Identification of cell signaling pathways based on biochemical reaction kinetics repositories

Gustavo Estrela de Matos

Text presented
to
Institute of Mathematics and Statistics
of the
University of São Paulo
For
The Qualification Exam Of Master Of Science

Field of knowledge: Computer Science

Advisor: Dr. Marcelo da Silva Reis

Center of Toxins, Immune-Response and Cell Signaling (CeTICS)

Special Laboratory of Cell Cycle, Butantan Institute

During the development of this work the author received financial support of FAPESP.

São Paulo, May 13, 2019

Abstract

Cell signaling pathways are composed of a set of biochemical reactions that are associated with signal transmission within the cell and its surroundings. Traditionally, these pathways are identified through statistical analyses on results from biological assays, in which involved chemical species are quantified. However, once generally it is measured only a few time points for a fraction of the chemical species, to effectively tackle this problem it is required to design and simulate functional dynamic models. Recently, it was introduced a method to design functional models, which is based on systematic modifications of an initial model through the inclusion of biochemical reactions, which in turn were obtained from the interactome repository KEGG. Nevertheless, this method presents some shortcomings that impair the estimated model; among them are the incompleteness of the information extracted from KEGG, the absence of rate constants, the usage of sub-optimal search algorithms and an unsatisfactory overfitting penalization. In this project, we propose a new methodology for identification of cell signaling pathways, which will make use of a myriad of public interactome and biochemical reaction kinetics repositories to deal with the incompleteness of a priori information. Moreover, we will use optimal algorithms for model selection, as well as more effective cost functions for overfitting penalization. The new methodology will be tested on artificial instances and also on cell signaling pathways identification in our case study, the Y1 mouse adrenocortical tumor cell line. (AU)

Contents

Chapter 1

Introduction

Cell signaling pathways are cascades of chemical interactions that allow the communication between the cell environment and the cell itself. These pathways are also able to regulate many cell functions, including DNA replication, cell division and cell death. We can observe the functioning of signaling pathways as a mechanism that can conform the cell behavior with signals that come from the environment conditions in which the cell is placed. The studies of cell signaling pathways can lead to determining how cells can respond to different stimuli; for instance, with the studies of signaling pathways activated by a chemical species, one could determine how an unhealthy cell would respond to a drug containing this species.

It is possible to construct mathematical models to represent a set of chemical reactions and consequently a signaling network. One approach on the modeling of those interactions is based on the law of mass action. This law proposes that the rate of a chemical reaction is proportional to the product of reactants concentrations, i.e. we can calculate the concentration change rate of a species in an interaction by calculating the product of reactants concentrations, up to a multiplying constant. If we consider the set of interactions of a signaling pathway, we can then come up with a system of ordinary differential equations (ODEs) that can model the dynamics of the concentration of each chemical species from the pathway. Generally, these systems are complex and cumbersome, if not impossible, to be solved analytically, therefore we resort on computational tools that apply numerical methods to approximate solutions of these systems.

In this work, we are interested in computational models that can reproduce the behavior of signaling networks, comparing experimental measures—generally based on Western blot data—to simulated results. The figure 1.1 shows a set of interactions as well as parameters of a model of a signaling network. To create computational models that are able to simulate the behaviour of a signaling pathway, two main tasks need to be accomplished.

The first task one must complete to create a model that is able to simulate a signaling pathway is to determine a set of interactions that will be considered in the ODE system. Searching for pathway maps on the Kyoto Encyclopedia of Genes and Genomes (KEGG) [KG00] is a good start for this task. The KEGG PATHWAY Database provides manually drawn diagrams that represent signaling networks created with experimental evidences. However, it is possible that there is no pathway on KEGG that is able to correctly represent the biological experiment of interest; for those situations, it is necessary to modify the pathway by adding or removing interactions. One might reason that we should use as many interactions as we can to get a better simulation, however, this usually implies in poor or computationally infeasible models because of two reasons: first, complex models will require more time for a numerical solution computation, which may be infeasible due to limited computational resources; and second, when considering many interactions, we are also placing many parameters (multiplying

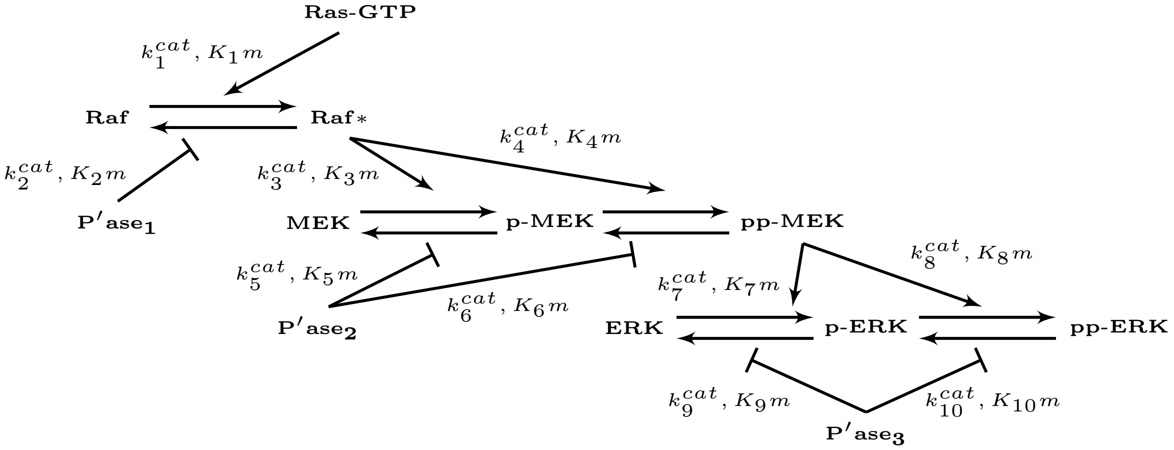


Figure 1.1: The above diagram show a hypothesis for a signaling pathway that flows through Raf-MEK-ERK cascade. Names in bold represent chemical species. Names in italic represent parameters of the ordinary differential equation of each interaction.Horizontal arrows represent phosphorylation when directed from left to right or dephosphorylation when directed in the opposite direction. Other arrows represent positive feedback if they are directed downwards or negative feedback otherwise. Original image of Marcelo S. Reis et al. (2017) [Rei+17a].

constants of the differential equations) on the model, and finding appropriate values for them becomes harder as we increase the number of parameters.

The second task is to find values for all the model parameters. There are two approaches for this task, one can either fetch values for these constants from the literature or one can find values that makes the model output approximate the experimental observations. For the first approach, repositories such as BioModels [LN+06] can be used; for the second approach, statistical and optimization methods can be used. For optimization, it is necessary to define a metric that can evaluate how close the parameters brings the model output to the experimental observation so that it is possible to search for the optimal parameter in the parameter space. Statistical inference, in the other hand, will usually try to maximize some likelihood function (find parameters that makes the data more likely to happen) on a more classical approach, while in a Bayesian approach the goal is usually to compute some posterior distribution for the model parameters (the probability of parameter values given the experimental observation).

After completing both tasks, however, as we mentioned before, we might still not have found a pair of model and parameter values that fairly approximates the biological experiment of interest. That could indicate that the set of chemical interactions chosen for the model is incomplete or has interactions that are not relevant for the biological experiment. Therefore, it is desirable to construct a systematic method of modifying the set of chemical reactions of the model in order to find a good set to represent the signaling network.

With the title “A method to modify molecular signaling networks through examination of interactome databases” [Wu15] Lulu Wu presented in her masters dissertation a methodology to systematically modify computational models of signaling networks to better simulate biological experiments. Starting with a model that does not approximate well the biological data, this methodology proposes to add to the model topology a set of chemical interactions that are relevant for the biochemical experiment, and consequently approximates the model simulation to experimental data. This set of interactions is a subset of interactions from a database created by Lulu Wu, joining information from many static maps of signaling networks available on KEGG. The choice of this subset can be modeled as a combinatorial optimization problem, the feature selection problem, in which the search space is the set of all possible subsets of interactions (features) to be added. The cost function of this problem, however, is not as simple to define as the search space. Note that points from the search space do not fully define models, but only the topology of the model, i.e. the set of interactions, so there is still need to define parameter values to produce a simulation. Therefore, to analyze the quality of a model, the cost function must take into account the set of values for the model parameters. As an example, we could define the cost as the minimum distance between experimental and model measures considering all possibilities of parameter values; however, unfortunately, finding the minimizing parameter values is a hard problem.

Since this is a hard problem, the method presented by Lulu Wu implements a heuristic version of this cost function, moreover, the algorithm used to traverse the search space is also a heuristics. The cost function heuristics is based on a Simulated Annealing procedure that searches for a set of parameter values trying to minimize (as much as possible) the distance between model and experimental measures. The best found distance is then considered as the cost of the model. The size of the search space is exactly the number of all possible subsets of interactions to be added, and this number grows exponentially on the number of interactions from the database. That explains why Lulu Wu used a heuristic to traverse the search space. This heuristic is based on the greedy algorithm called Sequential Forward Selection (SFS) [Whi71]. The heuristic implemented by Lulu Wu selects a fixed number of interactions from the database and then creates candidate models by adding to the current solution the respective interaction; then, after evaluating the cost function for each model, the algorithm moves to the best candidate.

The results presented on Lulu Wu's dissertation show that the method is useful when there are only a few differences between the starting model and a model that closely approximates the biological experiment. This limitation could be explained by the intrinsic difficulty of the problem, which demands fitting complex models with few experimental data; however, we would like to highlight three aspects of the work that contributes to its limitations. The first aspect is that the constructed database could be more nearly complete, adding information from other interactome databases, such as STRING [Szk+10], and also by adding information about model parameters, i.e. chemical reaction constants, that are available in other databases, e.g. the SABIO-RK [Wit+11] database. Second, the search algorithm used to modify the models can only add interactions, therefore, if the algorithm starts with (or add along the search) a spurious interaction on the topology, then the algorithm will not be able to “regret” that interaction even though there might be similar solutions without it with better fit. Third, the cost function does not include a proper penalization of complex models; the used penalization is based on a execution time limit on the simulated annealing procedure, implying on a random penalization for more complex models, which typically demand more execution time. Without a proper penalization, the algorithm is doomed to select overly complex models that, even with a good fit to the experimental data, are not likely to reproduce the same experiment conduced with any kind of perturbation to the biological environment or to the data collected.

We propose on this project to create a new method for modifying models of signaling networks, based on the work of Lulu Wu, and including possible solutions to the three aspects mentioned on the last paragraph. To the first aspect, we propose to create a database that includes interactions informations gathered from KEGG and STRING, and also that includes reaction constants values, which can be fetched from SABIO-RK, Brenda [Sch04] or BioNumbers [Mil+09]. To the second aspect, we propose to create new search algorithms that are more general than SFS, and to test and compare these new algorithms we intend to use the featsel framework [Rei+17b]. For the last aspect, about the cost function, we intend to use Bayesian approaches to rank models [VG07] based on the likelihood of them to reproduce the observed data; if we say M is a model with parameter space Θ and D is a set of observations, then we would like to estimate

$$p(D|M) = \int_{\theta \in \Theta} p(D|\theta, M)p(\theta|M)d\theta,$$

where $p(D|\theta, M)$ is the likelihood of data D, given that the model M with parameters θ are "correct", which is the same as stating that this model and parameters determine the behaviour of the cell; $p(\theta|M)$ is the prior probability of θ; and finally, $p(D|M)$ is the probability of the data being generated by model M. This cost function has as an advantage the fact that models are not ranked using a single value for parameters, instead, the cost considers all possibilities of parameters, integrating over the parameter space. Another advantage of this cost function is that, since it is based on likelihoods of the model to reproduce data, overly complex models are automatically penalized.

The method developed on this work for model selection will then be applied in biological models, mainly related to tumor cells, that are relevant for Center of Toxins, Immune-response and Cell Signaling (CeTICS).

1.1 Objectives

In this work we propose to create a software that applies a method, based on the work o Lulu Wu, that is able to select a model to reproduce some biological experiment. This method should be able to consult a vast database of interactions and modify a starting model by removing or adding those interactions in order to construct a model that approximates more closely the biological experiment. The construction of this database and how the method consults it is also part of this work. We also propose to validate the methodology with real cell data. To achieve these goals, we should complete the following tasks:

1. **Define a cost function for models**. We propose to use a Bayesian approach to implement an algorithm that allows us to estimate the value of $p(D|M)$, which should be used as the "score" of the model. To complete this activity we will use as reference the Bayesian inference-based modeling method (BIBm) [Xu+10] and also the software ABC-SysBio [Lie+14].

2. **Build a database of interactions**. This database should include interactions gathered from KEGG and STRING, and reaction rate constants of interactions, which are available on other databases such as SABIO-RK, BioModels and BioNumbers.

3. **Formulate the systematic modifications on the models as a feature selection search space**. Given the initial model, we should be able to identify this model as a node of the search space, and we also should be able perform valid jumps from one node to another, in other words, we should be able to perform valid modifications to the model.

4. **Define search algorithms on the feature selection problem**. Given that we successfully structured the modifications of the model as a Boolean lattice, we should define algorithms to determine how to traverse this space in order to find a model with the least possible cost in a reasonable amount of execution time.

