrightarrow{k_6}$	$\emptyset$

Table 4.3: Budding yeast cell-cycle reactions

The second input setting is the set of formulae to be evaluated. We target the experiments to the study of the so-called $S/G2/M$ transition which begins in states with low level of activated APC, high concentration of Cdk/CycB and (initially) low level of Cdc20. By looking at the topology of the network in Figure 4.5(c), and at the form of the corresponding equations (Table 4.3), it is evident that a (i.e. Cdc20) plays a fundamental part in the activation of y hence in the controlling the $S/G2/M$ transition. Specifically the progressive growth of a results in the (initially slow) activation of y which then, in turns, is responsible for the degradation of x. The influence of a on y can be studied through BLTLc formulae of the following type:

$$\phi_1 \equiv (a \leq i) \; U \; (y \geq j), \qquad \phi_2 \equiv (a \leq i) \; U^{\leq t} \; (y \geq j)$$

Component	**Rate Constant**	**Dimensionless constants**
Cdk/CycB	$k_1 = 0.04,\ k'_2 = 0.04,\ k''_2 = 1,\ k'''_2 = 1$	$J_3 = 0.04, J_4 = 0.04$ $J_5 = 0.3$ $m = 0.80$ $\alpha = 0.00236012$
APC/Cdh1	$k'_3 = 1,\ k''_3 = 10,\ k'_4 = 2,\ k_4 = 35$ $k^*_3 = \frac{k_4 m\alpha xy}{J_4+(\alpha y)}\ k'''_3 = \frac{k''_3 \alpha a y_{in}}{J_3+(\alpha y_{in}}$ $k^*_4 = \frac{k'_3 y_{in}}{J3+(\alpha y_{in})}$	
Cdc20	$k'_5 = 0.005,\ k''_5 = 0.2,\ k_6 = 0.1,\ k_4 = 35$ $k^*_5 = \frac{k''}{\alpha} / \frac{J_5}{m\alpha x}$	

Table 4.4: Parameter Values: Cell Cycle toy model

Formula ϕ_1 represents the possibility that y grows above the threshold j while a does not exceed the threshold i. Since y gets abruptly activated only after a has reached high concentration (see Figure 4.5(d)) then, for $i < j$ and $\delta = j - i$, we expect a low probability of ϕ_1 for large δ, and a higher probability of ϕ_1 for high i and small δ[2].

Results

Figure 4.6 compares exact versus estimated probability measure for the time-bounded formula ϕ_2, verified with respect to different time points ($t \in [0.2, 1.6]$ step 0.2). The (cross marked) point estimates (depicted in Figure 4.6 together with their confidence interval), have been calculated with 99.99% confidence and 0.005 interval semi-amplitude ($\epsilon = 0.005$). The exact values computed with PRISM (red plot in Figure 4.6) fall within the confidence interval of each point estimates, confirming the accuracy of the statistical verification method we have realized.

[2] ϕ_2 allows also to study the dependence on time of such an attitude.

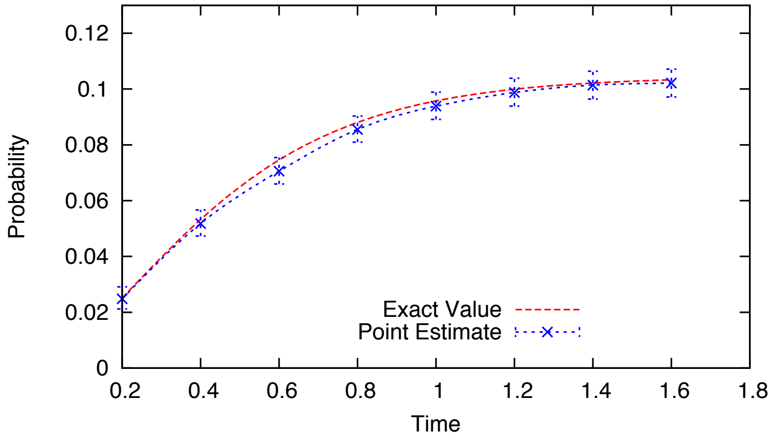


Figure 4.6: Exact vs Estimated probability of the time-bounded Until formula: $(a \leq 4)U^{(0,t]}(y \geq 5)$, estimates with 99.99% confidence and $\epsilon = 0.005$ semi-interval amplitude.

4.4 Multivariate analysis to detect effects of parameters changes in stochastic models

A model of a biological system is characterized by its structure and associated parameters. Model parameters describe the temporal dynamics of the system components. It is then clear that once identified a model behavior of interest, we may want to focus on the effects that model parameters have on that behavior, and possibly try to evaluate and predict them.

One of the biggest challenges comes form the size of parameters space associated to a model, that often makes it difficult to estimate the effects of a perturbation of the parameters. Moreover, when dealing with stochastic models, we have to take into account the differences arising in the model dynamics even considering fixed initial conditions. For this reason, multiple replications of the same model simulated in parallel (MRiP) is widely used as it allows to estimate an averaged behavior and thus to use statistical tools to justify the sensitivity of parameters [3, 4].

We extend the concept developing a framework for efficiently generating and detecting peculiar behaviors of a given model by perturbing its parameters space [25].

4.4.1 Sampling the parameter space

The first step in order to evaluate the effects of parameters changes on a model output is to properly define how these parameters have to be varied. Different approaches for properly sampling the parameter space exist. The simplest one lies in changing a single parameter at a time, keeping fixed all the others. It captures 'first order' effects only. However, often it happens that interesting behaviors are linked to "higher order" effects, which are those effects that derive from the combination of some parameters. In this case, sophisticated sampling techniques are required.

Simple Random Sampling picks a number of samples by randomly selecting them among the entire population. With this kind of sampling we can estimate higher order effects, but with low or no efficiency gain. On the contrary, a full factorial sampling scheme takes every possible combination of parameters values. In this way, the parameters space is fully explored and higher order effects considered. If, on one hand, this sampling approach works well with small models, on the other hand, it fails when the number of parameters configurations explodes.

A more efficient sampling scheme is the one named Latin Hypercube Sampling (LHS). LHS, firstly introduced by McKay [45] allows to efficiently sample the parameters space, but still preserving the potential of estimating high order interactions. LHS divides the space of each parameter into N equiprobable intervals and picks (randomly) a single value from every subinterval. Thus, LHS allows sampling of the entire parameters space in an efficient manner, by essentially reducing the number of required simulations.

4.4.2 Simulation, aggregation and analysis

A valid sampling scheme for the parameter space returns a number of parameter configurations to be simulated. We consider here stochastic simulation, i.e. the evolution of a biological system is described by a stochastic process. The stochastic algorithm used is the well-known Stochastic Simulation Algorithm (SSA) [28].

The generation of a number of parameters configurations leads to the generation of as many sub-models. Consequently, their simulation furnishes as many traces. The aim of this step is to focus on a property of interest that is common among all the traces and, then, to aggregate them over it. Depending on the observer's interests, different measures of aggregation of the model output can be used. In order to properly derive the forthcoming analysis, a great attention should be paid on this measure. An erroneous measure could later lead to conclude that model parameters do not statistically effect the output trend when, instead, it could possibly be the case.

After aggregating simulation results, the last step is to perform statistical analyses in order to estimate and validate potential effects of parameter changes on the model output.

Statistics is an essential tool for identifying and quantifying the effects of the parameters changes. There is a vast literature describing numerous statistical methods which can be used in different situation. If, for instance, we were interested in determining the possible statistical relationships among some model's parameters and its simulation output, then we should be aware that for qualitative (or categorical) outputs we should use the analysis of variance, whilst for quantitative output, the multiple regression analysis. If multiple regression can be used also for categorical data, with proper output transformation, the reverse is not true.

After the estimation process, the next step is the statistical validation that we conduct through t-test or F-test for regression. These are used to give significance to the previous estimate and to check that a relationship among a specific configuration and a simulation output actually holds. In some cases these tests are not sufficient to reach a conclusion. This is even truer for those statistical methods that rely on some assumptions over the estimates or residuals of the results. Regression, for instance, relies on the assumption that residuals of the estimated model are uncorrelated and normally distributed. Before drawing a final conclusion, it is then necessary to check this assumption by means of plots or quantile-quantile distribution.

4.4.3 Implementation

The methodology has been implemented in a plug-in based software prototype that runs under the `.Net framework 4` runtime and it is written in `C#`.

The tool takes care of loading the proper settings (that include input models location, stochastic simulators, analysis plugins, and hardware configuration). It then generates a model for each parameter configuration supporting both factorial and LHS samplings. Stochastic simulations are then launched (either locally on multicore processors or remotely on clusters of processors). To obtain statistical significance, the same model is simulated a number of times equal to a given input parameter. Once simulations are complete the analyzer evaluates the property of interest on every simulation and stores the results in a shared data structure that holds as many records as the number of parameter configurations. This data structure is a simple table whose rows contain each parameters configuration plus the relative value calculated by the analyzer.

Finally, multivariate analysis for the estimation and validation of parameters effects is performed using R statistical computing software and its built-in statistical functions.

4.4.4 Test: Predator-Prey model oscillation frequency

To explain the methodology, we consider here the classic Lotka-Volterra (Predator-Prey) model [43, 71] depicted in Fig. 4.7(a). The model comprises three entities: a *prey* and a *predator*, which represent general species of animals, and a third entity *food*, modeling an unbounded resource as, e.g., grass.

The dynamic of this system is governed by three specific probability rate constants, corresponding to as many basic reactions:

Ax. the Prey eats some food and then it duplicates;

Xy. the Predator feeds on prey and then it duplicates;

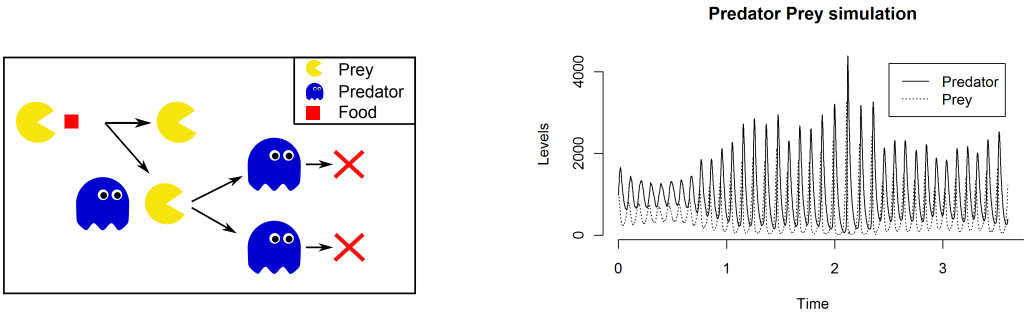
Yb. the Predator dies.

Given an initial large number of preys, the population of predators grows rapidly due to the abundance of food. As soon as the predators number increases, that of preys starts to decrease, thereby resulting in a consequent reduction of predators. Due to the unbounded quantity of food, the preys start to repopulate the systems, and the cycle restarts. An example of the oscillatory behaviour of the Predator-Prey model is given by the simulation reported in Fig. 4.7(b).

We consider the parameters space formed by *Ax*, *Xy*, and *Yb* and we are interested in estimating the influence exerted on the system oscillation frequency.

Input settings

A BlenX program encoding the Predator-Prey model is presented in [17]. In order to proceed we have also to select a proper sampling scheme for the parameter set. Given the relative small dimension of the parameters



(a) Predatory Prey model cartoon

(b) Predatory Prey stochastic simulation with Ax = 0.001, Xy = 0.1, and Yb = 50

Figure 4.7: Predatory Prey Model

space, we rely on a factorial generation of configurations. The sub-models generated are then simulated on an HPC infrastructure.

Finally we use a proper aggregation measure for the system behavior of interest. In our case we are interested in the frequency of oscillation of the systems species. More precisely, our goal is to investigate the possibility that some parameter values can somehow influence the frequency of oscillations of a given stable Predator-Prey system. To do that, we use the fast fourier transform to get oscillations out of the traces.

Results

After performing sampling, parallel execution of simulations and aggregation, we make extensive use of statistical methods to evaluate possible effects of parameter changes on the system's oscillation frequency.

A first descriptive analysis gives us a clear indication of a possible relationship existing between the first parameter Ax and the preys frequency of oscillation (Fig. 4.8). The second and third parameters do not seem to affect the frequencies of the oscillation, although a small relationship between them is registered.

Following a more accurate analysis it becomes clearer that the relation-

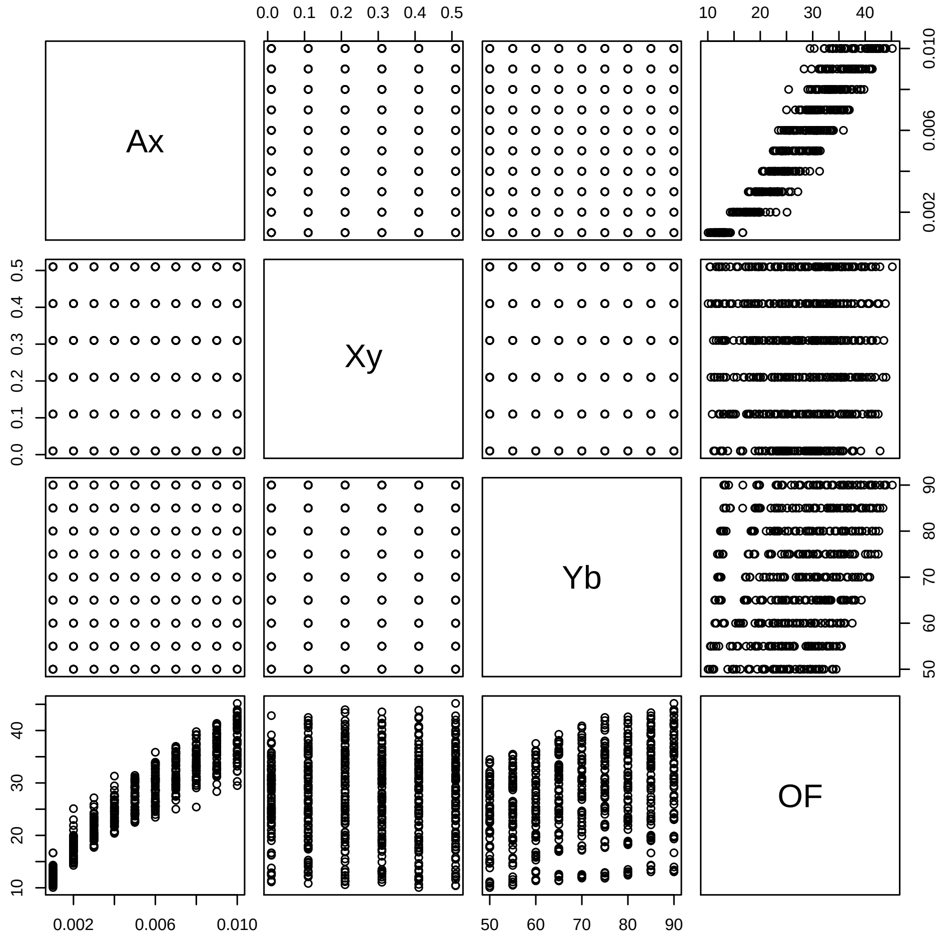


Figure 4.8: Model parameters and Oscillation Frequency paired scatterplot

ship among the first parameter and the oscillation frequency is not linear. Indeed, by employing a bivariate analysis, we easily notice that the combined effect of first and third parameters is almost linear (if plotted against the oscillation frequency).
By testing different regression model hypotheses, we end up with a final model which well describes the oscillations frequency (OF). In particular we have:

$$OF = 1.052 + 44.497\sqrt{AxYb} + \hat{\epsilon}, \quad \hat{\epsilon} \sim \mathcal{N}(0, 1.32) \tag{4.5}$$

where both the intercept and the parameter of the interaction term have a strong statistical significance. T-test values are 5.5 and 145, respectively, with a correspondent p-value of 4.59e-08 and 2e-16.

The goodness of fit of the model, given by the coefficient of determination (R-squared) that shows a value of 0.9752, clearly indicates the validity of the model as also derived from the analysis of the residuals.

This statistical model describes how the oscillation frequency is influenced by the squared root of the interaction between the first parameter (i.e. the rate of eating and reproduction of the prey) and the third parameter (i.e. the rate of death of the predator). Greater these parameters, higher the oscillation frequency.

Such a model lets us make some guesses about the oscillation frequencies. As an example, considering the following basic configuration: $Ax = 0.001$, $Xy = 0.1$ and $Yb = 50$, we know from simulation (and from the statistical model above) that the average oscillation frequencies of the predator and of the prey populations are around 11 Hz. Now, we use the model to predict the oscillation frequency of the preys population with perturbed configurations. In addiction we compute prediction intervals on the estimated frequency values in order to take into account the system variability.

Table 4.5 reports three parameters configurations. If we supposed to set

Ax to be equals to 0.0003, the model would predict a mean value for the oscillation frequency of $OF = 6.50$. If, instead, we supposed to decrease Yb up to 30, we would get a predicted OF equals to 8.76. But, if we decreased both parameters we would obtain $OF = 5.27$, thereby discovering the combined effect of the parametric interaction.

Table 4.5: Statistical model predictions and 95% prediction intervals for modified model parameters

Parameters configurations	Lower bound	**Expected Predicted Value**	Upper bound
$Ax = 0.0003$, $Yb = 50$	3.89	**6.50**	9.12
$Ax = 0.001$, $Yb = 30$	6.15	**8.76**	11.37
$Ax = 0.0003$, $Yb = 30$	2.66	**5.27**	7.89

To finally confirm the prediction provided by the model, we run a bench of simulations with varying parameters. As we can see in Tab. 4.6, the mean oscillation frequency obtained by simulating the three model configurations are close to those predicted and lie within the prediction intervals bounds.

Table 4.6: Statistical model and simulation results for modified model parameters

Parameters configurations	Simulations Results
$Ax = 0.0003$, $Yb = 50$	$OF = 5.84$
$Ax = 0.001$, $Yb = 30$	$OF = 10.60$
$Ax = 0.0003$, $Yb = 30$	$OF = 4.43$

An analytical study of a Predator-Prey model [62] shows a relation between the oscillation frequency and the squared root of the product of Ax and Yb. With the proviso that the relation between a stochastic and a deterministic system is not trivial, the result of [62] agrees with the statistical hypothesis of Eq. 4.5. More important, the two analyses concord on a counterintuitive conjecture: the speed a Predator feeds on a

Prey does not influence OF, i.e., the period of oscillation is not influenced by the greediness of the Predator.

Chapter 5

Case Studies

Knowledge discovery clearly aims at unrevealing useful information about a particular system which is not completely studied. For this reason we put at the very centre of this thesis, the need for developing approaches and methodologies able to face with real systems.

When we test some new methodology on a toy or well studied problem we are completely aware of what we have to expect from it. Contrarily, working with real systems, and possibly in close synergy with experimenters and experiments, make it clear which are the current conditions to work with when dealing with knowledge discovery.

We present two real cases derived from direct collaboration with research institutes. The former is related to the description of V. vinifera general pathway for phenolics biosynthesis leading to flavonoids, in collaboration with Instituto Agrario San Michele all'Adige (IASMA), located in Trento, Italy. The latter case study is related to the development of Leishmaniasis disease, in collaboration with the Universidad La Laguna, Tenerife, Spain.

We want here also to highlight how the synergic use of the proposed methodologies let it possible to deeply face the problem under study, deriving essential information and thus finally making the *in-silico* analysis of biological systems useful.

5.1 V. Vinifera flavonoid biosynthesis

Flavonoids are regarded as one of the most important determinants of quality in red grapes and wines. Color and taste of red wines are strongly related to the amount of anthocyanins, flavonols and proanthocyanidins. Moreover in recent years some flavonoids compound (anthocyanins, proanthocyanidins and flavonols) have attracted additional attention for their potential health benefits ([14], [67]).
Antocyanins, proanthocyanidins and flavonols are synthesized via the flavonoid pathwway depicted in Figure 5.1.
Three different types of compounds are highlighted, as they are those of greatest interest for experimentalists:

- Quercetin, Kaempferol and Myricetin (highligheted in blue) represent flavonols, and they are all synthesized by a flavonol synthases (FLS).

- Catechin, Gallocatechin, Epicatechin and Epigallocatechin (green) represents flavan-3-oli (or flavanols) and they are synthesized by two enzymes (LAR and ANR).

- Cyanidin-3-glucoside and Deplhinidin-3-glucoside (red) are instead two primary anthocyanins synthesized by UFGT.

The research institute IASMA (Istituto Agrario San Michele all'Adige) located in Trento, Italy has been investigating the Vitis Vinifera genome [70]. Given the interest in the flavonoid biosynthesis pathway, the research centre has designed several experiments with the aim at unrevealing the pathway's metabolites evolution over time.
To positively integrate the experimental information into new knowledge at the pathway level, the entire work has been divided into tasks:

- building of a computational model of the pathway;

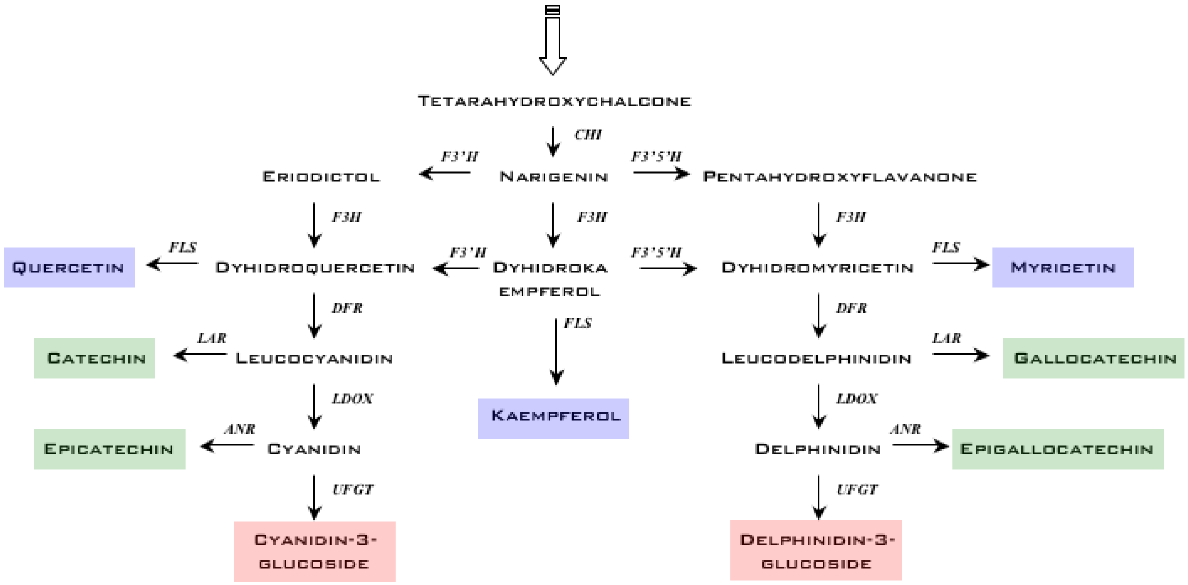


Figure 5.1: V.Vinifera pathway

- preliminary analysis on experimental data;
- inference on model parameters using experimental data.

5.1.1 Computational model of the flavonoid biosynthetic pathway

The knowledge about the V. vinifera general pathway for phenolics biosynthesis leading to flavonoids is reported in 5.1 and includes the involved metabolites and the gene coding for the enzymes involved in the reactions. A complete stochastic model for the pathway has been built by using the BlenX programming language (see A.1). BlenX modeling language lets us to describe the pathway, and thus the enzymatic reactions, by means of communications among species. More precisely, given the intrinsic characteristics of the language, there is not the need for specifying every species involved in the pathway, as it would be usually done by using other modeling techniques. All we need to specify is the initial substrate and the enzymes. The enzymatic reactions are then described as sequential state change from the initial substrate once the proper enzyme binds and communicates.

As for any modeling activity, the level of detail of a model should be related to the problem goals, and of course to the available theoretical and experimental knowledge. In this case, primary goal of the study was to identify possible differences in the dynamics of accumulation of several metabolites among plants. We have already spoken about the theoretical knowledge. Experimental evidences consisted instead in concentration levels for 9 metabolites at the extremes of the pathway (those highlighted in the picture). Data are single concentration measurements at berries maturation for 63 individuals (plants) of red berries.

Given the above considerations, during the development of the computational model we decided to regard at some working hypothesis.

We did not model the enzyme synthesis from their gene copies, while instead we considered a constantly available single molecule for every enzyme. While this assumption is clearly unrealistic, it helps in scaling down the complexity of the model given the complete lack of information on the enzymatic abundance and kinetics.

Still to preserve model complexity, we do not model and consider metabolites degradation. This hypothesis obviously influences the dynamics of metabolite accumulation.

Finally, in order to have a complete computational model, we require to indicate initial conditions for the system. Since there is no available information on the initial substrate, meaning that there is no indication for its plausible initial condition, we decided to consider a single molecule of the initial substrate and an external source of production for that metabolite which continuously synthesizes new molecule at a given rate. This aspect, together with the lack of degradation for the metabolites, produces a linear accumulation of metabolites during time. Moreover, every other metabolite in the pathway is considered to be not present in the system at its initial state.

5.1.2 Preliminary analysis on experimental data

The computational model described in the previous section cannot be used to represent pathway dynamics, yet. It lacks of all the kinetic parameters necessary to derive the time evolution of metabolites abundance. That is, we need to correlate the model with the experimental evidence.

Experimental data should be analyzed first. Statistical descriptive analyses are necessary to isolate potential strange observations (outliers) which could lead to erroneous conclusions, but also to provide proper guidance to following analysis or work.

Experimental data on the V. vinifera general pathway have been produced by the IASMA research centre, and they are about the concentration levels for 9 metabolites at the extremes of the pathway (those highlighted in Figure 5.1). Data have been collected in two years, 2007 and 2008 and they are metabolites concentration measurements at berries maturation for 63 individuals (plants) of red berries. We consider here the first 61 individuals leaving out the parental individuals (Syrah and Pinot Noir).

We initially investigate the variability of the data. Figure 5.2 shows the distribution of the measurements for every species in the two years. Concentration levels are in mg/Kg of berries skin. We can easily notice that values for the Delphinidin-3-glucoside metabolite concentration, over the 61 individuals, are much higher than the others, as it seems for its variability. This peculiarity is kept from 2007 to 2008, while a strong difference can be seen for the distribution of Cyanidin-3-glucoside in 2007 and 2008 (squared highlighting). Distribution of this metabolite in 2008 is narrower and average values are much lower denoting a very different behavior from

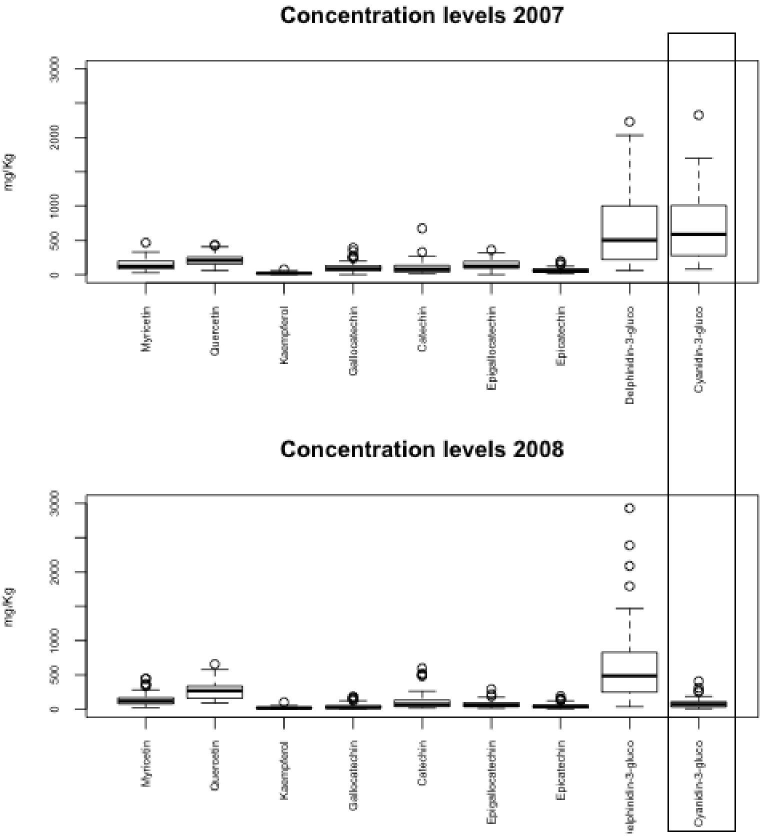


Figure 5.2: Distributions of the metabolites concentration levels, year 2007 and 2008

one year to the other. To extend the variability analysis, we calculate the coefficients of variation of metabolite concentrations (Table 5.1) for the two years. We can notice that Catechin and Gallocatechin present the highest variability. Furthermore, metabolites synthesized by the same enzymes present similar variability (they are grouped together in the table).