5. **Test feature selection algorithms**. Using artificial and then real data, we should test if the methodology can select a model that is able to reproduce the behaviour measured on the signaling network.

6. **Apply the methodology on a real case**. Finally, with a tested implementation of the methodology, we should help researchers from CeTICS to identify cell signaling networks of cells that are relevant to their research.

1.2 Organization

Chapter 2

Fundamental Concepts

In this section we provide the concepts that are fundamental to understand the biological and computational problems, methodologies and results that we will present in this work. We start this chapter presenting what is a cell signaling pathway and how can one take measures to identify its activity on the cell. Later we present how it is possible to represent chemical iteractions as differential equations, which implies in how can one model a cell signaling pathway in a system of ordinary differential equations. Then, we present more formally the problem we are trying to solve on this project, the identification of cell signaling pathways, as well as the state of the art methods of model (of signaling pathways) ranking. Finally, we present the basics of posterior distribution sampling, which is a useful tool when working with Bayesian approaches, such as the ones used on this project to rank models.

2.1 Cell Signaling Pathways

Cell signaling pathways are part of the complex cell communication system, and it allows the cell to perceive the conditions of the environment in which it is placed and change its behaviour accordingly. Signaling pathways participate in the regulation of many cell functions, including development, division and cell death. Bad functioning signaling pathways can also be related to diseases, as in many cases of cancer.

The signal perceived by a cell can come from cells that are close (including the same cell that produced the signal), as in synapses, or it can travel long distances in the organism, as in hormones. When a signal reaches a cell, it can either penetrate the cell or bind to some specific receptor in the membrane. Once either of those events happen, the signal or the receptor can trigger a sequence of chemical interactions that can include change of conformation of proteins, activation or inactivation of proteins, and change of concentration of chemical species in the cell. Ultimately, this chain of chemical reactions caused by the signal can alter the behaviour of the cell, what is called signal transduction.

Since signaling pathways participate in many of the cell functions, and are also related to diseases, it is important to study those structures in order to get a better understanding of the cell mechanisms and diseases. One approach on the study of the cell signaling pathways is to measure the concentration change of proteins that participate on the pathway of interest.

2.2 Measurements of Proteins in Cell Signaling Pathways

Western blot is a laboratory technique that can indicate the amount of a specific protein that is present in a mixture. This technique show the presence of a protein in a mixture by "blotting" a membrane where the molecules of interest are located. We can superficially summarize the procedure in the following steps: first a mixture containing a sample of cells of interest must be created; second, proteins from the mixture should be fixed on the blotting membrane; third, an antibody should bind to the target protein molecules; and finally, a method for highlighting the bound antibody should be applied. An image of the resulting membrane can then be analyzed with computer programs to quantify the relative concentration (with respect to some other protein, usually a control protein that has fairly the same concentration during the whole experiment) of the protein of interest.

By repeating this procedure in different times it is possible to create time-course observations of proteins throughout the biological experiment. With this tool, a researcher can choose a set of relevant proteins from a signaling network and gain knowledge about the dynamics of such chemical species during the experiment. For instance, in a signaling network experiment in which it is desired to understand how the change of concentrations of a species at the beginning of the pathway changes the concentration of some species at the end of the cascade, then measurements of both are relevant to understand the biological experiment. Figure 2.1 presents an example of time-course Western blot for an experiment where it is desirable to understand how extracellular signal-regulated kinase (ERK) is activated (phosphorylated) as a function of levels of Rat sarcoma bound to guanosine triphosphate (Ras-GTP).

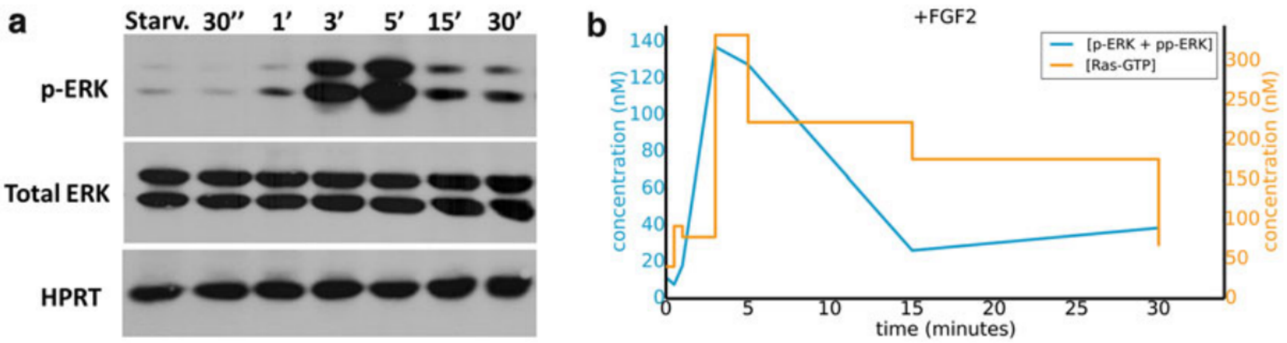


Figure 2.1: Figure **a** *shows time-course measurements of ERK, phosphorylated ERK and hypoxanthine-guanine phosphoribosyltransferase (HPRT). HPRT is a "loading" protein, that means that its concentration is fairly the same through the experiment, and therefore it is used as a normalizing factor to total ERK concentration. Figure* **b** *shows values of phosphorylated ERK that are obtained after processing figure* **a**. *Original image of Marcelo S. Reis et al. (2017) [Rei+17a].*

These measurements alone do not always provide means for researchers to understand a cell signaling pathway experiment. However, if we create a computational models for this signaling networks that is able to reproduce experimental data, then it is possible to use this model as a summary of the signaling network, which can provide to researchers evidences of the biological phenomena.

2.3 Dynamic Modeling of Cell Signaling Pathways

One approach onto modeling cell signaling pathways is to model the dynamics of the concentrations of chemical species involved. This can be accomplished when using the law of mass action. This law states that, in an elementary reaction, the speed (or rate) of a chemical reaction is proportional to the product of the concentration of all reactants. An elementary reaction is a reaction in which there is no participation or need of an intermediate reaction to describe the first in a molecular level. In practice, it is more common to see two types of elementary reactions, they are first or second order reactions.

2.3.1 Modeling Elementary Reaction Rates

A first order reaction is composed of one reactant only. Suppose A is the only reactant and B is the only product of a reaction, then we can write this reaction as:

$$\mathrm{A} \longrightarrow \mathrm{B}.$$

The reaction rate of this reaction, according to the law of mass action, is

$$k_1[\mathrm{A}],$$

where k_1 is some constant and [A] is the concentration of A. It is import to note that the constant k_1 is a rate coefficient of the reaction and, therefore, it can only assume positive values.

A second order reaction is composed of two reactants. Suppose C and D are both and the only reactants and F is the product of a reaction, then we can write this reaction as:

$$\mathrm{C} + \mathrm{D} \longrightarrow \mathrm{F}.$$

The reaction rate of this reaction is:

$$k_2[\mathrm{C}][\mathrm{D}],$$

where k_2 is a (positive) constant and [C] and [D] are the concentrations of C and D, respectively.

Using these two laws to calculate the speed of reactions, we are able to describe how the concentration of chemical species in a system change though time using differential equations. To illustrate this and future concepts of this section, we are going to consider a minimal system composed of a simple enzymatic reaction:

$$\mathrm{E} + \mathrm{S} \underset{k_r}{\overset{k_f}{\rightleftharpoons}} \mathrm{ES} \xrightarrow{k_{cat}} \mathrm{E} + \mathrm{P}, \tag{2.1}$$

where E is an enzyme, S is a substrate, ES is the enzyme-substrate complex, and P is the product.

Each arrow in equation 2.1 represents one elementary reaction, and the names over or under arrows represent reaction rate constants. All three reactions can be represented by the equations:

$$\mathrm{E} + \mathrm{S} \xrightarrow{k_f} \mathrm{ES} \tag{2.2a}$$

$$\mathrm{ES} \xrightarrow{k_r} \mathrm{E} + \mathrm{S} \tag{2.2b}$$

$$\mathrm{ES} \xrightarrow{k_{cat}} \mathrm{E} + \mathrm{P} \tag{2.2c}$$

and they have, respectively, reaction rates of:

$$k_f[\mathrm{E}][\mathrm{S}]$$
$$k_r[\mathrm{ES}]$$
$$k_{cat}[\mathrm{ES}].$$

Now, to determine a model of the concentration dynamics of every chemical species involved in reaction 2.1, we will write a system of ordinary differential equations. To do so, for every species we should calculate its concentration change rate based on the rate of each reaction that it participates. For instance, the enzyme E is a reactant on reaction 2.2a and is also a product on reactions 2.2b and 2.2c, then we consider that E changes its concentration over time (t) according to the differential equation:

$$\frac{d[\mathrm{E}]}{dt} = -k_f[\mathrm{E}][\mathrm{S}] + (k_r + k_{cat})[\mathrm{ES}] \tag{2.3}$$

Repeating this procedure for every other species of the enzymatic reaction induces the desired system of ordinary differential equations:

$$\frac{d[\mathrm{E}]}{dt} = -k_f[\mathrm{E}][\mathrm{S}] + (k_r + k_{cat})[\mathrm{ES}] \tag{2.4a}$$
$$\frac{d[\mathrm{S}]}{dt} = -k_f[\mathrm{E}][\mathrm{S}] + k_r[\mathrm{ES}] \tag{2.4b}$$
$$\frac{d[\mathrm{ES}]}{dt} = k_f[\mathrm{E}][\mathrm{S}] - (k_r + k_{cat})[\mathrm{ES}] \tag{2.4c}$$
$$\frac{d[\mathrm{P}]}{dt} = k_{cat}[\mathrm{ES}]. \tag{2.4d}$$

2.3.2 Simplification of Dynamic Models

The system 2.4 can be simplified if we apply properties of enzymatic reactions together with algebraic simplifications. We will show then how to derive the quasi-steady-state Michaelis-Menten model for enzymatic reactions. With the correct assumptions, this model is able to reproduce the behaviour of an enzymatic reaction without considering the intermediate enzyme-substrate complex.

A basic the principle we need to apply to our system in order to derive the Michaelis-Menten model is the principle of mass conservation. This principle is valid if we assume that the reactions 2.1 are isolated, meaning that the chemicals on these reactions are not involved in other reactions at the same time. Applying this principle to the enzyme chemical, produces the following equation:

$$[\mathrm{E_0}] = [\mathrm{E}] + [\mathrm{ES}].$$

If we apply this equation to the derivative of the concentration of ES, we will get the following equation:

$$\frac{d[\mathrm{ES}]}{dt} = k_f([\mathrm{E_0}] - [\mathrm{ES}])[\mathrm{S}] - (k_r + k_{cat})[\mathrm{ES}]. \tag{2.5}$$

One more assumption is necessary to derive the simplification. This assumption states that the concentration of substrate-enzyme complex does not change over time, i.e. $\frac{d[\mathrm{ES}]}{dt} = 0$, and it was first proposed in 1925 by Briggs and Haldane [BH25]. Generally, this assumption is applicable whenever [S] $\gg$ [E]. Applying this assumption together with the mass conservation assumption on the equation 2.5, we get:

$$[\mathrm{ES}](k_r + k_{cat}) = k_f([\mathrm{E}_0] - [\mathrm{ES}])[\mathrm{S}],$$

$$[\mathrm{ES}] = \frac{[\mathrm{E}]_0[\mathrm{S}]}{K_m + [\mathrm{S}]},$$

in which $K_m = \frac{k_{cat}+k_r}{k_f}$ is known as Michaelis constant. Considering this, we can rewrite the rate of [P] as:

$$\frac{d[\mathrm{P}]}{dt} = k_{cat}\frac{[\mathrm{E}]_0[\mathrm{S}]}{K_m + [\mathrm{S}]}. \tag{2.6}$$

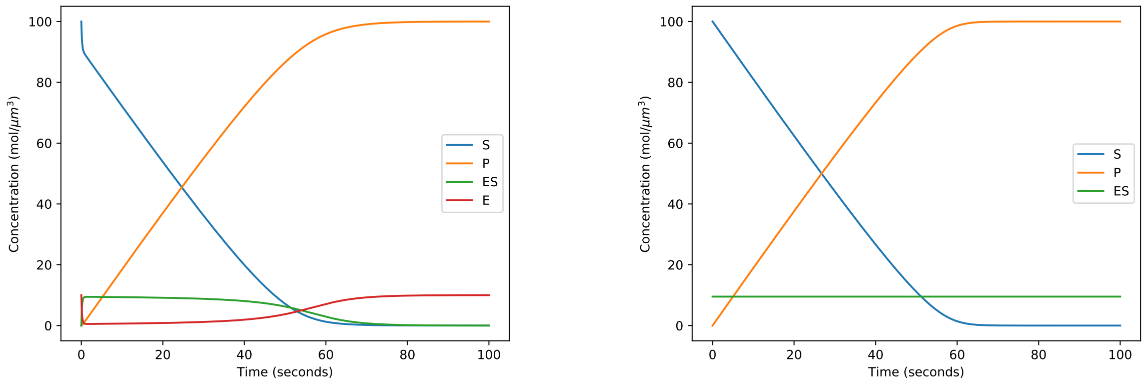
And finally, if we apply mass conservation to the substrate, we will get the following equation:

$$[\mathrm{S}_0] = [\mathrm{S}] + [\mathrm{ES}] + [\mathrm{P}],$$

then, we can differentiate this equation on t and use the quasi-steady-state assumption ($\frac{d[\mathrm{ES}]}{dt} = 0$) to obtain:

$$\frac{d[\mathrm{S}]}{dt} = -\frac{d[\mathrm{P}]}{dt}. \tag{2.7}$$

Now, with equations 2.6 and 2.7 we are able to reproduce the dynamics of the substrate and product of the enzymatic reaction. Therefore, using the Michaelis-Menten model, we could simplify the system 2.4 that had four equations and three parameters to a new model that has only two equation and two parameters (k_{cat} and K_m). Figure 2.2 shows a comparison between the complete and Michaelis-Menten models of enzymatic reactions.