By comparing the coefficients of variations we can notice that values in 2007 are smaller than in 2008, meaning that concentration levels among the 61 plants are less dispersed around a mean value. To be noticed is also the coefficients of variations for Cyanidin-3-glucoside metabolite. From Figure 5.2 we highlighted a great difference in the two years, but their coefficients of variation do not reflect this huge difference. This is due to

Metabolite	CV_2008	CV_2007	Enzyme
Myricetin	0.66	0.59	
Quercetin	0.46	0.42	FLS
Kaempferol	0.77	0.62	
Gallocatechin	0.98	0.73	
Catechin	1.11	0.99	LAR
Epigallocatechin	0.73	0.53	
Epicatechin	0.79	0.61	ANR
Delphi3gluco	0.90	0.83	
Cyani3gluco	0.88	0.72	UFGT

Table 5.1: Coefficients of variation

the big difference which instead exists among the average value of the 61 plants. We moved from an average of 700 mg/kg in 2007 to 87 mg/Kg in 2008.

By looking at the differences in the mean of the other metabolites we can also notice that other metabolites present relevant changes. Average Gallocatechin concentrations have more than halved from 2007 to 2008 (107 mg/Kg in 2007 VS 44 mg/kg in 2008) and Epigallocatechin has halved (147 mg/kg in 2007 VS 75 mg/kg in 2008). Interestingly, all these three metabolites are placed in the right branch of the pathway.

The above analyses opened a question on the variability of the experimental information. In order to identify similar behaviour (or recurrent patterns) among individuals in the production of metabolites, we performed cluster analysis. With this analysis we are able to identify those plants having greatest interest for the study. In particular we are interested in those plants that are high producers of anthocyanins and in comparing them with low producers, extending the analysis over the available two years information.

We performed a hierarchical clustering based on euclidean distance of standardized data, and, by using the Ward's method for the aggregation we finally obtained dendrograms representing possible partitioning. The dendrogram is a tree which illustrates how individuals (on the leafs) are clustered together and in which order. At the beginning every plant constitutes a single group which is then consequently merged with others, based on similarity, ending up with a single cluster composed by all the individuals. Based on dendrograms and variability analysis, we identified proper data clusters. By plotting the distribution of the metabolites concentration over the different groups we are able to identify the distinguishing characteristics of each group. In particular, being us interested in anthocyanins, Figure 5.3 reports their group division (Delphinidin-3-glucoside and Cyanidin-3-glucoside). As it can be noticed, a good data partitioning has been found in both years with 6 groups. From the figure we can also easily extract the information required. From 2007 data, we pick as high anthocyanins producers, those plants clustered in group 4, while plants in group 3 will be

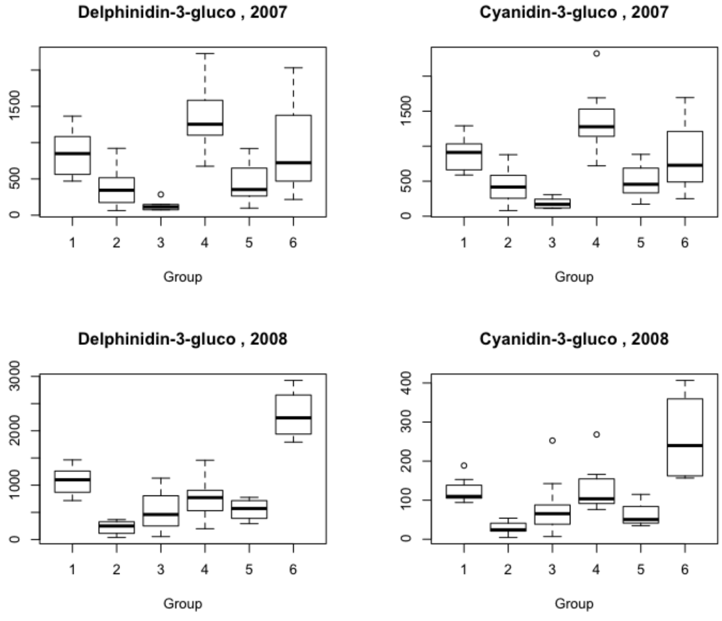


Figure 5.3: Clustering results

considered low producers. For what concerns 2008 data, group 7 contains high antocyanins producers while plants in group 2 are low producers. Table 5.2 summarizes the plants constituting each group.
We can further notice that some of the plants are preserved as high or low anthocyanins producers among 2007 and 2008, while none of them switched from high to low producers and viceversa.

	Year 2007	Year 2008
High producers	6, 13, 14, 15, 17 18, 20, 23, 26, 31 34, 41, 45, 48, 51	9, 15, 18, 20
Low producers	5, 25, 55, 57, 60, 61	2, 3, 7, 11, 16, 24, 30 33, 38, 39 42, 43, 44 46, 49, 54, 55, 57

Table 5.2: High and low anthocyanins producers

5.1.3 Parameter inference

As a starting point, we run the evolutionary inference framework, described in chapter 4 on the average of the experimental data.
Concentrations have been clearly translated into absolute values obtaining thus quantities representing number of molecules. To complete the input settings of the framework we fixed as halting criteria, a maximum number of iteration equal to 150. The cost function used is a sum of normalized squared error between the experimental evidence and simulated results. We also fixed a threshold for this objective function. We consider to have obtained good estimates when the average cost function value among the best 10 solutions is lower than a 5% distance from experimental evidence. This value is then fixed to 25.

The inference process took 65 minutes on an Apple Macbook, 2.0 GHz processor and 1 Gb RAM. Results are reported in Figure 5.4. The picture shows the best estimated solutions as well as experimental data with mean value used for the estimation (circle) and variability bars.

The estimated solutions fit well the mean values of experimental data.

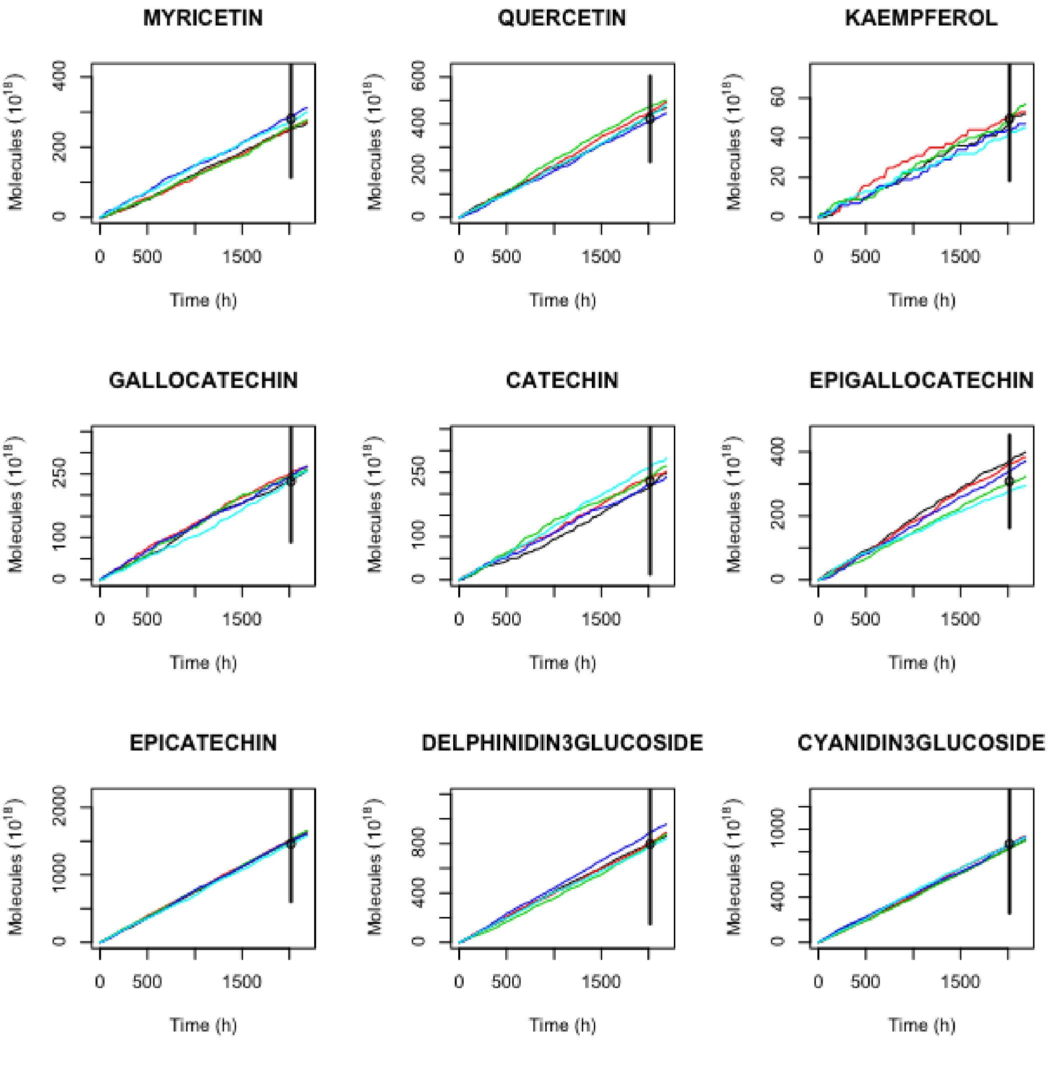


Figure 5.4: Estimation results for the average of experimental data

However, they will hardly describe the great variability the data exhibit. To give support to this hypothesis we used the approximate model checking tool we presented in Chapter 4. With this tool we can derive a formula codifying our hypothesis to be confirmed (that is, the estimated model cannot represent data variability).

We defined some thresholds for one of the metabolites we are mostly interested in, Cyanidin-3-glucoside. In particular, by looking at the experimental data, we took the 1^{st} and 3^{rd} quartile of its distribution (378 and 1248 molecules respectively). Twenty-five percent of the experimental data lies below the first quartile, and another 25% lies over the third quartile. Thus, by checking a formula that evaluates the probability of the model to obtain those values, we can conclude about the goodness of our model. If the estimated probabilities are very far from those exhibited by experimental data, then we will refuse the model.

Table 5.3 reports the result obtained by the approximate model checking tool on three different formulae. The first two formulae describe the probability to observe values lying below the first quartile or over the third quartile. In both cases at a 95% the true probability falls inside the interval $[0, 0.0099]$ with a point estimate equal to 0. In the last formula we took the threshold up to 800, and the probability for the model to obtain that level at the considered time falls in the interval $[0.00117, 0.01538]$ at a 95% confidence.

Formula (95% confidence), $\epsilon = 0.02$	Lower bound	Upper bound	Probability
Cyanidin-3-glucoside < 378 at $t = 2016$	0	0.0099	0
Cyanidin-3-glucoside > 1248 at $t = 2016$	0	0.0099	0
Cyanidin-3-glucoside < 800 at $t = 2016$	0.00117	0.01538	0.00425

Table 5.3: Checking the model probability to describe data variability

These results clearly indicate the inappropriate behavior of the model in describing data variability.

We decided then to proceed to infer pathway kinetics from clustered data reported in Table 5.2, as the goal is that of identifying the differences in kinetics for high and low anthocyanins producers.

We run again the evolutionary inference framework, for each of the 4 configurations of clusters. Low and high anthocyanis producers in 2007 and 2008.

In particular, we considered as target experimental information, the centroid of the cluster. From Table 5.4 we can notice that for every configuration considered we reached the predefined halting criteria based on the value of the cost function. Nonetheless in one case (namely high anthocyanins producers in 2008) the inference procedure took a longer time, requiring more iterations. Estimation time spans from 37 minutes to about 4.5 hours.

As we have seen when we have explained the evolutionary inference approach, being it a stochastic procedure, performance and results are different each time it has been ran. In this case we have obtained the expected results in terms of goodness of fit in every on the 4 times it has been used. By the way, just for the sake of clarity we run the estimation procedure for a second time on the high anthocyanins producers group in 2008 which has shown longer convergence time. After 95 iterations, and with an estimation time equal to 166 minutes we obtained an average cost function value equal to 24.4.

	Low 2007	Low 2008	High 2007	High 2008
Estimation Time (min)	37	91	87	260
Iterations	46	87	60	126
Cost function value (best 10 average)	24.8	24.4	24.1	24.8

Table 5.4: Estimation performance for low and high anthocyanins producers in 2007 and 2008

Table 5.4 summarizes quantitative information on the estimation problem. We move now to qualitatively analyze the model behavior by considering the estimated parameters. Figure 5.5 reports the best solutions identified by the evolutionary procedure with a cost function lower than 24.8 (best 10 average). Points at 2016 hours are the experimental data together with standard error bars, representing the variability within the group on the considered metabolites. Colored traces represent the dynamical behavior of the model with the estimated parameters.

It can be noticed that every metabolites abundance is well matched by

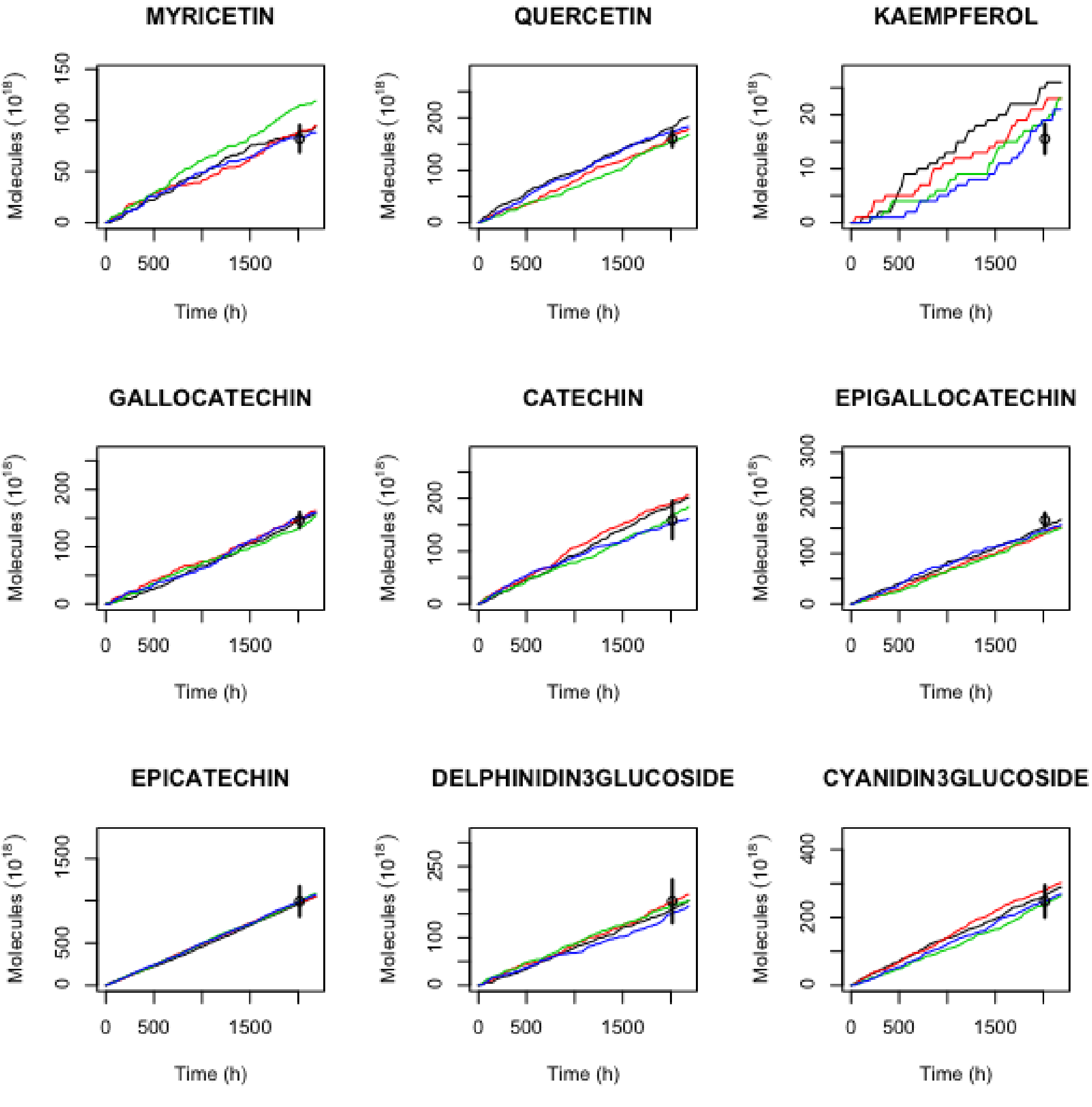


Figure 5.5: Estimation results for low anthocyanins producers in 2007

the model with a greater variability for Kaempferol metabolite, derived by its lower abundance which highlights the effects of stochastic fluctuations. We can also notice that, for those metabolites characterizing the group, namely anthocyanins (delphinidin-3-glucoside and cyanidin-3-glucoside), we have achieved very good matching, even considering group variability, i.e. stochastic fluctuations seems to properly describe group variability.
The identified good solutions for the estimation problem contains thus an estimate of each parameter of the pathway model. As we are in a stochastic setting, there is the possibility to obtain solutions that differ for one or more estimated parameters, still preserving the overall good fitting properties with respect to the experimental data. This is due to the stochastic effects that may arise during the time evolution of the simulated system.
To visualize the similarity of the solutions we make use of a radial (also known as spider) plot. Figure 5.6 reports the radial plot representing the five solutions with a cost function lower than 24.8 of the estimation of high producers in 2007. Polygons in the centre of the plot connect points representing values of each estimated parameter. Form this figure we can easily see that all the polygons are similar. The only small difference is represented by the value of $rEriodicF3h$ parameter which slightly differs in the three solutions. This parameter is the one governing the transformation of Eriodictol to Dyhidroquercetin with the act of enzyme F3h.
We can confirm this observation by performing a cluster analysis on the estimated solutions. Figure 5.7 plots the cluster dendrogram clearly highlighting the similarities of the solutions. We can notice the early aggregation of three of the five solutions, while the last two are aggregated to the previous cluster later, due to the difference in the previous parameter.

More generally, given the nature of the evolutionary procedure used for the inference, good solutions tend to be considered again, or with small changes, in the proceeding of the iterations. We then expect to have clus-

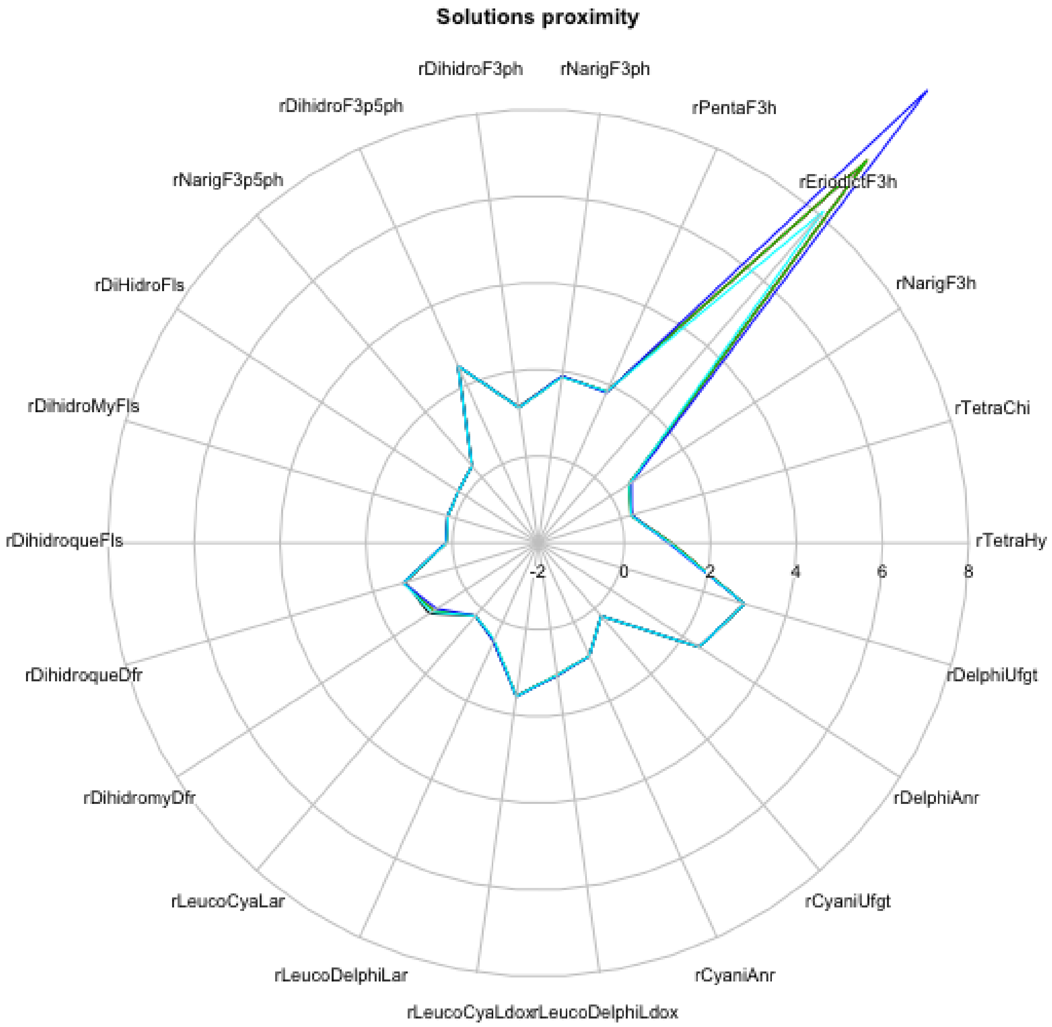


Figure 5.6: Radial plot of estimated solutions for high anthocyanins producers in 2007

ters of similar solutions in the pool of the best solutions reached, while the presence of a single solution in that pool may indicate that its good behavior has been the result of stochastic fluctuations.

The next step is to select the solutions from which derive the final parameters set. To do so we consider the similar solutions obtained by the cluster analysis and we average them. Table 5.5 contains the estimated parameters for the low anthocyanins producers in 2007. It contains the average value among the similar good solutions derived from the cluster analysis

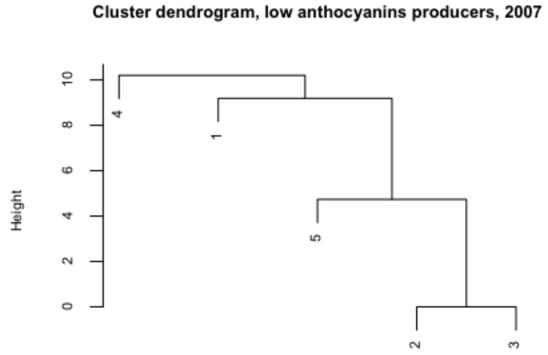


Figure 5.7: Cluster dendrogram of high anthocyanins producers (2007) estimated solutions

together with their standard deviation.

With the aim at focusing on the final goal of the project, we extended these analysis steps to the remaining three clusters of individuals considered, as to possibly identify strong differences in groups producing low or high amount of anthocyanins and/or year to year variations.

Plot in Figure 5.8 shows the abundance of metabolites for low anthocyanins producers in 2007 and 2008. As we can see some of the metabolites are more abundant in 2007 while others in 2008. More specifically all flavanols are more abundant in 2007 (black line) with a peak for epicatechin, while flavonols and anthocyanins are more abundant in 2008. From the comparison of the estimated parameters, we then expect to see higher values for those parameters governing the production of flavanols in the year 2007, while, in contrast, greater values for those parameters regulating anthocyanins production in the year 2008. Figure 5.8 reports this information in the form of a radial plot. Black polygon, connecting year 2007 estimated parameters, presents higher values with respect to those estimated for year 2008 data only for $rEriodictF3h$ and $rDihidroF3p5ph$. These are 2 parameters regulating the production of Dyhidroquercetin from Eriodictol and DyhidroMyricetin from Dyhidrokaempferol respectively. They are precursors of all flavanols.

Parameter	Mean value	Standard deviation
rTetraHy	1.08	1.64e-02
rTetraChi	0.24	1.69e-03
rNarigF3h	0.51	4.09e-04
rEriodictF3h	9.15	9.05e-01
rPentaF3h	1.83	6.53e-03
rNarigF3ph	1.87	9.01e-03
rDihidroF3ph	1.15	4.02e-04
rDihidroF3p5ph	2.48	3.52e-03
rNarigF3p5ph	0.35	3.15e-05
rDiHidroFls	0.20	1.21e-05
rDihidroMyFls	0.19	6.57e-04
rDihidroqueFls	0.14	3.49e-03
rDihidroqueDfr	1.26	5.68e-04
rDihidromyDfr	0.91	3.66e-02
rLeucoCyaLar	0.22	2.77e-04
rLeucoDelphiLar	0.47	1.06e-02
rLeucoCyaLdox	1.57	1.36e-03
rLeucoDelphiLdox	1.07	1.74e-03
rCyaniAnr	0.86	1.24e-03
rCyaniUfgt	0.22	9.57e-06
rDelphiAnr	2.44	7.21e-04
rDelphiUfgt	2.99	2.19e-06

Table 5.5: Estimated parameters for low anthocyanins producers in 2007

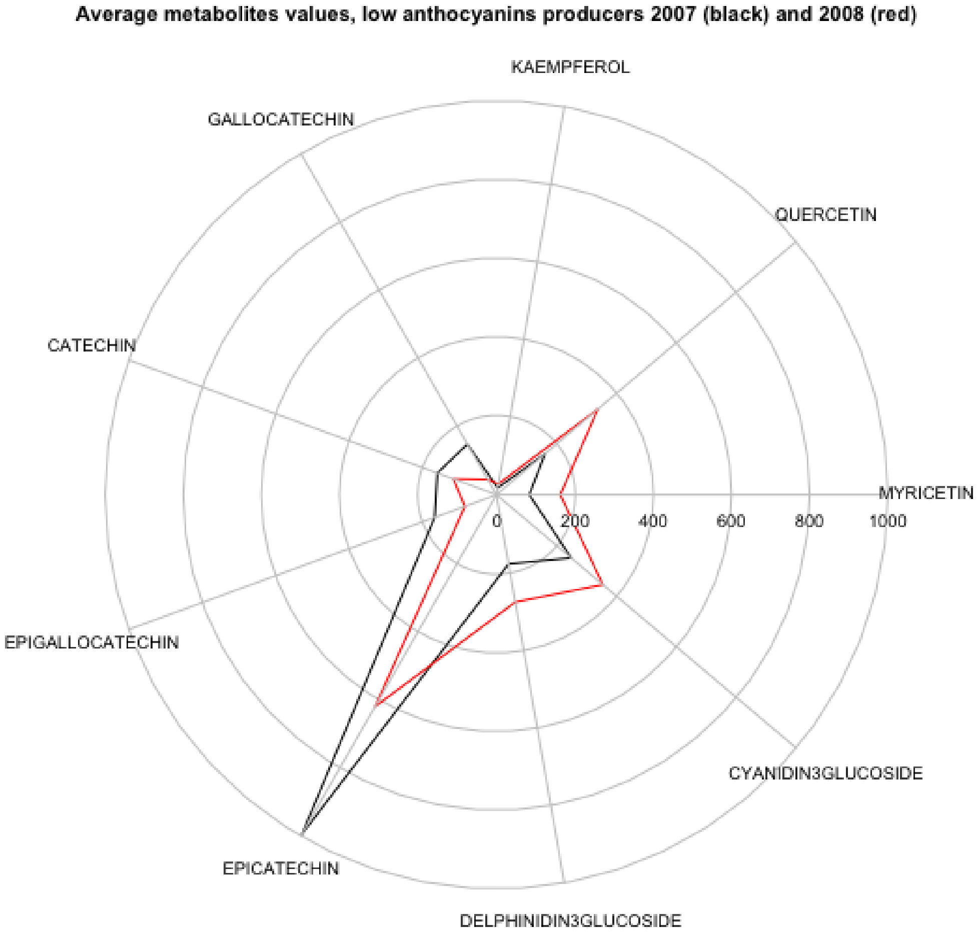


Figure 5.8: Radial plot of metabolites abundance in low anthocyaninns producers in 2007 and 2008

Red polygon, connecting year 2007 estimated parameters, presents instead higher values for all the parameters with higher differences for those parameters directly involved with the production of anthocyanins ($rDelphiUfgt$ for the production of Delphinidin-3-glucoside form Delphinidin) or immediate precursors ($rLeucoCyaLdox$ for the production of Cyanidin from Leucocyanidin). In this last case the subsequent pathway step for the production of Cyanidin-3-glucoside from Cyanidin present similar values among the two years (parameter $rCyaniUfgt$).