(a) (b)

Figure 2.2: An example of the dynamics produced by two models of enzymatic reactions. The figure 2.2(a) presents the dynamics of the model 2.4 and figure 2.2(b) presents the dynamics of the Michaelis-Menten simplification to the same model. For this simulations, it is necessary to define initial concentrations of the chemical species involved, and it is used: 10 molecules/$(\mu m)^3$ for the enzyme (E); 100 molecules/$(\mu m)^3$ is used for the substrate (S); and 0 molecules/$(\mu m)^3$ is used for the other species. In addition to this, it is also necessary to define model parameter values, and it is used: 0.06 $(\mu m)^3(molecules{*}s)^{-1}$ for k_f; 0.1 (s^{-1}) for k_r; 0.2 (s^{-1}) for k_{cat}; and, following the Michalis-Menten model, 5 molecules/$(\mu m)^3$ is used for K_m.

2.4 Identification of Cell Signaling Pathways

Identification of signaling pathways is the problem of finding the components of a signaling pathway and how they interact in order to reproduce a behaviour of the cell that has been previously measured experimentally. The input of this problem is usually a description of the biological experiment, containing previous information about the signaling pathway, and a set of measurements, commonly Western blot data. The output to this problem is then composed of a set of interactions that are actively controlling the behaviour of interest of the cell, and also the set of parameter values that should be used on these interactions to create a model that approximates the experimental observations; it is possible to output a single value for each parameter, as it was presented in [Wu15], or output information about these values, using a posterior (to the experimental data) distribution, as it was presented in [Lie+14] and [Xu+10].

Two main tasks must be completed to produce this output. The first task is to find candidate topologies for the pathway model, i.e. different set of interactions that are relevant to the pathway of interest. The second task is to rank those models according to their ability to simulate the pathway and approximate the experimental measurements.

The second task is also known as the model selection problem and even though it is a broad area, there are works on the literature that treats specifically the problem in a biochemical context. Solutions to this problem should be able to choose model candidates according to their ability to reproduce observed data and penalize overly complex models to avoid overfitting.

One approach on setting the score of a model is to search for the set of parameter values that makes the model measurements the closest to the experimental data and then define a distance between these two measurements; then it is possible to use this distance plus a penalty for complexity to create a ranking of the models. We can write this scoring function as:

$$score_1(M) = -min_{\{\theta \in \Theta' \subset \Theta\}} dist(\phi(M, \theta), D) + R(M),$$

where M is the model, Θ is the parameter space, Θ' is the subset of the parameter space where the search for the best parameter values was conduced, D is the experimental measurement, ϕ is a function that determines the simulated measurement on the model, and R is a regularization function that penalizes model complexity. This approach was implemented on the work of Lulu Wu [Wu15], using a Simulated Annealing algorithm to search for parameter values that minimizes the distance between simulation and experiment; however, the methodology showed limitations when testing the ability to reconstruct models from experiments, and this could be related to the used penalization term. In fact, choosing a good regularization function is crucial to the performance of this methodology.

Another approach is to consider the model parameters as random variables and then marginalize the probability of the model and parameters to reproduce the observed data, i.e. estimate (because calculating is usually hard) the marginal probability:

$$p(D|M) = \int_{\Theta} p(D|M, \theta) p(\theta|M) d\theta, \tag{2.8}$$

where D is the observed data, M is the model and Θ is the parameter space. The function $p(\theta|M)$ is the prior probability function of the parameter θ on model M and the function $p(D|M, \theta)$ is the likelihood of observing the data D when simulating the model M using parameters θ. Sometimes, however, the likelihood function is unknown or computationally intractable; for these cases, it is possible to use an alternative Bayesian approach, called Aproximate Bayesian Computation (ABC), to estimate the probability $p(M|D)$ and use it as a ranking score [Ton+09].

For the first task of identification of signaling pathways we described, the creation of model candidates, there are not as many works on the literature as there are for the second task. Commonly, researchers must resort on their own knowledge on the biological experiment and consult interactome maps available on repositories such as KEGG and BioModels to construct manually the hypothesis of models for a signaling pathway. That enlightens the importance to create a methodology that systematically creates candidate models of signaling networks, as we propose on this project.

2.5 State of the Art in Selection of Biochemical Models

The state of the art in biochemical model selection is based on Bayesian inference. The Bayesian approaches provide the benefits of ranking models with statistical formalism, and automatically penalizes overly complex models. More than that, through the prior distribution of the parameters, the researcher is able to input prior knowledge about interactions constants; this type of information can facilitate the parameter inference of models since it tends to concentrate the search.

Bayesian approaches consider that model parameters are random variables, instead of fixed unknown constants. We can argue that this modeling is fair to the reality in biochemical processes because interactions constants can vary depending on the cell conditions. Therefore, in a biological experiment in which there might be perturbations to the cell environment, it should be more adequate to rank models integrating the models score over a probability space of parameters instead of fitting the model to data using a single point of the parameter space. We will now present the basic concept of two methods that use this idea for model selection, the Annealing-Melting Integration (AMI) [VG07] and Approximate Bayesian Computation (ABC) [Ton+09].

The Annealing-Melting Integration is a method that estimates the integral 2.8 using concepts of thermodynamics. With thermodynamic integration, it is possible to write the logarithm of this integral as:

$$\ln p(D|M) = \int_0^1 \mathbb{E}_{q_\beta(\theta)}[\ln p(D|M,\theta)]d\beta \tag{2.9}$$

where $p(D|M,\theta)$ is the likelihood function (the probability of observing the data D when M is the correct model, with parameters θ); and $\mathbb{E}_{q_\beta(\theta)}$ is an expectation taken over the probability space of $q_\beta(\theta) \propto p(D|M,\theta)^\beta p(\theta|M)$. The variable β works in the integral as a temperature term, determining the probability functions $q_\beta(\theta)$; note that when $\beta = 0$ then

$$q_0(\theta) = p(\theta|M),$$

the prior distribution of the parameters, and when $\beta = 1$ then

$$q_1(\theta) = \frac{p(D,\theta|M)}{\int_\Theta p(D,\theta|M)} = \frac{p(\theta|D,M)p(D|M)}{p(D|M)} = p(\theta|D,M),$$

the posterior distribution of parameters. Therefore, the integral 3.6 takes the expected value of the likelihood function of D over a sequence of probability distributions that is a “bridge of distributions” connecting the prior and posterior distributions of parameters. Calculating the integral is usually infeasible, hence in practice it is needed to estimate this integral using samples of a finite number of tempered distributions, $q_\beta(\theta)$.

As we mentioned before, the likelihood function $p(D|M,\theta)$ may be very hard to calculate if not impossible. For those cases it is possible to use parameter inference approaches that are likelihood-free, called Approximate Bayesian Computation (ABC). This method has the goal of producing a sample of parameters that brings the model simulations close to the observed data. A generic ABC algorithm starts proposing a candidate parameter θ^* from a proposal distribution; then, a simulation, $\phi(M, \theta^*)$ of the model using the candidate parameters is produced; if, for some distance function d, is true that $d(\phi(M, \theta^*), D) < \epsilon$, then we accept θ^* as part of the sample. If ϵ is sufficiently small, then the produced samples approximates well the posterior distribution. To use ABC methods for model selection it is enough to add a model indicator parameter to the parameter array, then it is possible to extract model distribution of the accepted parameters.

2.6 Metropolis-Hastings to Generate Samples

Both methods for model selection, based on ABC or using thermodynamics integration need to generate samples of probability distributions. Generating samples of distributions is simple for many well known distributions, however, for some other distributions it may not be as simple. Metropolis-Hastings algorithms are capable of generating a sample that has some probability distribution p, which is called the target distribution. In fact, the method can be used even when it is not possible to access directly the target function, because all it is needed to perform the sampling is a function that is proportional to the target.

Being able to generate a sample of a target distribution without accessing the probability function itself is useful for our applications in Bayesian model ranking. Consider that we need to create a sample of the parameters posterior distribution $p(\theta|M, D)$. Calculating this probability function is very hard because

$$p(\theta|M, D) = \frac{p(D|\theta, M)p(\theta|M)}{p(D|M)},$$

and this equation has the term $p(D|M)$, which is only known (by some estimation) at the end of the model ranking; however, if we can access the likelihood function $p(D|\theta, M)$ and the prior $p(\theta|M)$ than the product of these two is proportional to the posterior distribution (since $p(D|M)$ is only a constant because it does not depend on θ), and therefore they are enough to generate a sample of the posterior.

A generic Metropolis-Hastings algorithm that creates a sample of a target distribution $p(\lambda)$ proceeds as follows:

1. Choose some starting point λ_0 for which $p(\lambda_0)$ is not zero. Also set $t = 1$.

2. Sample a candidate point λ^* from a proposal (or jumping) distribution with probability $J_t(\lambda^*|\lambda^{t-1})$.

3. Calculate the ratio:
$$r = \frac{p(\lambda^*)J_t(\lambda^{t-1}|\lambda^*)}{p(\lambda^{t-1})J_t(\lambda^*|\lambda^{t-1})} \tag{2.10}$$

4. With probability $min(1, r)$ set $\theta^t = \theta^*$ and set $\theta^t = \theta^{t-1}$ otherwise.

5. Increase t by one and, if not reached limit number of iterations, go back to step 2.

Note that if the target $p(\lambda)$ is not available, and rather another function $q(\lambda) = \frac{1}{c}p(\lambda)$ is available, then the ratio 2.10 can be calculated as:

$$r = \frac{p(\lambda^*)J_t(\lambda^{t-1}|\lambda^*)}{p(\lambda^{t-1})J_t(\lambda^*|\lambda^{t-1})} = \frac{(q(\lambda^*)c)J_t(\lambda^{t-1}|\lambda^*)}{(q(\lambda^{t-1})c)J_t(\lambda^*|\lambda^{t-1})} = \frac{q(\lambda^*)J_t(\lambda^{t-1}|\lambda^*)}{q(\lambda^{t-1})J_t(\lambda^*|\lambda^{t-1})}$$

More than that, if the proposal distribution is symmetric, the produced algorithm is called Metropolis algorithm and has the ratio $r = p(\lambda^*)/p(\lambda^{t-1})$.

Different implementations of the Metropolis-Hastings algorithm are possible. The possible changes include the choice of starting point, the choice of proposal distributions and number of iterations. As an example, some algorithms are adaptive in the sense that they can change the proposal distribution according to the acceptance rate of proposed points [Gel+13].

Chapter 3

Model Selection

In this chapter we will present two state of the art methodologies that can be used to rank models, both of them are Bayesian approaches. The first approach is to estimate the marginal likelihood of the data D being reproduced by a model M, $p(D|M)$. The estimation of this probability is done by taking samples of tempered posterior distributions of model parameters; these tempered posterior distributions bridge the prior and posterior parameter distributions and allow a better estimation of the marginal likelihood. Another method, likelihood-free, is to use Approximate Bayesian Computation; this method allows us to estimate the probability $p(M|D)$ in a simpler algorithm.

3.1 Model ranking using Marginal Likelihood

The marginal likelihood of an experiment measurement D being reproduced by a model M, $p(D|M)$, can be used as a model ranking metric as it determines which model makes the experimental observations more likely to happen. Before defining how to calculate the marginal likelihood, we must define what the likelihood function is. To calculate the likelihood $p(D|M,\theta)$, we must understand that conditioning the observation to the model and parameters means that in the probability space from which D is taken, the model M is the "real" model and it has the parameter values of θ; i.e. the model M with parameters θ controls the behaviour of the system from which D was observed. Then, assuming that the observations have a Gaussian error, and that they are taken in a time series of m time steps, we can define the likelihood as:

$$p(D|M,\theta) = p_{\mathcal{N}_{(\vec{0},\Sigma)}}(\phi(M,\theta) - D), \tag{3.1}$$

where $\phi(M,\theta) \in \mathbb{R}^m$ is the experimental measurement on the simulation generated by the model M with parameters θ, and $p_{\mathcal{N}_{(\vec{0},\Sigma)}}(\cdot)$ is the probability density function of a Multivariate Normal variable with mean $\vec{0}$ and covariance matrix Σ. As a matter of fact, as it is done in the work of Xu et al. (2010), we can consider that the observation error is independent for each time step [Xu+10], therefore we can simplify 3.1 to:

$$p(D|M,\theta) = \prod_{i=1}^{m} p_{\mathcal{N}_{(0,\sigma^2)}}(\phi_i(M,\theta) - D_i). \tag{3.2}$$

The σ^2 used in equation 3.2 is also a parameter of the model, which means that, for some k, $\theta_k = \sigma^2$.