If we move the analysis towards the comparison of low and high antho-

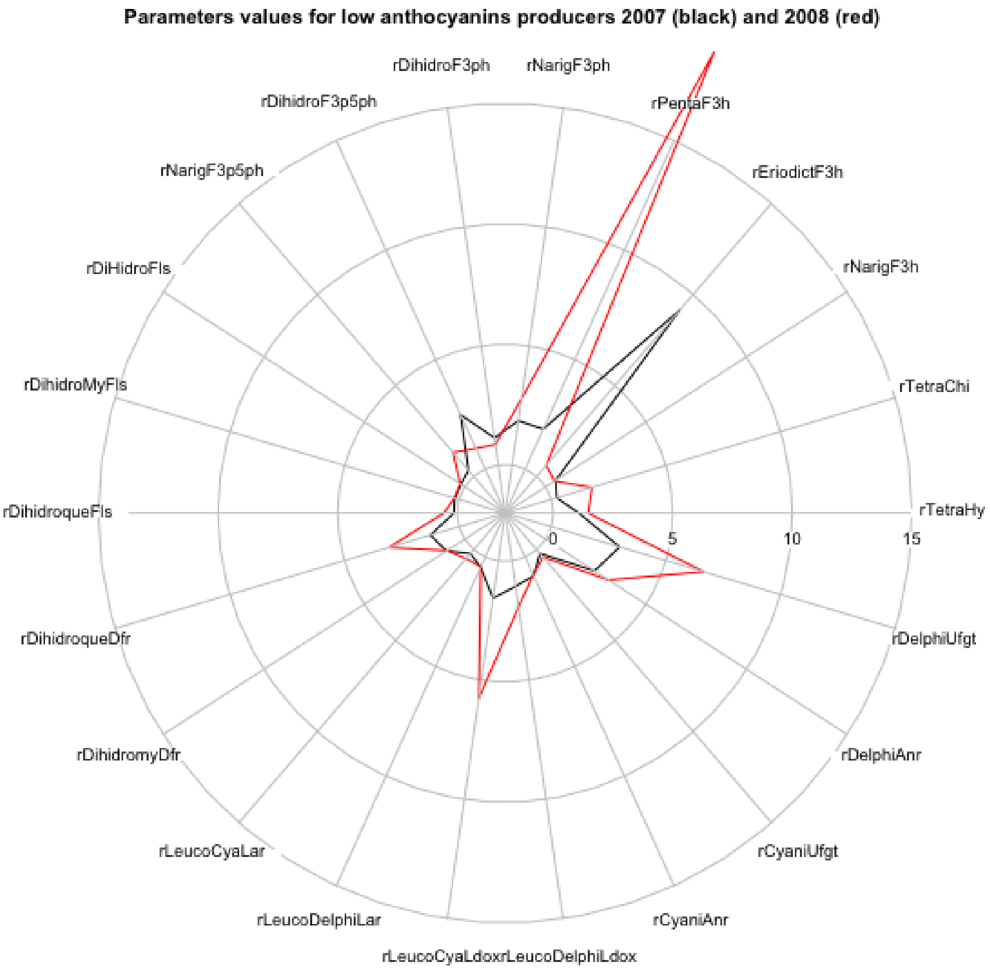


Figure 5.9: Radial plot of estimated parameters for low anthocyaninns producers in 2007 and 2008

cyanins producers in the same year we are able to identify which are the parameters that mostly affect the abundance variability. In particular in figure 5.10 we can see the great difference in the metabolites abundance among low and high anthocyanins producers in 2007. Obviously, the most significant difference is on anthocyanins (delphinidin-3-glucoside and cyanidin-3-glucoside) abundance, but also the remaining metabolites are more abundant in the high producers than in the low ones. This make us guessing that the kinetics is increased on the entire pathway and not

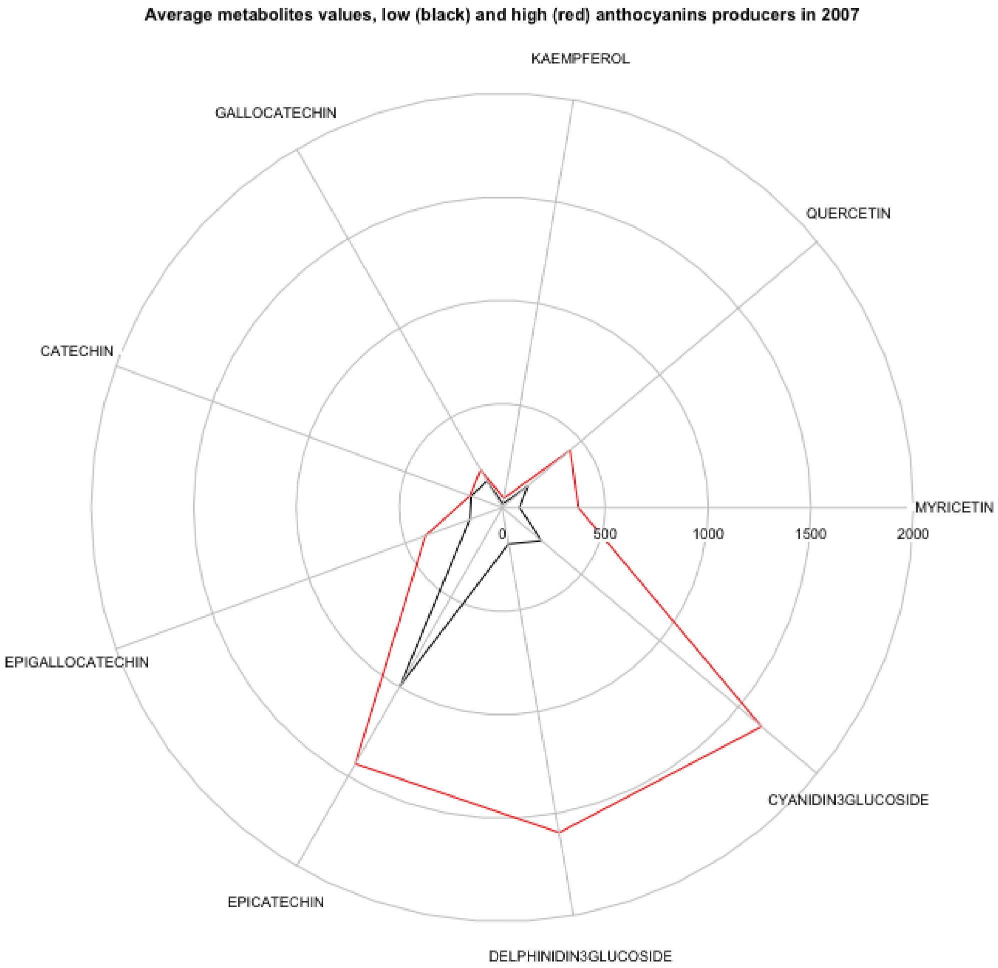


Figure 5.10: Radial plot of metabolites abundance in low and high anthocyaninns producers in 2007

only in the last steps of it, i.e. just when producing anthocyanins. In fact, by looking at the estimated parameters for low and high anthocyanins producers in 2007 and 2008 (figure 5.11 we can notice higher parameter values for most of the model parameter, but without a clear difference. The most interesting observation should be done in pointing out the remarkable difference in the initial parameters: $rTetraHy$ and $rTetraChi$. These are the parameters that regulate the production of the initial substrate (Tetrahydroxychalcone) from the external environment and its transformation into

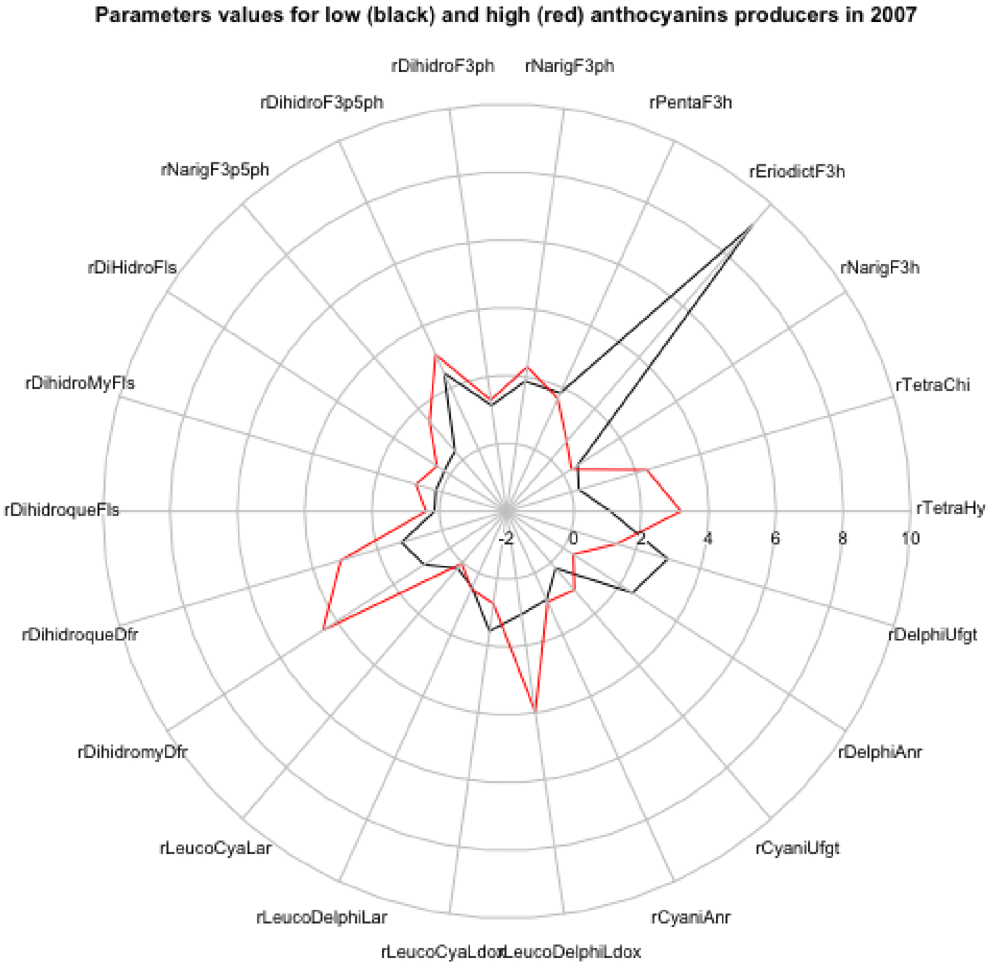


Figure 5.11: Radial plot of estimated parameters for low and high anthocyaninns producers in 2007

Narigenin respectively. These are the initial steps of the entire pathway. Greater values for these parameters represent the necessity for increasing the entire flux of metabolites to describe the overproduction of final metabolites (anthocyanins), as previously hypothesized.

5.1.4 Discussion

The flavonoids biosinthetic pathway has received great interest mostly due to the potential health benefits of some of its compounds. Experimentation

conducted at the IASMA centre in San Michele all'Adige - Trento, Italy made available concentration levels for part of the metabolites involved in the pathway measured on a number of plants.
The use of stochastic modeling has highlighted the different dynamical behavior among groups of plants. By estimating kinetics on pathway stochastic model we were able to identify those differences related to the production of anthocyanins, and specifically to differentiate low and high anthocyanins producers. Differences in the production of such metabolites cannot be described just by stochastic fluctuations, while there's the clear evidence of an enzymatic overactivity.
Clearly this conclusion should be validated with new experiments, and if confirmed, attention should be paid on an upper level, that of regulation.

5.2 Leishmaniasis disease progression

Leishmaniasis is recognized as one of the most important tropical diseases. It is present in three main forms: visceral leishmaniasis, the most aggressive and usually fatal when untreated; muco-cutaneous and cutaneous Leishmaniasis.
The disease is caused by a parasite belonging to the genus Leishmania and it is spread by the bite of a female sandfly to humans and animals. Desjeux [20] presents an extensive review of the disease, situation and new perspectives.
From a theoretical point of view, mathematical modeling has been used to study the evolution of the disease. Most of the attempted models deal with epidemiological settings [11, 1] or with the genome and metabolism features of the parasite [10, 57]. Only recently two integrated works were focused on the description of the host-parasite interaction, the first one [15] is an agent-based model of the immune response to Leishmania major

infection in mice. A still more recent work [38] presented a mathematical model based on GMA power law formalism to describe the disease evolution and the host-pathogen interaction in mice.

A joint collaboration with the Biochemical Technology Group of the Universidad La Laguna in Tenerife, Spain has let us to develop a stochastic model for the progression of the leishmania disease, based on their previous mathematical description.

For this case study, we present an integrated work that uses and connects the methodologies described in this thesis. After creating the stochastic model we used the evolutionary inference tool to estimate model parameters as to reproduce experimental data presented in the paper. We analyzed how parameters may affect particular model outputs. We aimed at highlighting the model nodes that play a central role in the disease development and we finally produce a quantitative information on the potentialities of disease reduction.

5.2.1 Leishmaniasis progression model

The theoretical model is the one presented in [38], which describes the interaction of four variables (here considered as species) and it is depicted in figure 5.12.

Species are Parasite Load on mouse internal organs, population of Lymphocytes and the immune response given by antibodies IgG1 and IgG2a. Each of the considered species displays an inbound and outbound flux representing respectively their production and degradation. Arrows stand for influence of a single species on one another, affecting either the production or degradation and possibly showing positive or negative effects.

To start with, once parasites have been inoculated they self proliferate (p_{14}) and enhance the immune system to produce lymphocytes (p_{11}). At the same time, the increase of parasite load leads to a decrease in lymphocytes

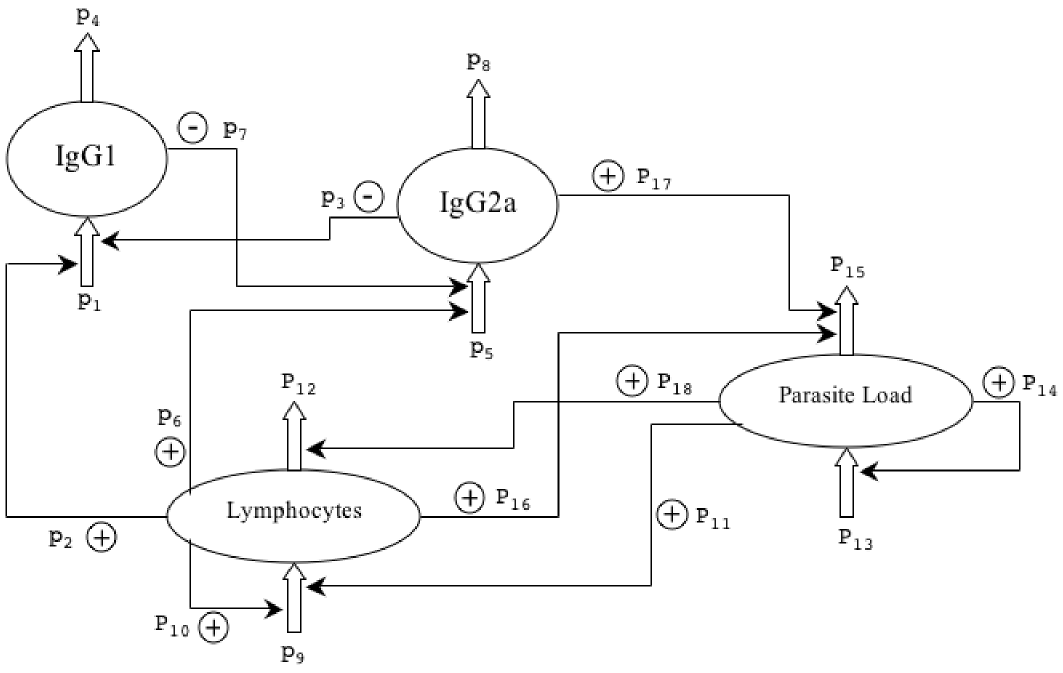


Figure 5.12: Leishmaniasis disease progression model

activity (p_{18}). Lymphocytes proliferate (p_{10}) and act as parasite suppressor (p_{16}). Simultaneously they enhance the synthesis of immunoglobulins IgG1 and IgG2a (p_2 and p_6 respectively). Being these immunoglobulins antagonists they have a negative influence on the other's generation (p_3 and p_7 respectively). Antibody IgG2a is able to identify parasites and thus enhancing parasite degradation (p_{17}), while this does not happen for IgG1. Finally all of species degradation follows mass action law (p_4, p_8, p_{12} and p_{15}).