Now that we defined the likelihood function, we can write the marginal likelihood as:

$$p(D|M) = \int_{\Theta} p(D|M,\theta)p(\theta|M)d\theta. \tag{3.3}$$

However, calculating this integral analytically is only possible in very special cases and, usually, it would depend on knowing models for the distributions associated to these probability functions, which is generally not possible in our case.

Even though this integral is very hard to be calculated, there are methods that allow us to estimate its value. A straight forward method to estimate this integral value is using Importance Sampling Estimators [NR93]. This method uses the Monte Carlo integral estimation method that can estimate integrals of the form $\int g(\lambda)p(\lambda)d\lambda$ using the estimator:

$$\hat{I} = \sum_{i=1}^{m} w_i g(\lambda_i) / \sum_{i=1}^{m} w_i,$$

where $w_i = p(\lambda)/p^*(\lambda)$, and $p^*(\cdot)$ is known as the importance sampling function. If we set $\lambda = \theta|M$ and use the prior $(p(\theta|M))$ or the posterior $(p(\theta|M, D))$ as importance sampling functions, then we would get respectively the estimators:

$$\frac{1}{m}\sum_{i=1}^{m} p(D|M, \theta^{(i)}) \quad (\text{with } \theta^{(i)} \sim p(\theta|M)),$$

$$\left(\frac{1}{m}\sum_{i=1}^{m} p(D|M, \theta^{(i)})^{-1}\right)^{-1} \quad (\text{with } \theta^{(i)} \sim p(\theta|M, D)).$$

However, as showed by Vyshemirsky et al. (2007), these estimators might produce very large variances and may not perform well for biochemical model selection applications. Hence, new methods with ideas of thermodynamics were proposed. These methods are based on rewriting the marginal likelihood equation using intermediate distributions of parameters between the prior and posterior distributions [FP08].

3.1.1 Thermodynamic Integration for Marginal Likelihood

The Thermodynamic Integration is a method that proposes to rewrite the integral 3.3 using ideas of thermodynamics, providing new estimators for the marginal likelihood. This method is able to rewrite the marginal likelihood integral in a way that it marginalizes the likelihood through many intermediate probability spaces of parameters, bridging the prior and posterior distributions of parameters. These distributions are also called tempered distributions or power posteriors [FP08].

Given a parameter prior distribution $p(\theta|M)$ and a posterior distribution $p(\theta|D, M)$, then we define a power posterior distribution with temperature β as:

$$p_\beta(\theta) = \frac{p(D|\theta, M)^\beta p(\theta|M)}{z(\beta)},$$

where

$$z(\beta) = \int_\Theta p(D|\theta, M)^\beta p(\theta|M) d\theta.$$

Note that when $\beta = 0$, then $p_{\beta=0} = p(\theta|M)$, the prior distribution of the parameters; also, when $\beta = 1$, then

$$p_{\beta=1}(\theta) = \frac{p(D|\theta, M)p(\theta|M)}{z(\beta)} = \frac{p(D, \theta|M)}{\int_\Theta p(D, \theta|M)d\theta} = \frac{p(\theta|D, M)p(D|M)}{p(D|M)} = p(\theta|D, M),$$

the posterior distributions of the parameters.

Now, let we consider the derivative of $\ln z(\beta)$.

$$\begin{aligned}\frac{d}{d\beta}\ln z(\beta) &= \frac{1}{z(\beta)}\frac{d}{d\beta}z(\beta)\\ &= \frac{1}{z(\beta)}\frac{d}{d\beta}\int_\Theta p(D|\theta,M)^\beta p(\theta|M)d\theta\end{aligned}$$

(using that $\frac{d}{dx}c^x = c^x \ln c$)

$$\begin{aligned} &= \frac{1}{z(\beta)}\int_\Theta p(D|\theta,M)^\beta p(\theta|M)\ln p(D|\theta,M)d\theta\\ &= \int_\Theta \frac{p(D|\theta,M)^\beta p(\theta|M)}{z(\beta)}\ln p(D|\theta,M)d\theta\\ &= \int_\Theta p_\beta(\theta)\ln p(D|\theta,M)d\theta\\ &= \mathbb{E}_{p_\beta(\theta)}[\ln p(D|\theta,M)]. \qquad (3.4)\end{aligned}$$

And it is not hard to see that:

$$\begin{aligned} z(0) &= \int_\Theta p(\theta|M)d\theta = 1\\ z(1) &= \int_\Theta p(D,\theta|M)d\theta = p(D|M)\end{aligned} \qquad (3.5)$$

Using equations 3.5 and equality 3.4 we can write that:

$$\begin{aligned}\int_0^1 \mathbb{E}_{p_\beta(\theta)}[\ln p(D|\theta,M)]d\beta &= \int_0^1 \frac{d}{d\beta}\ln z(\beta)d\beta\\ &= \Big[\ln z(\beta)\Big]\Big|_0^1\\ &= \ln p(D|M). \qquad (3.6)\end{aligned}$$

Then we have written an expression for the logarithm of the marginal likelihood. This expression is still hard to be calculated analytically, however from this equation we will be able to find estimators for the logarithm of the marginal likelihood, and consequently for the likelihood. To calculate this estimators, we will need to generate samples for a series of power posteriors of parameters, and this will be explained in the next section.

3.1.2 Estimation of the Marginal Likelihood

There are multiple approaches on estimating the integral 3.6, and for all of them, it is necessary to find samples of $p_{\beta_t}(\theta|M,D)$ for a sequence of values of β_t that vary from zero to one [Xu+10; VG07; FP08]. The differences on these methods are mainly on the choice of the sequence $\{\beta_1,\beta_2,\ldots,\beta_T\}$, on the Metropolis-Hastings (MH) algorithms used, and finally on the estimator.

We are now going to show a possible methodology to estimate parameter values. First, we assume that there is a chosen sequence of $\{\beta_1,\beta_2,\ldots,\beta_T\}$ to explain how to generate samples of power posterior distributions with these temperatures. Then, we explain how to choose these values and present two possible estimators for $p(D|M)$.

Power posteriors sampling

Given a sequence of temperatures, the method used to sample from the power posteriors has three different steps, all of them using Metropolis-Hastings algorithms, similarly to what is done by Xu et al. (2010). For each of the T temperatures, the first two phases are run separately, generating T chains that are samples of the power posteriors. Then, on the third phase, each chain continues to grow, but not independently because two random chains will have their last accepted point swapped, what causes the chains to be mixed; this approach on mixing these chains is called Populational Monte Carlo Markov Chain (Populational MCMC) [FP08].

The first step of sampling is run independently for each temperature. The Metropolis-Hastings performed on this step is started taking a sample from the parameter priors. The proposal distribution has to generate positive numbers and it should have mean approximately equal to the last point sampled; more than that, we decide to use a diagonal covariance matrix, i.e. parameters are proposed independently. This covariance matrix is updated after every predefined number of iterations with the goal of adapting the proposal distribution to the parameter space. This update is performed as proposed in Gelman et al. (2013):

- if the acceptance rate of parameter points in the last iterations is greater than 0.44, then increase the variance of the jump;
- if the acceptance rate is lower than 0.23, then decrease the variance of the jump.

Given that we are taking a sample from a power posterior with temperature β_t, the target function is $p_{\beta_t}(\theta)$. Hence, if the current point is $\theta^{(t,i)}$, the probability of accepting a proposed parameter $\theta^* \sim J_{(t,i)}(\theta^*|\theta^{(t,i)})$, is $\min(1, r)$ with

$$r = \frac{p_{\beta_t}(\theta^*)}{p_{\beta_t}(\theta^{(t,i)})} \frac{J_{(t,i)}(\theta^{(t,i)}|\theta^*)}{J_{(t,i)}(\theta^*|\theta^{(t,i)})},$$

with $J_{(t,i)}$ being the proposal jump distribution on iteration i on chain of temperature β_t. Because sampling from $p_t(\theta|M, D)$ directly is not possible, and since we defined in the beginning of the chapter that $p_{\beta_t}(\theta) \propto p(D|M,\theta)^{\beta_t} p(\theta|M)$, we can rewrite this equation as:

$$r = \frac{p(D|M,\theta^*)^{\beta_t}}{p(D|M,\theta^{(t,i)})^{\beta_t}} \frac{p(\theta^*|M)}{p(\theta^{(t,i)}|M)} \frac{J_{(t,i)}(\theta^{(t,i)}|\theta^*)}{J_{(t,i)}(\theta^*|\theta^{(t,i)})}. \tag{3.7}$$

The second iteration also samples each temperature independently, and differently from the first step, the parameters are not necessarily sampled independently. In fact, a portion of the last sampled parameters on step one is used to create a covariance matrix, and this matrix is then used as the covariance matrix of the first proposal distribution of the second step of sampling. After the first iteration, for every sampled parameter on the second step, the covariance matrix is updated to be the sample covariance of the selected points of step one plus the current points accepted on step two. The probability of accepting a proposed parameter on this step is the same as it was on the first step, as described by equation 3.7.

At the end of the second step, we will have T chains of sampled parameters, containing the selected parameters of the first and second steps. For each of these chains there is a covariance matrix that was being used as the covariance of the proposal distribution of the respective temperature. On step three, the covariance matrix is fixed and a Populational MCMC algorithm is performed. This algorithm continues the sampling for each temperature, now with a fixed covariance on the proposal distribution, and also mixes the samples of different temperatures in every iteration. This mix is achieved by swapping the last sampled elements of two random power posterior chains; the first of the two chains chosen to swap, of temperature β_i, is chosen uniformly, while the second chain, of temperature β_j, is chosen following a Discrete Laplacian distribution given β_i, with probability function $p(j|i) \propto e^{|i-j|/2}$. This algorithm was proposed by Friel et al. (2008) and can be summarized in the following steps:

1. For each temperature β_t, $t \in \{1, \dots, T\}$, update the t-th chain using MCMC with a proposal distribution that has the same covariance matrix used on the last iteration of temperature β_t of the second step.

2. Choose uniformly i from $\{1, \dots, T\}$. Then, choose j with probability density function $p(j|i) \propto e^{|i-j|/2}$. Finally, swap the last sampled points of the chains of temperature β_i and β_j, which are $\theta^{(i,k)}$ and $\theta^{(j,k)}$, with probability $\min\{1, r\}$, where:

$$r = \frac{p_{\beta_i}\left(\theta^{(j,k)}\right)}{p_{\beta_i}\left(\theta^{(i,k)}\right)} \frac{p_{\beta_j}\left(\theta^{(i,k)}\right)}{p_{\beta_j}\left(\theta^{(j,k)}\right)} \frac{p(j|i)}{p(i|j)}$$

which can be simplified to:

$$r = \left[\frac{p\left(D|\theta^{(j,k)}\right)}{p\left(D|\theta^{(i,k)}\right)}\right]^{\beta_i} \left[\frac{p\left(D|\theta^{(i,k)}\right)}{p\left(D|\theta^{(j,k)}\right)}\right]^{\beta_j} \frac{p(j|i)}{p(i|j)}$$

After finishing the third step of sampling, the samples obtained on the first two phases are discarded and only the samples from the last phase are used to estimate the marginal likelihood.

Estimators of the Marginal Likelihood

The sampling procedure explained on the section 3.1.2 produces samples of the power posteriors $p_{\beta_1}(\theta), \dots, p_{\beta_T}(\theta)$, and these samples can be used to estimate logarithm of the marginal likelihood:

$$\ln p(D|M) = \int_0^1 \mathbb{E}_{p_\beta(\theta)}[\ln p(D|\theta, M)]d\beta. \tag{3.8}$$

This can be achieved using both numerical integration or creating an unbiased estimator. The choice of either approach of estimation will imply in a method for choosing the sequence of temperatures $\beta_1, \dots, \beta_T$.

The method proposed by Friel et al. (2008), uses a numerical integration method to estimate the integral 3.8. Given that T temperatures, $0 = \beta_1 < \beta_2 < \dots < \beta_T = 1$, were selected and samples of its respective power posteriors were generated, then using trapezoidal rule over the temperature allows us to estimate the integral as:

$$\log p(D|M) \approx \sum_{t=0}^{T-1} (\beta_{t+1} - \beta_t) \frac{\mathbb{E}_{p_{\beta_{t+1}}(\theta)}[\log p(D|M, \theta)] + \mathbb{E}_{p_{\beta_t}(\theta)}[\log p(D|M, \theta)]}{2}$$

and if the sample of power posterior of temperature β_t has M_t parameter points, then we can rewrite this equation as:

$$\log p(D|M) \approx \sum_{t=0}^{T-1} (\beta_{t+1} - \beta_t) \frac{\frac{1}{M_{t+1}} \sum_{i=1}^{M_{t+1}} \log p(D|M, \theta^{(t+1,i)}) + \frac{1}{M_t} \sum_{i=1}^{M_t} \log p(D|M, \theta^{(t,i)})}{2} \tag{3.9}$$

According to the work of Friel et al. (2008), a good temperature schedule for $\beta_1, \dots, \beta_T$ that can be used in this approach is:

$$\beta_t = \left(\frac{t-1}{T-1}\right)^c,$$

with $t \in 1, \dots, T$; with better results achieved when T is between 20 and 100 and c is between 3 and 5.

Another method, proposed by Xu et al. (2010), considers that the temperature β can be treated as a random variable, and therefore we can rewrite integral 3.8 as:

$$\mathbb{E}_{p(\beta)}\left[\frac{\mathbb{E}_{p_\beta(\theta)}[\ln p(D|\theta, M)]}{p(\beta)}\right] \tag{3.10}$$

The author uses this ideas to derive the following estimator. First, the interval $[0, 1]$ is discretized into $S - 1$ disjoint intervals $\Delta\beta_i = [t_{i+1}, t_i]$ such that $\sum_{i=1}^{S-1}(t_{i+1} - t_i) = 1$. Then, for each interval $\Delta\beta_i$, T_i temperatures are taken randomly from the uniform distribution on the interval $[t_{i+1}, t_i]$. Finally, the estimator of the logarithm of the marginal likelihood is given by:

$$\sum_{s=1}^{S-1} \frac{|\Delta\beta_k|}{T_k} \sum_{i=1}^{M_k} \log p(D|M, \theta^{(\beta_{k,i})}) \tag{3.11}$$

where $\beta_{k,i}$ is the i-th sampled temperature from the interval $\Delta\beta_k$; $\theta^{(\beta_{k,i})}$ is a parameter sampled from the power posterior $p_{\beta_{k,i}}(\theta)$; and $|\Delta\beta_k| = t_{k+1} - t_k$. However, Xu et al. do not provide information about the discretization method of the interval $[0, 1]$.

3.2 Approximate Bayesian Computation

Approximate Bayesian Computation (ABC) is an approach that allows generating samples from the posterior distribution without accessing the likelihood function. Adding a model indicator parameter to parameters being sampled also allows us to create an approximate sample of the posterior $p(\theta, M|D)$, and from this sample it is possible to estimate $p(M|D)$, which can be used as a model ranking metric. The main idea of ABC methods is to generate a sample from posterior by generating parameter points that, when plugged into a model, generates simulations that dist from the experimental measurements at most by some small ϵ.

A generic ABC method that generates a sample of the posterior $p(\theta, m|D)$ is composed of the following steps:

1. Sample a parameter candidate (θ^*, M^*) from some proposal distribution.

2. Simulate the model M^* with parameter values θ^*, generating simulated measurements $\phi(M^*, \theta^*) = D*$.

3. Calculate, for some distance function d, the value of $d(D^*, D)$. If $d(D^*, D) < \epsilon$ for some previously specified ϵ, then add (θ^*, M^*) to the sample.

4. Repeat until some iteration limit.

The simplest ABC algorithm is the ABC Rejection, and it goes very similarly to the generic algorithm we just presented, with the specification that on step 1 the proposal distribution is the prior distribution. The output of this algorithm is a sample of the distribution

$p(\theta, M|d(\phi(M,\theta), D) \leq \epsilon)$; when $\epsilon \to \infty$ this is then a sample of the prior distribution, and as $\epsilon \to 0$ then this sample tends to be a sample of the posterior distribution [Pri+99]. This algorithm, however, does not perform well when the posterior distribution is very different from the prior distribution. For that reason, new ABC methods using Monte Carlo Markov Chains were created [Mar+03]. The ABC MCMC method proposed by Marjoram et al. (2003) produces a Markov Chain whose stationary distribution is $p(\theta, M|d(\phi(M,\theta), D) \leq \epsilon)$. Nonetheless, this algorithm might still suffer from correlation in samples or even get stuck in regions of local peaks of probability. For that reason, the ABC sequantial Monte Carlo (ABC SMC)method was proposed [Ton+09].

The ABC SMC method creates a sequence of samples with the goal of getting closer to a posterior sample in each step. Let we simplify the notation by saying that $d(\phi(M,\theta), D)$ is just $\rho_{M,\theta}$. From a predefined sequence $\epsilon_1, \ldots, \epsilon_T$ the algorithm generates a sequence of samples that represents the distributions $p(\theta, M|\rho_{M,\theta} \leq \epsilon_1), p(\theta, M|\rho_{M,\theta} \leq \epsilon_2), \ldots, p(\theta, M|\rho_{M,\theta} \leq \epsilon_T)$. At the first generation, a sample of $p(\theta, M|\rho_{M,\theta} \leq \epsilon_1)$ is created by proposing points from the prior distribution. Then, for next generations the candidates to the sample are proposed based on the points of the last generation and their weight, plus some noise determined by a perturbation Kernel; the weight of a point (θ^*, M^*) on generation t is a estimative of $p(\theta = \theta^*, M = M^*|\rho_{M,\theta} \leq \epsilon_t)$. The pseudo-code 1 presents the ABC SMC algorithm.

```
ABC SMC ($\mathcal{M}, D$)
Define the sequence $\epsilon_1, \ldots, \epsilon_T$.
Define $N$, the sample size for each generation.
Sample $\{(\theta^{(1,1)}, M^{(1,1)}), (\theta^{(1,2)}, M^{(1,2)}), \ldots, (\theta^{(1,N)}, M^{(1,N)})\}$ from $p(\theta, M|D)$.
Set $w^{(1,i)} = 1, \forall i \in 1, \ldots, N$.
for $t \in \{1, \ldots, T\}$ do
    $i \leftarrow 1$
    while $i \leq N$ do
        Sample $M^*$ from $p(M|D)$, the model prior.
        Sample $(\theta^{(t-1,k)}, M^*)$ from the last generation with weight $w^{(t-1,k)}$.
        Create $(\theta^*, M^*)$ by perturbing $\theta^{(t-1,k)}$; $\theta^* \propto K^t(\theta|\theta^{(t-1,k)})$.
        if $p(\theta^*|M^*) = 0$ then
            Continue to next iteration.
        end if
        $D* \leftarrow \phi(M^*, \theta^*)$
        if $d(D^*, D) \leq \epsilon$ then
            $i \leftarrow i + 1$
            $(\theta^{(t,i)}, M^{(t,i)}) \leftarrow (\theta^*, M^*)$
        end if
    end while
    Calculate the weights of the population: $w^{(t,i)} = \frac{p(\theta^{(t,i)}|M^{(t,i)})}{\sum_{j=1}^{N} w^{(t-1,j)} p_{K^t}(\theta^{(t-1,j)}, \theta^{(t,i)})}$
end for
return
```

Algorithm 1: Pseudo-code of ABC SMC.

The ABC SMC algorithm is implemented in Python language in a software called ABC-SysBio [Lie+14].

Chapter 4

Experiments on Model Selection

4.1 Software for Ranking of Models of Signaling Networks

In this section we describe the software we used to perform the ranking of signaling network models. The first software, SigNetMS, is an implementation of the model ranking method presented on section 3.1, which estimates the marginal likelihood of the data being reproduced by a model. The second software, ABC-SysBio implements the method based on the approximate Bayesian Computation, presented on section 3.2.

4.1.1 SigNetMS

To perform the model ranking using an estimate of the marginal likelihood, we created the software SigNetMS, which is an acronym of **Sig**naling **Net**work **M**odel **S**election. This software was implemented in Python and it is a free software, under the *GNU General Public License*, and available on GitHub. The SigNetMS software can read and parse files containing models of signaling network represented in the Systems Biology Markup Language (SBML) format and then construct the respective system of differential equations according to the chemical species and interactions defined on the file. The experiments observations and prior distributions of parameters are defined by the user with Extensible Markup Language (XML) files. Four other parameters are necessary to run SigNetMS, all of them related to the sampling of the power posteriors.

The method we used to sample each of the power posterior distributions is the one we presented on section 3.1.2. The four parameters related to the sampling determine the size of all sampling performed and also the adaptive behaviour of one of the Metropolis-Hastings algorithm used. The first parameter defines the number of iterations of the first sampling algorithm. This algorithm is adaptive because, after a fixed number of iterations, the covariance of the jump distribution is updated according to the acceptance rate of proposed points; this fixed number of iterations is the second sampling parameter. Finally, the third and fourth parameter determine the size of the second and third sampling steps respectively.

To implement the method we still have to define the proposal distributions. Since the proposed model parameters cannot have nonpositive values (because they are reaction rate constants), the proposal distribution should have a probability density function that has value zero for nonpositive numbers. Moreover, its desired, through the sampling steps, to control the mean and variance of the proposal distributions. For all of the three steps, if the current point of the chain is θ, then the jump distribution should have mean close to θ. Also, in the first step, the covariance matrix of the proposal distribution should be proportional to the covariance of the prior distribution of parameters. Then, for the two final steps, the covariance matrix of the jump distribution should be proportional to an estimate $\hat{C}$ of the covariance of the parameters, calculated with the accepted points of the chain.

In our first attempt, we used multivariate lognormal distributions for all the sampling steps. If X is a $MultivariateNormal(\mu, \Sigma)$, then $Y = e^X$ is $MultivariateLognormal(\mu, \Sigma)$. However, we found out that for some combinations of θ and $\hat{C}$, there is no combination of parameters (μ, Σ) such that $Y \propto MultivariateLognormal(\mu, \Sigma)$ with $\mathbb{E}[Y] = \theta$ and $var(Y) = \hat{C}$. Then the solution we proposed is to use a truncated normal distribution for which only positive numbers have positive probability. We can generate a truncated normal random variable, $Y \propto TruncatedNormal(\mu, \Sigma)$ by repeatedly generating a normal random variable, $X \propto Normal(\mu, \Sigma)$ until X is positive. However, this approach has the drawback that $\mathbb{E}[Y]$ is usually greater than μ, implying on a biased run of Metropolis-Hastings.

The SigNetMS program also has an optional argument that allows the user to get a verbose run, showing all proposed parameters for each temperature as well as the accepted parameters used to estimate the logarithm of the marginal likelihood.

4.2 ABC-SysBio

ABC-SysBio is a Python software that implements the Approximate Bayesian Computation Sequential Monte Carlo (ABC SMC) method [Lie+10]. This software, similarly to SigNetMS, also takes as input SBML models as well as prior distribution of parameters and experimental data. As the output, the software returns, for each candidate model, an estimate of the probability of that model reproducing the experimental data. The source code of the method is available on SourceForge.

4.3 Model Selection Experiments

The experiments we performed to test both methods consists in taking four candidate models and ranking them according to experimental data generated by one of them. To generate the simulation data of the “correct” model, a set of parameter values, initial concentrations, time frames, and a measurement based on the concentration of the chemical species are chosen; then, a simulation of the model is generated and the artificial measurements are produced, with an introduced Gaussian error with mean zero and standard deviation 0.01. After that, all information used to generate data, except initial concentrations, is discarded and the models are ranked based on the simulated measurements and prior distributions only.

We performed two experiments, the first is presented in [VG07] and consists of a common structure in signal transduction; the second, created for this project, consists of a simpler network, containing a few enzymatic reactions and other first order chemical interactions.

Even though we designed the experiment for both software ABC-SysBio and SigNetMS, we will only present experiments performed on ABC-SysBio, since experiments of SigNetMS are still being performed and analyzed.

4.3.1 The First Experiment

The four candidate models of the first experiment are presented in figure 4.1. The model 1, represented on figure 4.1(a) is used to generate artificial experimental data for which all models will be ranked. This network has as components the degradation of S into dS with reaction rate constant k_1; a reversible second order reaction $\mathrm{R}+\mathrm{S} \rightleftharpoons \mathrm{RS}$ with forward rate constant k_2 and reverse rate constant of k_3; a first order reaction $\mathrm{RS} \longrightarrow \mathrm{R_{pp}}$; and a Michaelis-Menten (MM) reaction $\mathrm{R_{pp}} \longrightarrow \mathrm{R}$ for which the reaction rate constant and the omitted enzyme concentration are combined into one single parameter V. Model 2, on figure 4.1(b) is a simplification of model 1 in which the enzymatic reaction of phosphorilation of R is reformulated as a MM reaction with enzyme S; the other MM reaction, $\mathrm{R_{pp}} \longrightarrow \mathrm{R}$, now has speed constant V and MM constant k_3. Model 3, on figure 4.1(c), is a simplification of model 2 because it neglects the degradation of the chemical species S into dS. The model 4, on figure 4.1(d), is a more complex version of model 1, in which the dephosphorylation of R_{pp} is not simplified as a MM reaction.