BlenX stochastic model of the Leishmania development is reported in A.2 and follows an event based style of modeling, in which each possible step in the evolution of the system is completely governed by a set of actions. These events are related to the production and degradation of each single species in the model. They also describe the regulatory effect of each species on the others.

5.2.2 Inference

We initially consider the problem as presented in [38], trying thus to obtain parameters estimate for the same data.

For this case we used a cost function based on least squares estimator weighted for the number of different experimental points available for each variable. As for two of the four variables we have 13 data point, while for the remaining just 4, we equally weighted the contribution of each variable to the cost function.

Data presented in the paper have been initially considered "as they are", and thus without transforming concentrations into molecules abundances, mostly due to technical unfeasibility to retrieve such information. To avoid meaningless simulations and inference, data have been rescaled to properly work in the stochastic setting. They all have been multiplied by a factor of 1000.

The choice of initial interval values for model parameters, although not limiting the search space, should be carefully selected, as it represents a fundamental step for achieving a faster convergence. For instance, by considering each parameter initial interval equal to [0,1], convergence time is quite high, while using simple relationships among parameters, we are able to decrease it drastically.

More precisely, if we consider the system in a quasi-equilibrium, with a very naive analysis on the species involved, we are able to guess interval values for involved parameters in a more realistic way. As a simple rule of thumb, differences in the parameters order of magnitudes are estimated from experimental data on species abundances, and evaluating production and degradation reactions. For every parameter involving a first order reaction (i.e., species degradation, other species regulation) we adopted an initial value equal to the interval for a zero order reaction divided by the average abundance of the other species (i.e., its own average abundance in case of degradation, the average abundance of the regulating species in the latter case). This procedure is repeated for every parameter in the model (e.g., the interval for a second order reaction is scaled by the cross product of species involved.

In summary, we run the evolutionary inference procedure taking as initial parameter intervals [0,1] for every zero order reaction and [0,0.001] for every first order reaction. We set 200 as maximum number of iterations and derived a cost function threshold equal to 0.5, a value representing a 12% average deviation from experimental data.

We run the procedure 3 times, and in one case we reached the cost function threshold after 163 generations and a computation time of 15 minutes.

We can have a look at the qualitative behavior of the inferred solutions by analyzing figure 5.13. Dots represent experimental evidence averaged on mice population and organs infection. Bars stand for data variability and finally solid lines are the best solutions identified by the evolutionary procedure.The estimated solutions well describe the qualitative behavior of the system so we can now proceed to the analysis of the model.

5.2.3 Model analysis

This step is of primary importance for a deeper understanding of the disease progression. Before moving toward extensive model analysis we tried to reproduce the results presented in [38].

Our estimated stochastic model is consistent with the deterministic one presented in the paper for what concerns the influence of parameter p_{10} (g_6 in the paper).

Figure 5.14 shows the evolution of parasite load for different values of parameter p_{10}. Simulation length has been extended to 250 days (dashed

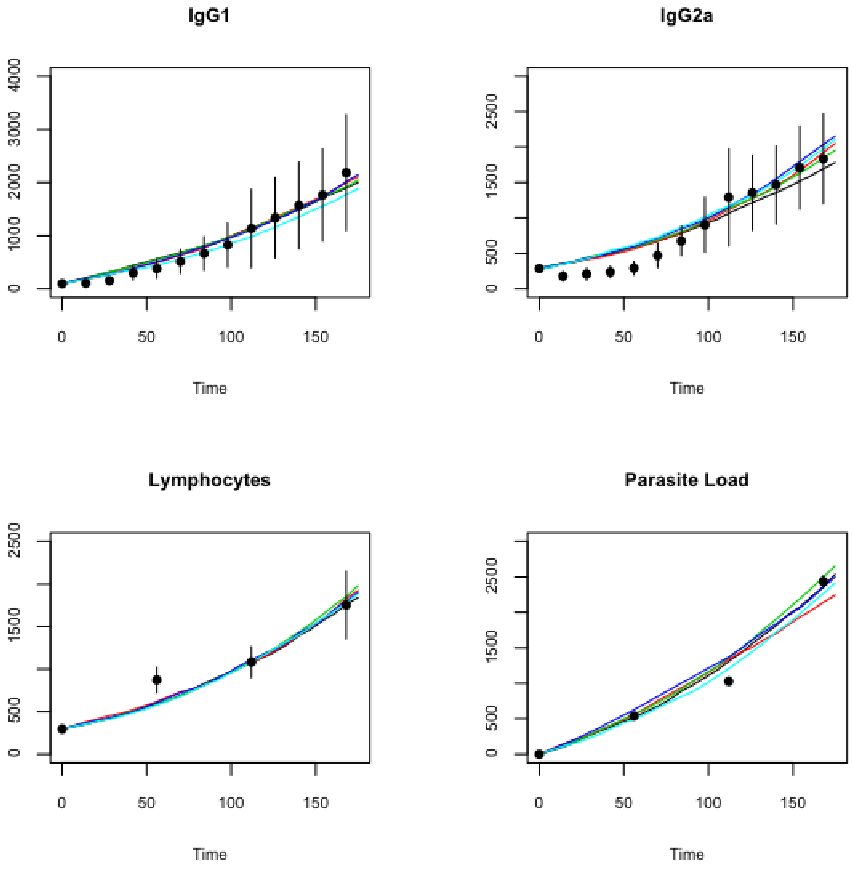


Figure 5.13: Best estimated solutions and experimental data with variability bars.

lines) as to taken into account possible longer effects. As it can be seen parasite load evolution start to decrease for $p_{10} = 0.03$ (which correspond to an increase of the parameter from the basal level of 3 times) even if the effect of parasite load reduction starts from about day 160. The effects are faster when the parameter is increased up to $p_{10} = 0.04$ and $p_{10} = 0.05$, respectively, 4 and 5 times its basal level.

For what concerns the influence of the parasite on its own proliferation, modeled by parameter p_{14}, our model predicts an increase of the overall parasite load when it is considered at higher level.

To make these predictions more plausible and to test other possible parameter effects, we used the tool for sweeping the model parameters to detect

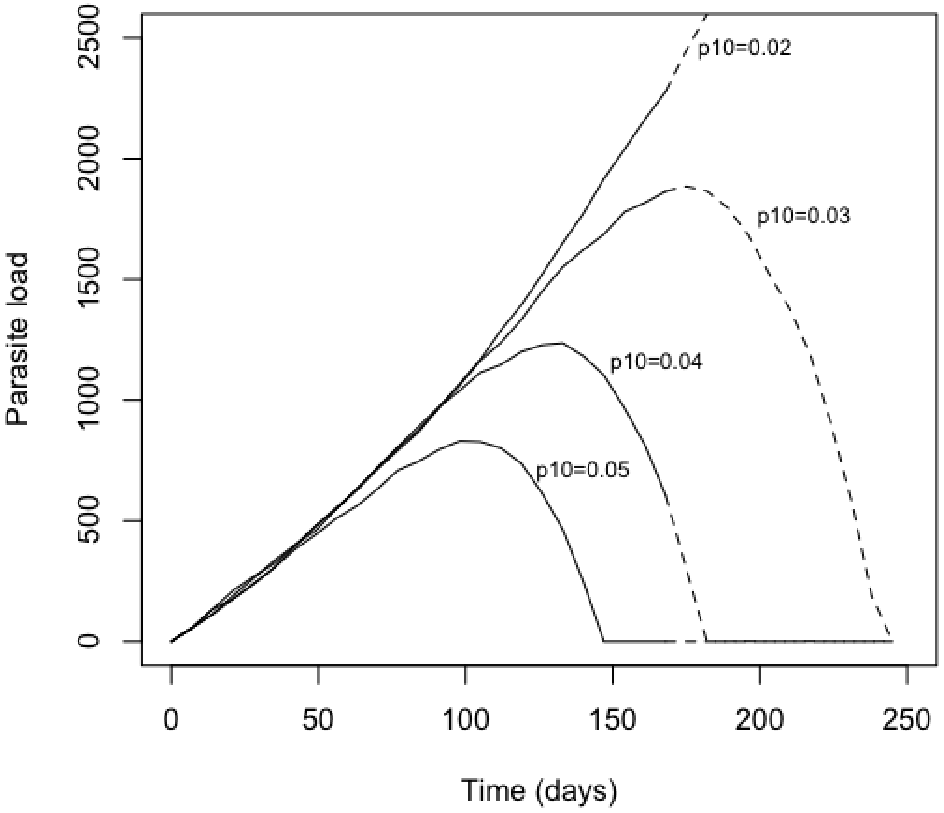


Figure 5.14: Parasite load evolution for different values of parameter p_{10}

those parameters which influence the most the parasite load.
We defined ranges of variation for every parameter involved in the model. These have been considered as proportional to the parameter value and more precisely, any parameter p_i varies in the interval $[0.5p_i, 5p_i]$. We are interested in identifying the possible effect of any of the parameters, and possibly higher order interaction among them producing a parasite load reduction. Given the number of involved parameters and and wideness of the intervals, we decided to consider a Latin Hypercube Sampling of the parameter space, with a sampling size $n = 1000$.
To accurately detect the effects of parameters' changes on the parasite load during disease progression, we considered two measures: the average para-

site load value and the final (at 24 weeks, or 168 days) parasite load value. They are simultaneously evaluated during the execution of the methodology. These aggregation measures are correlated. In fact, results clearly show this correlation when the two are plotted (Figure 5.15). As the goal is to identify possible parameter influence on the final parasite load (corresponding thus to healing), we focused on that measure.

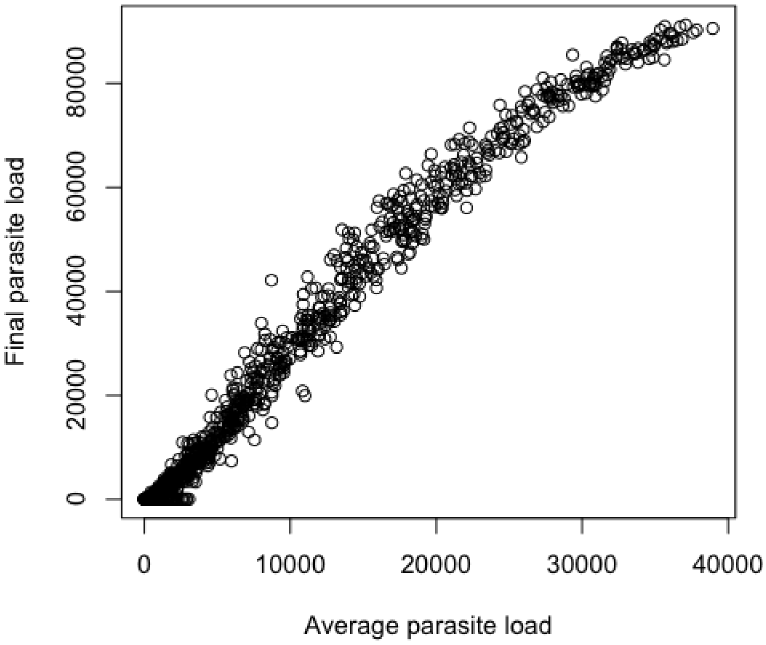


Figure 5.15: Scatterplot of average parasite load against final parasite load resulting from parameter sweeping

Results of multivariate analysis on the parameter sweeping reveal that other parameters influence the final parasite load (Table 5.6). In particular p_{17} which is responsible for the effect of immunoglubulin IgG2a on the identification and degradation of the parasites, influences the final parasite load with a negative sign. It means that its increase would result in a decrease of the final parasite load. Another negative effect, is the one given by p_6, the parameter regulating the positive effect of lymphocytes in the production of immunoglobulin IgG2a, while both positive are the effects of

Parameter	t-test p-value	Sign
p_6	$1.29e-11$	negative
p_{10}	$< 2e-16$	negative
p_{13}	$< 2e-16$	positive
p_{14}	$< 2e-16$	positive
p_{17}	$< 2e-16$	negative

Table 5.6: First order analysis of variance results for parameter influencing final parasite load

p_{13} and p_{14}, parameters related to parasite production are characterized by positive consequences on the final parasite load. Increasing the parameter value corresponds to increasing the final parasite load.

Still more interesting is the results when moving towards higher order interactions and effects (Table 5.7).

Higher order interactions are still related to those parameters previously identified. Nonetheless these results show us that varying simultaneously combinations of parameters, like p_{13} and p_{14} or p_{17} and p_6 may result in a stronger effect on the final parasite load.

Parameter	t-test p-value	Sign
p_{10}	$< 2e-16$	negative
p_{14}	$< 2e-16$	positive
$p_{13} : p_{14}$	$< 2e-16$	positive
$p_{17} : p_6$	$< 2e-16$	negative

Table 5.7: Higher order analysis of variance results for parameter influencing final parasite load

In order to make it clear this point highlighting the potential healing effects if multiple drugs would be developed, we performed a final in-silico experiment. We perturbed every parameter that was identified as influencing the final parasite load. Disturbance was set at a 15% magnitude from

each basal level, corresponding thus to a very low modification. Obviously, every parameter has been modified according to its sign in the influence of the parasite load (i.e p_{10}, p_{17} and p_6 have been increased, while p_{13} and p_{14} decreased) and 5 stochastic simulations have been performed.
Results, averaged among the 5 simulations are depicted in Figure 5.16. We can notice the decrease of parasite load starting from about day 170. The decrease in parasite load is then constant and in 3 simulations out of 5 it has reached zero before day 250.