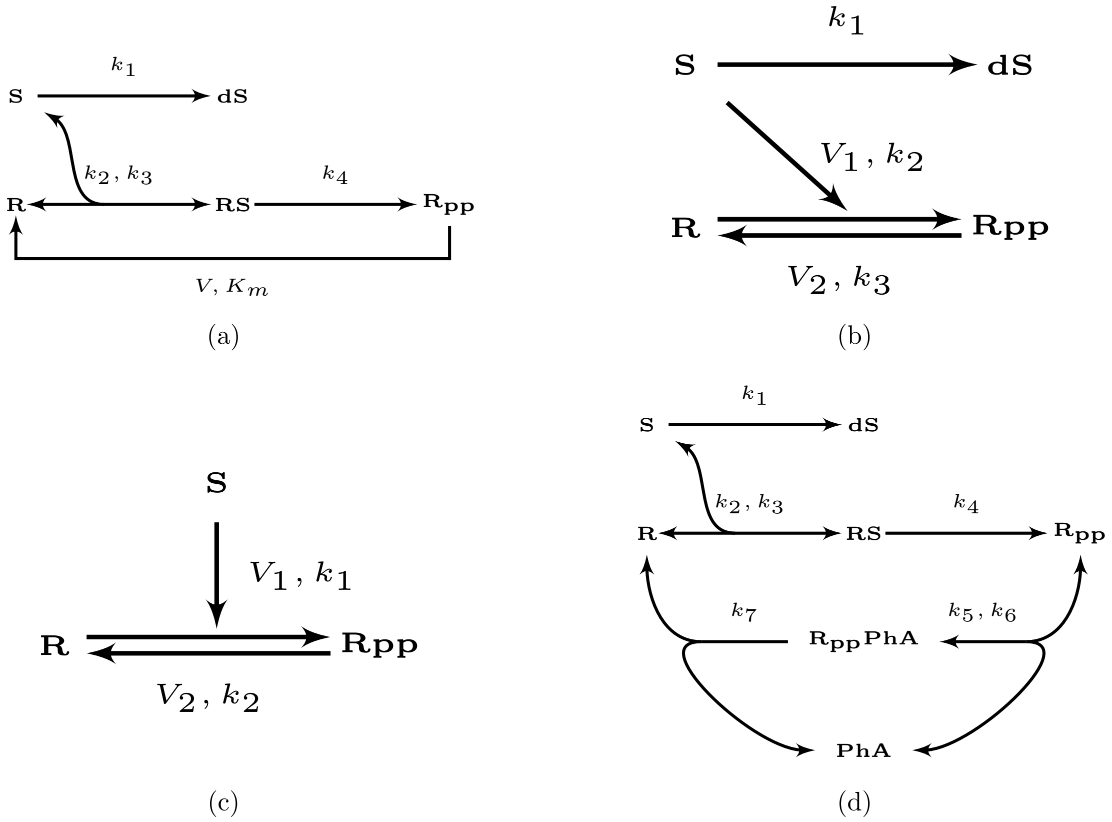


Figure 4.1: The four models used on the first experiment. The experimental measurement on those models is the concentration of the chemical species R_{pp} over time. Model 4.1(a) was used to generate artificial experimental data for which those models are ranked. Model 4.1(b) is a simplification of the first model in which the phosphorylation of R to R_{pp} was simplified using the Michaelis-Menten kinetics. Model 4.1(d) is a more complex version of the first model. Finally, model 4.1(c) is a very simplified version of the first model, because it neglects the degradation of S.

The experimental measurement chosen for the hypothesis is the concentration of R_{pp} on the time steps: 2s, 5s, 10s, 20s, 40s, 60s, and 100s. The parameter values chosen for simulation are: $k_1 = 0.07$, $k_2 = 0.6$, $k_3 = 0.05$, $k_4 = 0.3$, $V = 0.017$, and $K_m = 0.3$. The initial concentrations are: S = 1, R = 1, dS = 0, RS = 0, $\mathrm{R}_{pp} = 0$. TODO: Why can I omit the unit of concentrations and reaction rate? The model is then simulated and a Gaussian error with mean zero and standard deviation 0.01 is added to the simulation data; this procedure is repeated three times and hence produces three experimental observations. Figure 4.2 shows one of the experimental observations.

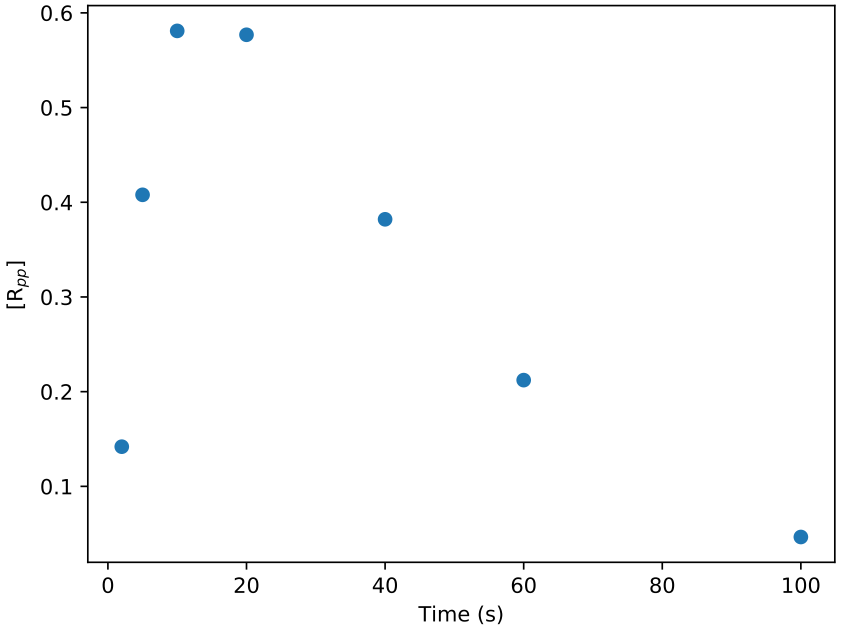
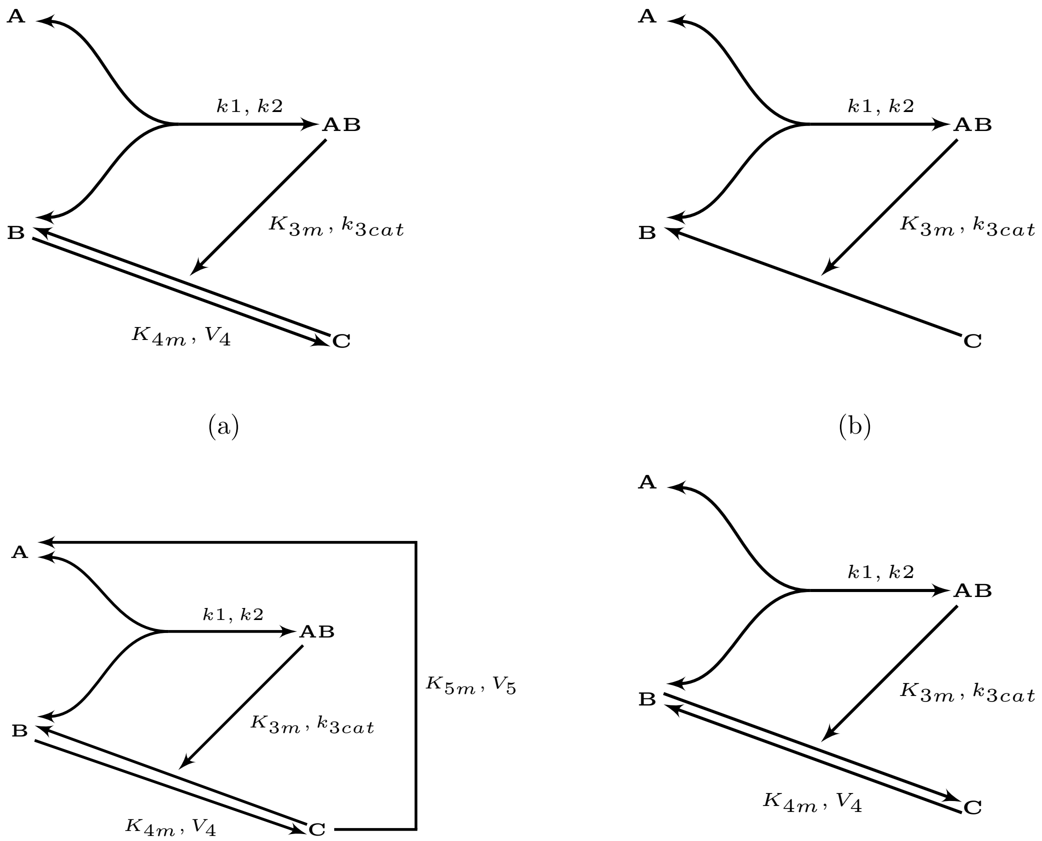


Figure 4.2: Artificial experimental observation generated with model 1.

4.3.2 The Second Experiment

The second experiment was created for this project and it consists in a problem with simpler hypothesis than experiment 1. The four candidate models are presented in figure 4.3. The model 1 is the one used to generate the artificial experimental data. This model is presented on figure 4.3(a) and it is composed by a reversible reaction $A + B \rightleftharpoons AB$ with forward rate constant k_1 and reverse rate constant k_2; an enzymatic reaction $B \longrightarrow C$ with Michaelis constant K_4 and with omitted enzyme concentration and catalytic rate constant combined into a single parameter $V5$; and another enzymatic reaction $C \longrightarrow B$ with catalytic action of AB. Model 2, presented on figure 4.3(b) is a simplification of the first model that do not consider the enzymatic reaction that transforms B into C. Model 3, presented on figure 4.3(c) is a generalization of the first model that adds an enzymatic reaction that can transform C into A, with Michaelis constant K_{5m} and has catalytic rate constant combined with the omitted enzyme concentration into the parameter V_5. Model 4, presented on figure 4.3(d), is a spurious version of model 1, since it inverts the two enzymatic reactions between B and C.



(a)

(b)

(c)

(d)

Figure 4.3: The four models used on the second experiment. The experimental measurement used for these models is the concentration of the chemical species B. Model 4.3(a) was used to generate artificial experimental data for which these models are ranked. Model 4.3(b) is a simplification of the first model, removing the enzymatic reaction that B into C. Model 4.3(c) is a generalization of the first model, and it is more complex too. Model 4.3(d) inverts the two enzymatic reactions that are happening between B and C, this hypothesis is the control as it should give the worst fit to experimental data.

The simulation measurement used on this experiment is to take the concentration of the chemical species B over the time steps: 0s, 142s, 285s, 428s, 571s, 714s, 857s, and 1000s. We simulated model 1 with parameter values: $k_1 = 1.7e - 4$, $k_2 = 0.4$, $k_{cat3} = 2$, $K_{3m} = 1.43e3$, $V_4 = 1$ and $K_{4m} = 1.07e2$; the initial concentrations used were: $A = 200$, $B = 20$, $AB = 0$, $C = 200$. Again, similarly to what was done on the first experiment, the model 1 generates an artificial measurement to which a Gaussian error, with mean zero and standard deviation 0.01, is added, yielding three observations of the model on the specified time steps. Figure 4.4 shows one of these observations.

4.4 Results

4.4.1 Results on the First Experiment

We performed the first experiment using a setup similar to the work of Vyshemirsky and Girolami (2007) when they presented the Annealing-Melting Integration method [VG07]. In this experiment, we are expected to achieve the same ranking as Vyshemirsky and Girolami,

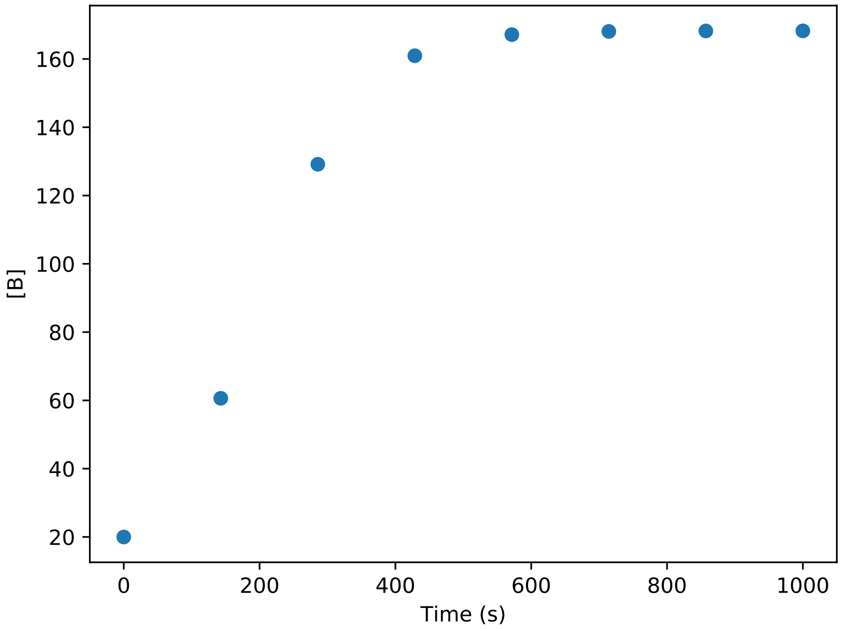


Figure 4.4: Artificial experimental observation generated with model 1 on the second experiment.

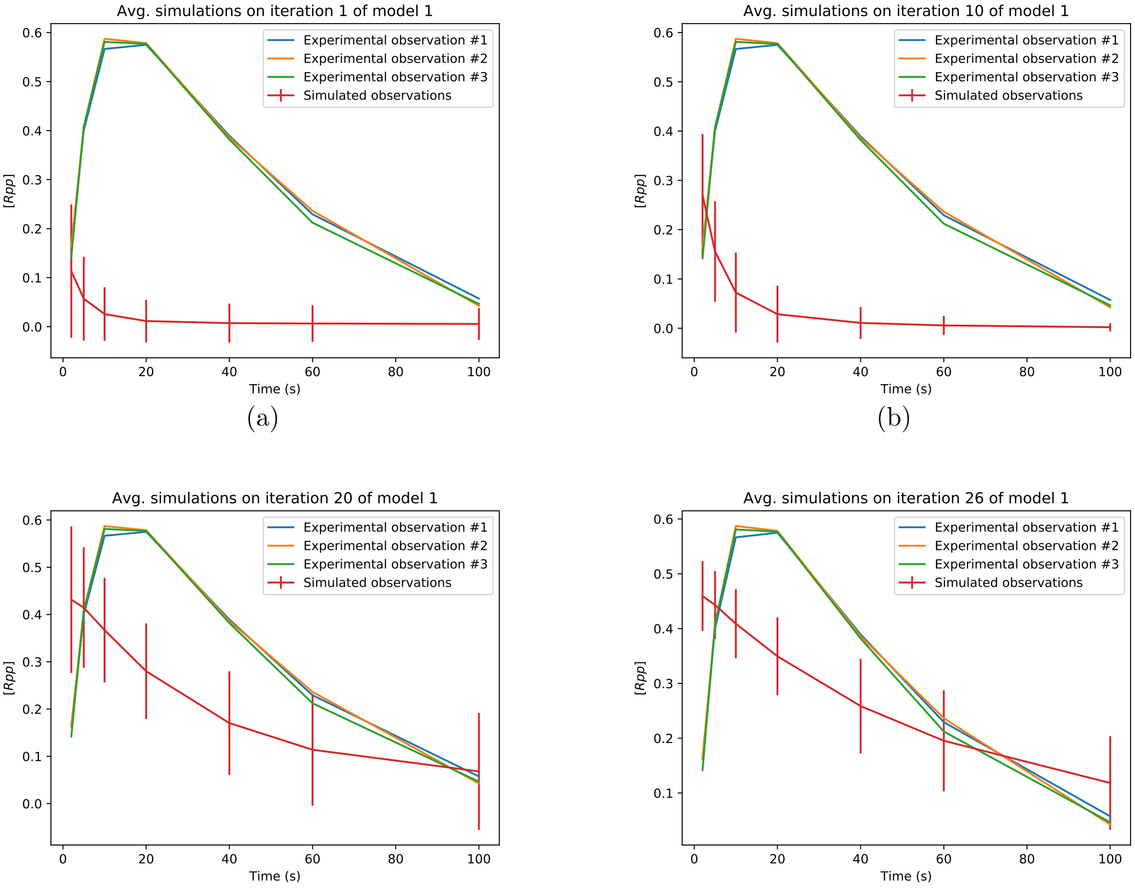
which is 1, 4, 2, 3 from the best to the worst model. We already showed part of the set up by showing the candidate models, the artificial data generation procedure and initial concentrations on section 4.3.1. Now we also define the prior distributions we used.

We started the experiment following the work of Vyshemirsky and Girolami (2007), therefore we defined all model prior parameters as $Gamma(1, 3)$. According to the authors, even though the distribution has a mean ($\mu = 3$), significantly larger than any of the parameters used, the results were satisfactory using their methodology. However, as we will show in the results, using $Lognormal(1, 0.2)$ priors, more concentrated on the real parameters, allowed us to achieve better results on ABC-SysBio.

Results with Gamma priors using ABC-SysBio

We used the ABC-SysBio software using the automatic iteration scheduler, which created 26 populations of parameter values, each of them with 100 individual parameters values. At the last iteration, the algorithm stopped with $\epsilon = 1$ and the following estimates: $\hat{p}(M = 1|D, \epsilon = 1) = 0.005$, $\hat{p}(M = 2|D, \epsilon = 1) = 0.014$, $\hat{p}(M = 3|D, \epsilon = 1) = 0.976$ and $\hat{p}(M = 4|D, \epsilon = 1) = 0.003$. These estimates induces the ranking 3, 2, 1, 4, which is almost the inverted ranking obtained by Vyshemirsky and Girolami: 1, 4, 2, 3.

To understand why this happened, we produced a few graphs that show how the population of parameters evolve as the number of iterations increase. The figures 4.5 and 4.6 show the evolution of the simulations generated by the population of parameters of different iterations, comparing models 1 and 3, which are respectively the true model and the model ranked best by this experiment on ABC-SysBio. We can see on these figures that parameters of further iterations do imply in better simulation for model 1, while for model 3 parameters tend to generate a simulation for which the concentration of R_{pp} stays constant and “in the middle” of the experimental values. To understand why this happens, we also plotted the actual simulations that generated these mean values.



(a) (b) (c) (d)

Figure 4.5: Average simulation generated by model 1 on each iteration indicated. For each iteration, the average simulation is created by taking all individuals that are associated to model 1 and evaluating the model with these parameters.

On figure 4.7 we can see the simulation of all individuals of model 1 and 3 of the populations of iteration 20 and 26. Through the red translucent lines of the graph we are able to see simulations generated by all parameters of a model in a specific iteration. We can observe that there is a clear concentration of simulations with a mean concentration value for model 3, while for model 1 there is just a trend that simulations generate curves that are close to experimental values. More than that, an important information that we should take from this graph is that the number of lines on the graph of a model represent the number of individuals of that model in a certain iteration. With that, we can note that the majority of individuals of the iterations showed on the graphs are from model 3, which explains why this model has greater ranking than model 1.

To understand why there are less parameters of model 1 than there are of model 3 in the last iterations we must recall the dynamics of the ABC-SMC algorithm. To constroy the population of the next iteration, the algorithm takes only the parameters from the current population that were accepted, i.e. parameters that imply in simulations that dist at most some ϵ from the model. Because of that, if a model has few individuals accepted in the current population, then there is a high chance that there will be few individuals of this model in the next population. Therefore, if model 1 had a “bad start”, with few parameters accepted in the early iterations,

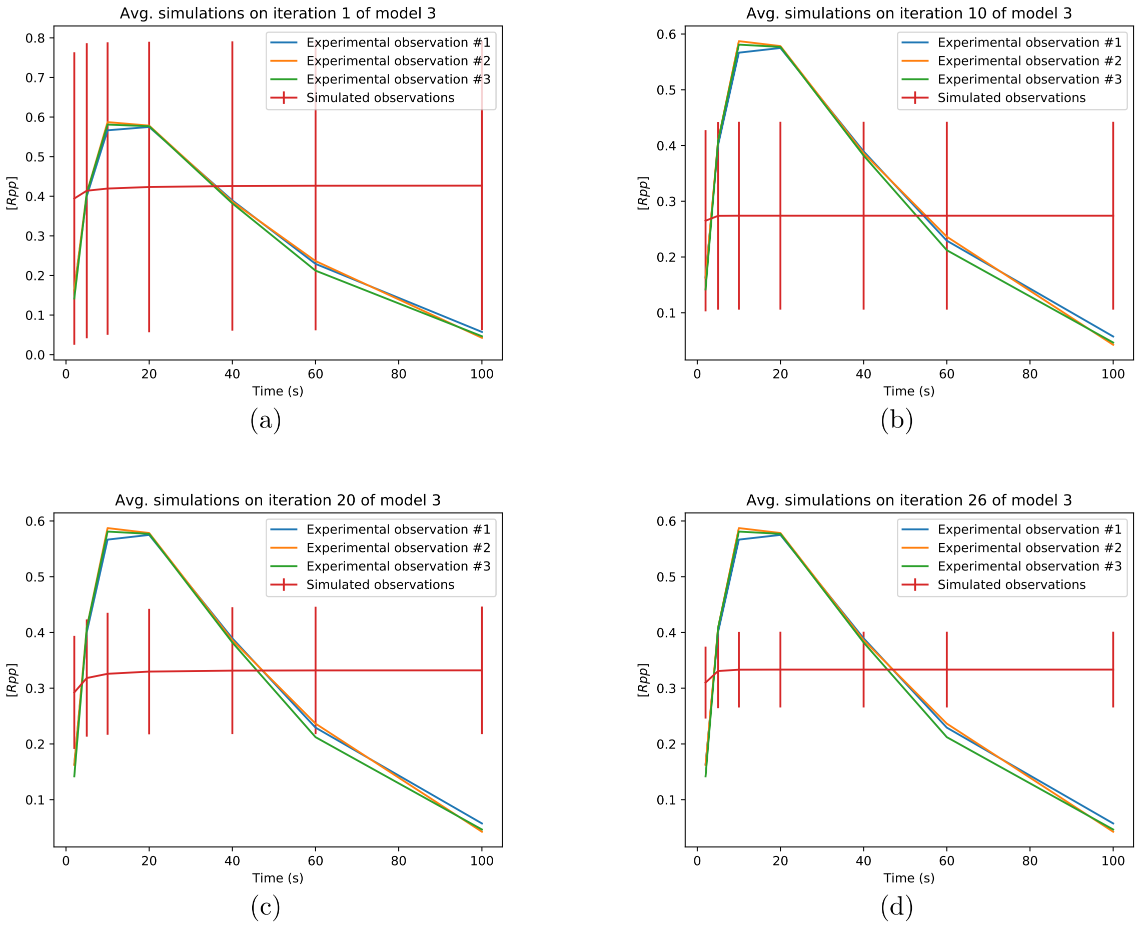
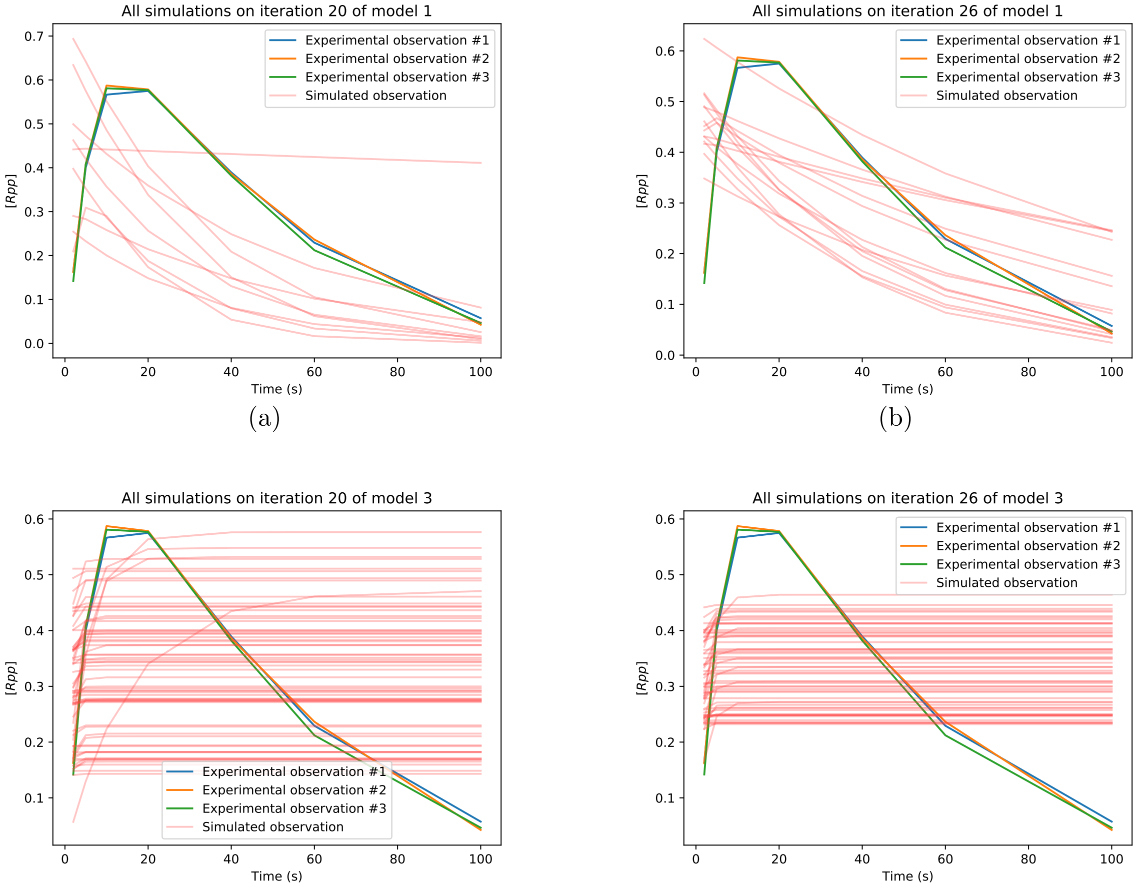


Figure 4.6: Continuation of figure 4.5, but average simulations are from model 3. Note that there is a concentration of the simulations around an intermediary value between a high and low concentration of R_{pp}.

while model 3 had a “good start”, caused by the big ϵ of first iterations, then this can be the reason why model 3 has been ranked first.

The “bad start” of model 1 may have been caused by the prior distribution used, which is not very concentrated around good parameter values, causing the first model to have many bad performing parameters in the first populations. Because of that, we decided to do this experiment again on ABC-SysBio, but using priors $Lognormal(1, 0.2)$.