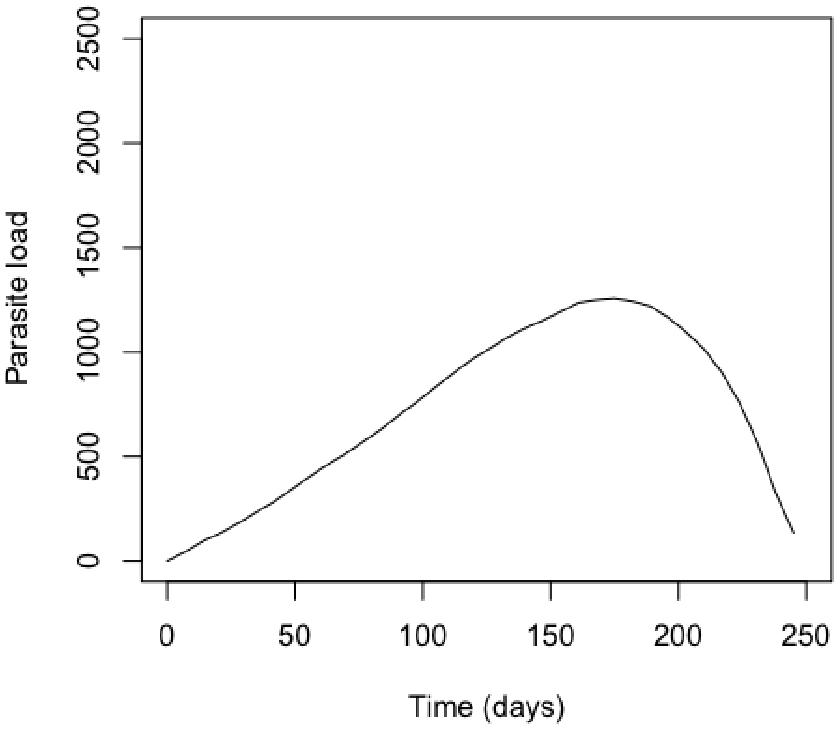


Figure 5.16: Parasite load for optimized model with a 15% change for every parameter involved in final parasite load

5.2.4 Discussion

In this section we studied the Leishmaniasis disease progression in mouse from a computational perspective. We initially modeled the system and identified a possible set of kinetic parameters describing its evolution over time. We then concentrated on the analysis of the model as it is of primary importance in identifying those parts that can mostly affect the disease progression.

The use of proper methodologies able to handle the complexity of real systems, both in terms of involved components and interactions, has let us to identify a set of parameters which are of great influence in the disease. We also were able to identify significant higher order interactions that can be vital for the study and development of possible new drugs.

We showed that considering most important parameters, even small changes may produce remarkable results.

We want also to point out the ranges of applications. With a so wide spectra of possible target parameters, the experimentalists are free to select those that more easily accessible from external drugs or treatment, or even those that have reduced impact on the disease host.

Chapter 6

Conclusion

Computer Science's unique way of describing systems through an algorithmic approach,is helping us understand how living systems behave. It offers tools and approaches to describe and model biological systems in a novel way, but, at the same time, a lot of work is still required in order to obtain a complete in-silico cycle of modeling and experimentation.

In this work we identified a couple of bottlenecks of this approach, namely the inference cycle for stochastic models and the subsequent model analysis.

We presented a new inference scheme that uses evolutionary computation techniques to evolve stochastic models, evaluated through simulations, towards solutions that match experimental data.The inference scheme presented is able to deal with cases where experimental information is sparse, noisy and even incomplete.

We also presented two approaches designed for efficiently tackling model analysis tasks within a stochastic setting. We developed an approximate method for the evaluation of logical properties of a stochastic model and a framework for efficiently generating and detecting peculiar behaviors of a given model by appropriately perturbing its parameter space.

All the above approaches and tools have been tested on a number of different test problems. Their application has then been extended to two real case studies, showing their potential to extract new knowledge from stochastic models of biological systems.

Process Algebra's stochastic approaches to Biology have proven to be a valid tool for describing living systems, but a lot of work is still required to provide a complete framework for appropriate model building and analysis. Further work, starting from the results presented in this thesis may help us reach that elusive goal. In particular, further study is needed to integrate the evolutionary inference scheme together with the model analysis. This would lead to a more general framework for knowledge inference able to describe systems for which only some qualitative characteristics are known and may not be quantitatively observable.

Bibliography

[1] N. Bacaer and S. Guernaoui. The epidemic threshold of vector-borne diseases with seasonality. *Journal of mathematical biology*, 53(3):421–436, 2006.

[2] J.C.M. Baeten and W.P. Weijland. *Process algebra.* Cambridge University Press Cambridge, 1990.

[3] P Ballarini, M Forlin, T Mazza, and D Prandi. Efficient Parallel Statistical Model Checking of Biochemical Networks. *EPTCS*, 14:47–61, 2009.

[4] P. Ballarini, T. Mazza, A. Palmisano, and A. Csikasz-Nagy. Studying irreversible transitions in a model of cell cycle regulation. *Electronic Notes in Theoretical Computer Science*, 232:39–53, 2009.

[5] E. Balsa-Canto, M. Peifer, J.R. Banga, J. Timmer, and C. Fleck. Hybrid optimization method with general switching strategy for parameter estimation. *BMC Systems Biology*, 2(1):26, 2008.

[6] GEP Box, WG Hunter, JF MacGregor, and J. Erjavec. Some problems associated with the analysis of multiresponse data. *Technometrics*, 15(1):33–51, 1973.

[7] R.J. Boys, D.J. Wilkinson, and T.B.L. Kirkwood. Bayesian inference for a discretely observed stochastic kinetic model. *Statistics and Computing*, 18(2):125–135, 2008.

[8] Lawrence D. Brown, T. Tony Cai, and Anirban Dasgupta. Interval estimation for a binomial proportion. *Statistical Science*, 16:101–133, 2001.

[9] Lawrence D. Brown, T. Tony Cai, and Anirban DasGupta. Confidence intervals for a binomial proportion and asymptotic expansions. *Annals of Statistics*, 30(1):160–201, 2002.

[10] A.K. Chavali, J.D. Whittemore, J.A. Eddy, K.T. Williams, and J.A. Papin. Systems analysis of metabolism in the pathogenic trypanosomatid Leishmania major. *Molecular Systems Biology*, 4(1), 2008.

[11] L.F. Chaves and M.J. Hernandez. Mathematical Modelling of American Cutaneous Leishmaniasis: incidental hosts and threshold conditions for infection persistence. *Acta Tropica*, 92(3):245–252, 2004.

[12] I. Chou, H. Martens, and E.O. Voit. Parameter estimation in biochemical systems models with alternating regression. *Theoretical biology and medical modelling*, 3(1):25, 2006.

[13] Edmund M Clarke, Orna Grumberg, and Doron A. Peled. *Model Checking*. MIT Press, 1999.

[14] P. Cos, TD Bruyne, N. Hermans, S. Apers, B. DVanden, and AJ Vlietinck. Proanthocyanidins in health care: current and new trends. *Current medicinal chemistry*, 11(10):1345–1359, 2004.

[15] G.M. Dancik, D.E. Jones, and K.S. Dorman. Parameter estimation and sensitivity analysis in an agent-based model of Leishmania major infection. *Journal of theoretical biology*, 262(3):398–412, 2010.

[16] A. Degasperi and S. Gilmore. Sensitivity analysis of stochastic models of bistable biochemical reactions. In *Formal Methods for Computational Systems Biology 2008*, volume 3082 of *LNCS*, pages 1–20, 2008.

[17] L. Demattè, C. Priami, and A. Romanel. BetaWB: modelling and simulating biological processes. In *SCSC: Proceedings of the 2007 summer computer simulation conference*, pages 777–784, 2007.

[18] L. Demattè, C. Priami, and A. Romanel. Modelling and simulation of biological processes in BlenX. *ACM SIGMETRICS Performance Evaluation Review*, 35(4):32–39, 2008.

[19] L. Demattè, C. Priami, and A. Romanel. The BlenX language: a tutorial. *Formal methods for computational systems biology: 8th International School on Formal Methods for the Design of Computer, Communication, and Software Systems, SFM 2008, Bertinoro, Italy, June 2-7, 2008: advanced lectures*, page 313, 2008.

[20] P. Desjeux. Leishmaniasis: current situation and new perspectives. *Comparative Immunology, Microbiology and Infectious Diseases*, 27(5):305–318, 2004.

[21] B. Di Ventura, C. Lemerle, K. Michalodimitrakis, and L. Serrano. From in vivo to in silico biology and back. *Nature*, 443(7111):527–533, 2006.

[22] E.D. Dolan, J.J. Moré, and T.S. Munson. Benchmarking optimization software with COPS 3.0. *Argonne National Laboratory Research Report*, 2004.

[23] D. Donaldson, R.;Gilbert. A monte carlo model checker for probabilistic ltl with numerical constraints. Technical report, University of Glasgow, Department of Computing Science, 2008.

[24] M. Forlin. vA computational design for high dimensional biochemical experiments. *6th Workshop on Simulation, St. Petersburg)*, 2009.

[25] M. Forlin, T. Mazza, and D. Prandi. Predicting the effects of parameters changes in stochastic models through parallel synthetic experiments and multivariate analysis. *2nd International Workshop on High Performance Computational Systems Biology (HiBi 2010)*, To appear, 2010.

[26] M. Forlin, I. Poli, D. De March, N. Packard, G. Gazzola, and R. Serra. Evolutionary experiments for self-assembling amphiphilic systems. *Chemometrics and Intelligent Laboratory Systems*, 90(2):153–160, 2008.

[27] N. Giglioli and A. Saltelli. SimLab 1.1, Software for Sensitivity and Uncertainty Analysis, tool for sound modelling. *Arxiv preprint cs/0011031*, 2000.

[28] DT Gillespie. Exact stochastic simulation of coupled chemical reactions. *J of Phys Chem*, 81(25), 1977.

[29] D.T. Gillespie. A rigorous derivation of the chemical master equation. *Physica A*, 188(3):404–425, 1992.

[30] D.E. Goldberg. *Genetic algorithms in search, optimization, and machine learning*. Addison-wesley, 1989.

[31] Thomas Hérault, Richard Lassaigne, and Sylvain Peyronnet. Apmc 3.0: Approximate verification of discrete and continuous time markov chains. In *QEST*, pages 129–130. IEEE Computer Society, 2006.

[32] A. Hoare, D. Gregan, and D.P Wilson. Sampling and sensitivity analyses tools (SaSAT) for computational modelling. *Theoretical Biology and Medical Modelling*, 5(1):4, 2008.

[33] H.G. Hunt and J.E. Hawkins. The rate of thermal isomerization of α-pinene and βpinene in the liquid phase. *Journal of the American Chemical Society*, 72:5618–5620, 1950.

[34] J.-P. Katoen, M. Khattri, and I. S. Zapreev. A Markov reward model checker. In *Quantitative Evaluation of Systems (QEST)*, pages 243–244, Los Alamos, CA, USA, 2005. IEEE Computer Society.

[35] J. Kennedy, R.C. Eberhart, et al. Particle swarm optimization. In *Proceedings of IEEE international conference on neural networks*, volume 4, pages 1942–1948. Perth, Australia, 1995.

[36] H. Kitano. Systems biology: a brief overview. *Science*, 295(5560):1662, 2002.

[37] M. Kwiatkowska, G. Norman, and D. Parker. Prism: Probabilistic model checking for performance and reliability analysis. *ACM SIGMETRICS Performance Evaluation Review*, 36(4):40–45, 2009.

[38] B.M. Langer, C. Pou, J.A. Hormiga, C. Gonzales-Alcon, B. Valladares, B. Wimmer, and N.V. Torres. Modelling of the Leishmaniasis infection dynamics. Novel application to the design of effective therapies. *submitted*, 2010.

[39] P. Lecca, A. Palmisano, C. Priami, and G. Sanguinetti. A new probabilistic generative model of parameter inference in biochemical networks. In *Proceedings of the 2009 ACM symposium on Applied Computing*, pages 758–765. ACM, 2009.

[40] H. Li, Y. Cao, L.R. Petzold, and D.T. Gillespie. Algorithms and software for stochastic simulation of biochemical reacting systems. *Biotechnology progress*, 24(1):56–61, 2008.

[41] C.J. Liao et al. A discrete version of particle swarm optimization for flowshop scheduling problems. *Computers & Operations Research*, 34(10):3099–3111, 2007.

[42] H. Lodish and S.L. Zipursky. Molecular cell biology. *Biochemistry and Molecular Biology Education*, 29:126–133, 2001.

[43] A.J. Lotka. *Elements of physical biology*. Williams & Wilkins company, 1925.

[44] S. Marino, I.B. Hogue, C.J. Ray, and D.E. Kirschner. A methodology for performing global uncertainty and sensitivity analysis in systems biology. *J. of Theoretical Biology*, 254:178–196, 2008.

[45] M.D. McKay, RJ Beckman, and WJ Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 42(1):55–61, 2000.

[46] R. Milner, J. Parrow, and D. Walker. A calculus of mobile processes, i. *Information and computation*, 100(1):1–40, 1992.

[47] C.G. Moles, P. Mendes, and J.R. Banga. Parameter estimation in biochemical pathways: a comparison of global optimization methods. *Genome research*, 13(11):2467, 2003.

[48] B. Novak and J.J. Tyson. *Cell Cycle Controls in: C. P . Fall et al., (Eds.), Computational Cell Biology*, pages 261–284. Springer, 2003.

[49] Walter W. Piegorsch. Sample sizes for improved binomial confidence intervals. *Computational Statistics and Data Analysis*, 46(2):309 – 316, 2004.

[50] A. Pnueli. A temporal logic of programs. In *Proc. of the 18th IEEE Symposium Foundations of Computer Science*, pages 46–57, 1977.

[51] P.K. Polisetty, E.O. Voit, and E.P. Gatzke. Identification of metabolic system parameters using global optimization methods. *Theoretical Biology and Medical Modelling*, 3(1):4, 2006.

[52] D. Prandi, C. Priami, and P. Quaglia. Communicating by compatibility. *Journal of Logic and Algebraic Programming*, 75(2):167–181, 2008.

[53] C. Priami. Algorithmic systems biology. *Communications of the ACM*, 52(5):80–88, 2009.

[54] C. Priami and P. Quaglia. Modelling the dynamics of biosystems. *Briefings in Bioinformatics*, 5(3):259–269, 2004.

[55] H. Radder. The philosophy of scientific experimentation: a review. *Automated Experimentation*, 1:2, 2009.

[56] S. Reinker, RM Altman, and J. Timmer. Parameter estimation in stochastic biochemical reactions. *IEE Proc.-Syst. Biol*, 153(4):168, 2006.

[57] A. Rochette, F. Raymond, J.M. Ubeda, M. Smith, N. Messier, S. Boisvert, P. Rigault, J. Corbeil, M. Ouellette, and B. Papadopoulou. Genome-wide gene expression profiling analysis of Leishmania major and Leishmania infantum developmental stages reveals substantial differences between the two species. *BMC genomics*, 9(1):255, 2008.

[58] M. Rodriguez-Fernandez, J.A. Egea, and J.R. Banga. Novel metaheuristic for parameter estimation in nonlinear dynamic biological systems. *BMC bioinformatics*, 7(1):483, 2006.

[59] M. Rodriguez-Fernandez, P. Mendes, and J.R. Banga. A hybrid approach for efficient and robust parameter estimation in biochemical pathways. *Biosystems*, 83(2-3):248–265, 2006.

[60] A. Saltelli, M. Ratto, S. Tarantola, and F. Campolongo. Sensitivity analysis for chemical models. *Chem. Rev.*, 7(105):28112827, 2005.

[61] M.A. Savageau. Biochemical systems theory: Operational differences among variant representations and their significance***. *Journal of theoretical Biology*, 151(4):509–530, 1991.

[62] SD Shih. The period of a Lotka-Volterra system. *Taiwanese J. Math*, 1:451–470, 1997.

[63] T. Sousa, A. Silva, and A. Neves. Particle swarm based data mining algorithms for classification tasks. *Parallel Computing*, 30(5-6):767–783, 2004.

[64] S. Suresh, PB Sujit, and AK Rao. Particle swarm optimization approach for multi-objective composite box-beam design. *Composite Structures*, 81(4):598–605, 2007.

[65] R. Team. R: A language and environment for statistical computing. *R Foundation for Statistical Computing Vienna Austria ISBN*, 3(10), 2008.

[66] M. Theis, G. Gazzola, M.F.I. Poli, M.M. Hanczyc, and M.A. Bedau. Optimal formulation of complex chemical systems with a genetic algorithm. In *ECCS06 online Proceedings (P193*. Citeseer, 2006.

[67] J.M. Thole, T.F.B. Kraft, L.A. Sueiro, Y.H. Kang, J.J. Gills, M. Cuendet, J.M. Pezzuto, D.S. Seigler, and M.A. Lila. A comparative evaluation of the anticancer properties of European and American elderberry fruits. *Journal of medicinal food*, 9(4):498–504, 2006.

[68] T. Tian, S. Xu, J. Gao, and K. Burrage. Simulated maximum likelihood method for estimating kinetic rates in gene expression. *Bioinformatics*, 23(1):84, 2006.

[69] G. Van Rossum, F.L. Drake, A. Kuchling, and Inc Books24x7. *Python tutorial.* Citeseer, 1999.

[70] R. Velasco, A. Zharkikh, M. Troggio, D.A. Cartwright, A. Cestaro, D. Pruss, M. Pindo, L.M. FitzGerald, S. Vezzulli, J. Reid, et al. A high quality draft consensus sequence of the genome of a heterozygous grapevine variety. *PLoS One*, 2(12), 2007.

[71] V. Volterra. Variazioni e fluttuazioni del numero dindividui in specie animali conviventi. *Mem. Acad. Lincei*, 2(31):113, 1926.

[72] D.J. Wilkinson. Stochastic modelling for quantitative description of heterogeneous biological systems. *Nature Reviews Genetics*, 10(2):122–133, 2009.

[73] E. B. Wilson. Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, 22:209–212, 1927.

[74] H. Younes, M. Kwiatkowska, G. Norman, and D. Parker. Numerical vs. statistical probabilistic model checking. *International Journal on Software Tools for Technology Transfer (STTT)*, 8(3):216–228, 2006.

Appendix A

BlenX models

A.1 Flavonoids pathway

A.1.1 The ".prog" file

```
[ steps = 91, delta = 24 ]

<<
   BASERATE:inf,
   CHANGE:inf,
   EXPOSE:inf,
   HIDE:inf,
   UNHIDE:inf
>>

let CHI : bproc = #(x,Chi)
[ rep x!(chi).nil ];

let F3H : bproc = #(x,F3h)
[ rep x!(f3h).nil ];

let FLS : bproc = #(x,Fls)
[ rep x!(fls).nil ];

let F3PH : bproc = #(x,F3ph)
[ rep x!(f3ph).nil ];
```

```
let F3P5PH : bproc = #(x,F3p5ph)
[ rep x!(f3p5ph).nil ];

let DFR : bproc = #(x,Dfr)
[ rep x!(dfr).nil ];

let LDOX : bproc = #(x,Ldox)
[ rep x!(ldox).nil ];

let LAR : bproc = #(x,Lar)
[ rep x!(lar).nil ];

let ANR : bproc = #(x,Anr)
[ rep x!(anr).nil ];

let UFGT : bproc = #(x,Ufgt)
[ rep x!(ufgt).nil ];



let TetraHy : bproc = #(y,TetrHy), #h(y1,yONE), #h(y2,yTWO)
[
  xx!() | rep xx?(). (  y?(ez).ez!().nil + y1?(ez1).ez1!().nil
                      + y2?(ez2).ez2!().nil  )

  |

  chi?().ch(y,NarigenErio).unhide(y1).ch(y1,NarigenDy).unhide(y2).
  .ch(y2,NarigenPenta).xx!().nil  |

  rep f3h?(). ( if(y1,NarigenDy) then ch(y,DyHidroque).ch(y1,DyHidroKae).
                  .ch(y2,DyHidroDyhidroMy).xx!().nil endif +
                if(y,ErioDyhidroque) then ch(y,DyHidroqueQue).unhide(y1).
                  .ch(y1,DyHidroqueLeo).xx!().nil endif +
                if(y,PentaDiHydroMy) then ch(y,DyHidroMyLeuco).unhide(y1).
                  .ch(y1,DyHidroMyRice).xx!().nil endif
               )  |
  rep f3ph?(). ( if(y,NarigenErio) then ch(y,ErioDyhidroque).ch(y1,yONE).
                  .hide(y1).ch(y2,yTWO).hide(y2).xx!().nil endif +
                 if(y,DyHidroque) then ch(y,DyHidroqueQue).ch(y1,DyHidroqueLeo).
                  .ch(y2,yTWO).hide(y2).xx!().nil endif
               )  |

  rep f3p5ph?(). ( if(y2,NarigenPenta) then ch(y,PentaDiHydroMy).
                   .ch(y1,yONE).hide(y1).ch(y2,yTWO).hide(y2).xx!().nil endif +
                   if(y2,DyHidroDyhidroMy) then ch(y,DyHidroMyLeuco).
                   .ch(y1,DyHidroMyRice).ch(y2,yTWO).hide(y2).xx!().nil endif
               )  |

  rep fls?(). ( if(y,DyHidroKae) then hide(y).hide(y1).hide(y2).
                   .xx!().nil endif +
                   if(y,DyHidroqueQue) then hide(y).hide(y1).xx!().nil endif +
                   if(y1,DyHidroMyRice) then hide(y).hide(y1).xx!().nil endif
               )  |

  rep dfr?(). ( if(y1,DyHidroqueLeo) then ch(y,LeucoCate).
                   .ch(y1,LeucoCya).xx!().nil endif +
                   if(y,DyHidroMyLeuco) then ch(y,LeucoDelphi).
                   .ch(y1,LeucoGallo).xx!().nil endif
               )  |

  rep ldox?().( if(y1,LeucoCya) then ch(y,CyaniEpica).
                  .ch(y1,CyaniFla).xx!().nil endif +
                  if(y,LeucoDelphi) then ch(y,DelphiFla).
                  .ch(y1,DelphiEpigallo).xx!().nil endif
               )  |

  rep lar?().( hide(y).hide(y1).xx!().nil
               )  |

  rep ufgt?().( if(y1,CyaniFla) then ch(y,Cyanidin3Gluco).
                 .hide(y).hide(y1).xx!().nil endif +
                 if(y,DelphiFla) then ch(y,Delphi3Gluco).
                 .hide(y).hide(y1).xx!().nil endif
               )  |

  rep anr?().( if(y,CyaniEpica) then ch(y,Epicatechin)
                 .hide(y).hide(y1).xx!().nil endif +
                 if(y1,DelphiEpigallo) then hide(y).
                 .hide(y1).xx!().nil endif
               )

];

when(TetraHy :: rate(rTetraHy)) new;

run 1 CHI || 1 F3H || 1 FLS || 1 F3PH || 1 F3P5PH ||
    1 DFR || 1 LDOX || 1 LAR || 1 ANR || 1 UFGT || 1 TetraHy
```

A.1.2 The ".types" file

```
{
   Chi, F3h, Fls, F3ph, F3p5ph, Dfr, Ldox,  Lar, Anr, Ufgt, TetrHy,
   yONE, yTWO, NarigenDy, NarigenErio, NarigenPenta,
   DyHidroKae, DyHidroque, DyHidroDyhidroMy, ErioDyhidroque,
   DyHidroqueLeo, DyHidroqueQue, LeucoCya, LeucoCate,
   PentaDiHydroMy, DyHidroMyLeuco, DyHidroMyRice,
   LeucoDelphi, LeucoGallo, CyaniFla, CyaniEpica, DelphiFla,
   DelphiEpigallo, Epicatechin, Cyanidin3Gluco, Delphi3Gluco
}

%%

{
 (Chi,TetrHy,rate(rTetraChi)),
 (F3h,NarigenDy,rate(rNarigF3h)),
 (F3h,ErioDyhidroque,rate(rEriodictF3h)),
 (F3h,PentaDiHydroMy,rate(rPentaF3h)),
 (F3ph,NarigenErio,rate(rNarigF3ph)),
 (F3ph,DyHidroque,rate(rDihidroF3ph)),
 (F3p5ph,NarigenPenta,rate(rNarigF3p5ph)),
 (F3p5ph,DyHidroDyhidroMy,rate(rDihidroF3p5ph)),
 (Fls,DyHidroKae,rate(rDiHidroFls)),
 (Fls,DyHidroMyRice,rate(rDihidroMyFls)),
 (Fls,DyHidroqueQue,rate(rDihidroqueFls)),
 (Dfr,DyHidroqueLeo,rate(rDihidroqueDfr)),
 (Dfr,DyHidroMyLeuco,rate(rDihidromyDfr)),
 (Lar,LeucoCate,rate(rLeucoCyaLar)),
 (Lar,LeucoGallo,rate(rLeucoDelphiLar)),
 (Ldox,LeucoCya,rate(rLeucoCyaLdox)),
 (Ldox,LeucoDelphi,rate(rLeucoDelphiLdox)),
 (Anr,CyaniEpica,rate(rCyaniAnr)),
 (Anr,DelphiEpigallo,rate(rDelphiAnr)),
 (Ufgt,CyaniFla,rate(rCyaniUfgt)),
 (Ufgt,DelphiFla,rate(rDelphiUfgt))
}
```

A.1.3 The ".func" file

```
let rTetraHy :  const = 1.64;
let rTetraChi : const = 1.64;
let rNarigF3h : const = 0.547;
let rEriodictF3h : const = 0.547;
let rPentaF3h : const = 0.547;
let rNarigF3ph : const = 0.547;
let rDihidroF3ph : const = 0.43;
let rDihidroF3p5ph : const = 0.525;
let rNarigF3p5ph : const = 0.547;
let rDiHidroFls : const = 0.022;
let rDihidroMyFls : const = 0.135;
let rDihidroqueFls : const = 0.264;
let rDihidroqueDfr : const = 0.714;
let rDihidromyDfr : const = 0.506;
let rLeucoCyaLar : const = 0.124;
let rLeucoDelphiLar : const = 0.0425;
let rLeucoCyaLdox : const = 0.59;
let rLeucoDelphiLdox : const = 0.463;
let rCyaniAnr : const = 0.537;
let rCyaniUfgt : const = 0.053;
let rDelphiAnr : const = 0.029;
let rDelphiUfgt : const = 0.39;
```

A.2 Leishmaniasis disease progression

A.2.1 The ".prog" file

```
[ steps = 25, delta = 7 ]

<< BASERATE:inf >>


let IgG1 : bproc = #(w,bs0)
           [ nil];

let IgG2a : bproc = #(e,bs1)
           [ nil];

let IE : bproc = #(r,bs2)
           [ nil];

let PL : bproc = #(t,bs3)
           [ nil];


when (IgG1  :: f_bornIgG1) new;
when (IgG1  :: rate(p4)) delete;

when (IgG2a  :: f_bornIgG2a) new;
when (IgG2a  :: rate(p8)) delete;

when (IE  :: f_bornIE) new;
when (IE  :: f_deathIE) delete;

when (PL  :: f_bornPL) new;
when (PL  :: f_deathPL) delete;
when (IgG1  :: f_IgG1_control) delete;
when (IgG2a :: f_IgG2a_control) delete;
when (IE  :: f_IE_control) delete;
when (PL  : : f_PL_control) delete;


run 18 IgG1 || 22 IgG2a || 1000 IE || 0 PL
```

A.2.2 The ".types" file

```
{bs0, bs1, bs2, bs3 }

%%
{ (bs0, bs1, inf)}
```

A.2.3 The ".func" file

```
let p1 : const = 0.003331426;
let p2 : const = 0.056690934;
let p3 : const = 0.007130887;
let p4 : const = 0.006977599;
let p5 : const = 0.094012998;
let p6 : const = 0.169765354;
let p7 : const = 0.033729534;
let p8 : const = 0.079018305;
let p9 : const = 0.09848451;
let p10 : const = 0.064639724;
let p11 : const = 4.79;
let p12 : const = 0.053664816;
let p13 : const = 0.513853473;
let p14 : const = 0.110547994;
let p15 : const = 0.000129843;
let p16 : const = 0.001430713;
let p17 : const = 0.001402321;
let p18 : const = 0.10;
let f_bornIgG1 : function = p1 + p2 * |IE| - p3 * |IgG2a| ;
let f_bornIgG2a : function = p5 + p6 * |IE| - p7 * |IgG1| ;
let f_bornIE : function = p9 + p10 * |IE| + p11 * |PL| ;
let f_bornPL : function = p13 + p14 * |PL| ;
let f_deathPL : function = p15 * |PL| + p16 * |IE| + p17 * |IgG2a|;
let f_deathIE : function = p12 * |IE| + p18 * |PL|;

let IgG1_limit : const = 10000;
let f_IgG1_control : function = ( p2 / IgG1_limit ) * |IgG1| * |IgG1|;

let IgG2a_limit : const = 10000;
let f_IgG2a_control : function = (p6 / IgG2a_limit) * |IgG2a| * |IgG2a|;

let IE_limit : const = 10000;
let f_IE_control : function = (p10 / IE_limit) * |IE| * |IE|;

let PL_limit : const = 10000;
let f_PL_control : function = ( p14 / PL_limit) * |PL| * |PL|;
```