(a) (b)

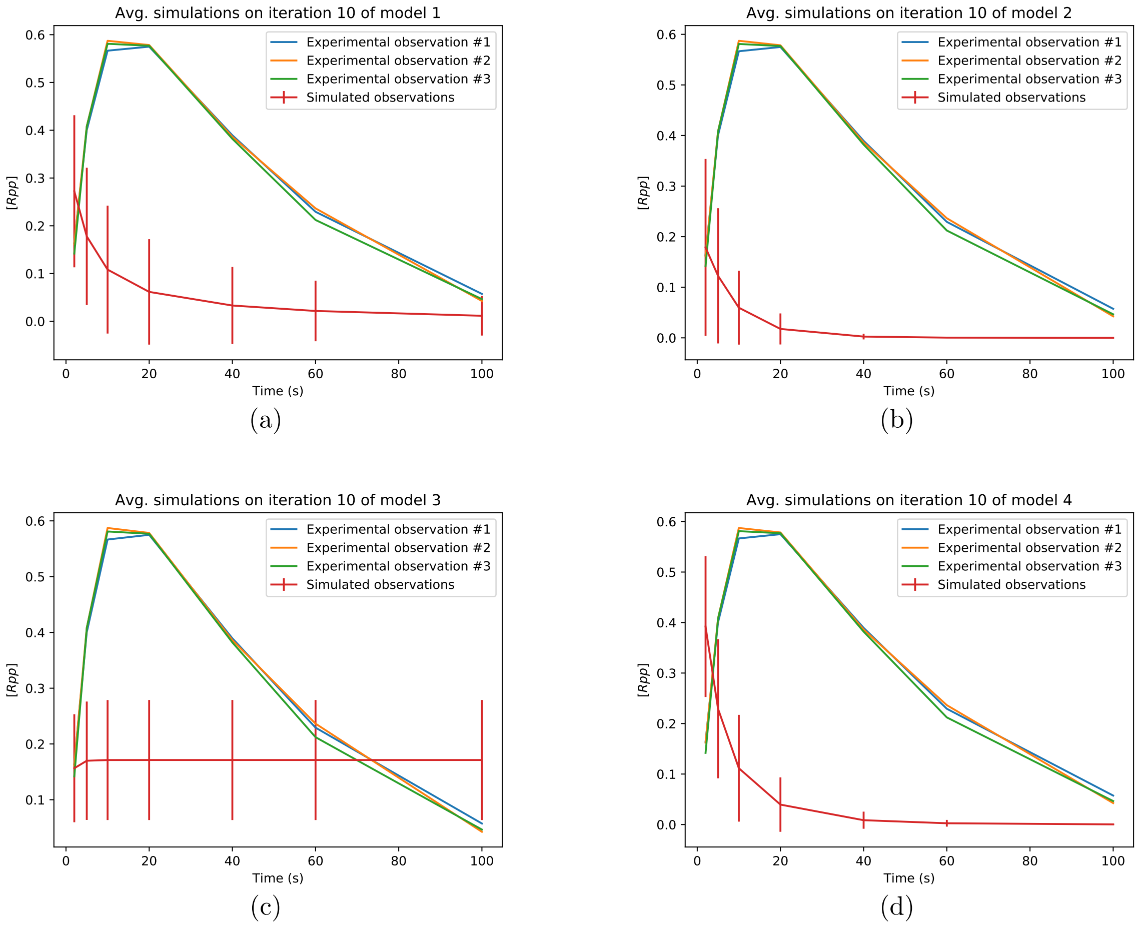
(c) (d)

Figure 4.7: Simulations generated by parameters of iteration 20 and iteration 26 of the ABC-SysBio run. Red translucid lines show the simulation of a single parameter. Stronger red color means overlapping points between simulations. Figures 4.7(a) and 4.7(d) show individuals of model 1 while figures 4.7(c) and 4.7(d) show individuals of model 3.

Results with Lognormal priors using ABC-SysBio

With the goal of investigating the behaviour of ABC-SysBio when using a prior more concentrated on real parameter values, we decided to run the experiment again using parameter priors distributed as $Lognormal(1, 0.2)$. We again ran ABC-SysBio using the automatic scheduler with population of size 100. The algorithm stopped after 32 iterations, with final epsilon $\epsilon = 1$, however on iteration 23 the only model present on the populations were from model 1. The resulting ranking for this experiment was 1, 4, 3, 2; very similar to the ranking presented by Vyshemirsky and Girolami (2007): 1, 4, 2, 3.

The last algorithm iteration that had individuals from all models was the 10th iteration. Figure 4.8 shows the average simulations of all models considering the parameter population of iteration 10. We can see that models 1 and 4 perform better than models 2 and 3. On the 11th population, no parameter of the produced population will be from model 2, and after the 12th population, no parameter will be from models 3 either. Then, from iteration 12 to 23, only parameters of models 1 and 4 will be on the populations. Model 1 is considered the best fit, as it should be by the design of the experiment.



(a)

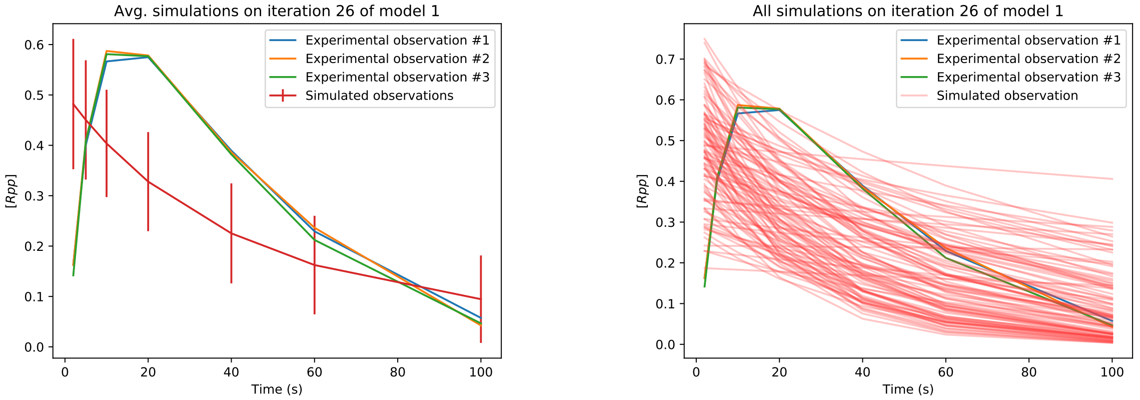
(b)

(c)

(d)

Figure 4.8: Average simulation of iteration 10 for all candidate models.

Even tough the ranking produced by the algorithm is reasonably close to the one produced by Vyshemirsky and Girolami, we decided to plot new graphs to investigate if the parameters of the populations for model 1 actually bring the simulated results close to the experiments. On figure 4.9 we present the average and individual simulations of model 1 produced by the parameters of this model on iteration 26. We can see through the average simulation that the mean value for each time step is not close to the experimental value, and more than that, we can note on the individual simulations plot that the dynamics induced by the parameters are not similar to the experimental dynamics. While in the experiments there is a increase and then decrease of concentration of R_{pp}, in the dynamics induced by the found parameters the trend is to have only decrease of concentrations of R_{pp} over time. We also plotted an estimation of the posterior distribution of parameters of model one, presented on figure 4.10. This estimative was created with parameters of model 1 on iteration 26 of the algorithm, and it is possible to see that the estimated posterior is not concentrated around the expected values, which should be the parameters values used to create the experimental data.



(a)

(b)

Figure 4.9: Average simulation and individual simulation generated by the parameters of model 1 on the 26th iteration.

4.4.2 Results on the Second Experiment

For the second experiment we used only Gamma priors for the model parameters. A complete definition of the priors is presented on table 4.1. We used the ABC-SysBio software to perform the model ranking using the automatic ϵ scheduler with a population size of 100 individuals per iteration. As a result, the ranking produced by the software was: 2, 1, 4, 3.

Parameter	Models	Prior
$k1$	1, 2, 3, 4	$Gamma(1, 0.01)$
$k2$	1, 2, 3, 4	$Gamma(2, 0.5)$
k_{3cat}	1, 2, 3, 4	$Gamma(4, 1)$
K_{3m}	1, 2, 3, 4	$Gamma(2, 1500)$
V_4	1, 3, 4	$Gamma(2, 1)$
K_{4m}	1, 3, 4	$Gamma(2, 100)$
V_5	3	$Gamma(2, 0.4)$
K_{5m}	3	$Gamma(2, 100)$

Table 4.1: Prior distribution for each parameter of models of the second experiment.

The algorithm created 64 iterations until it reached the last iteration with $\epsilon = 31.03$. After iteration 26, no individuals from the populations were from model 3, considered the worst model. After iteration 56, no individuals from model 4 were in the populations. Then, from iterations 56 to 64 only individuals from model 1 and 2 were in the populations. In the last population, the estimated probabilities created by the software were: $\hat{p}(M = 1|D, \epsilon = 31.03) = 0.44$ and $\hat{p}(M = 2|D, \epsilon = 31.03) = 0.56$. Therefore, the result is inconclusive for which model performs better, 1 or 2.

On figure 4.11 we show the average of the simulations produced by parameters of iteration 20. It is possible to see that model 3 indeed performs worse comparing to all other models, and more than that, the curve produced by model 3 is different from the experimental data and curves produces by other models; while in model 3 the concentration of B increases and then decreases, for the other models the concentration increases until it stabilizes. On figure 4.12

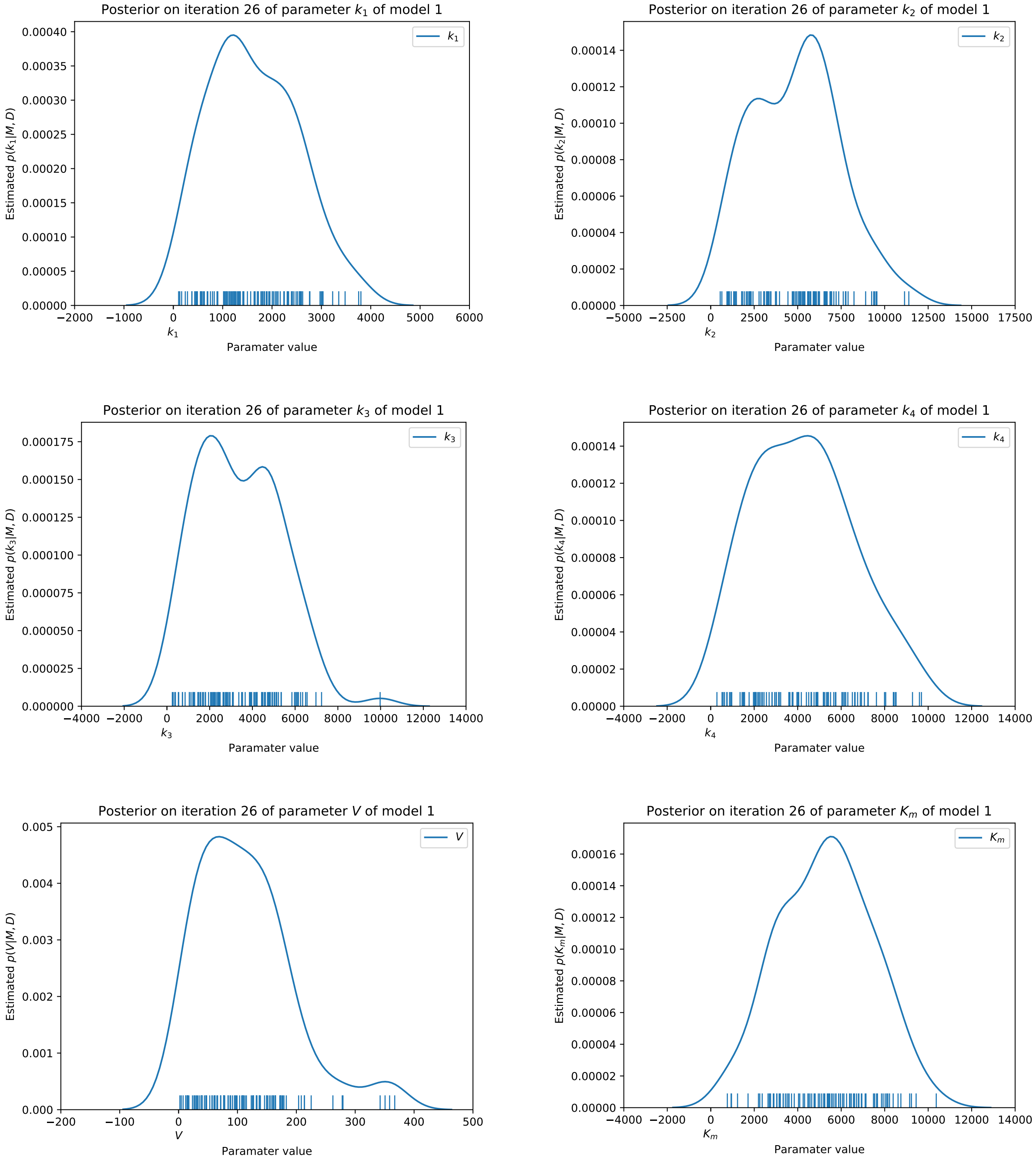


Figure 4.10: Estimates of the posterior distribution of model 1 parameters. These estimates were created using the kernel density estimation method from the seaborn Python package. The parameter name on the abscissa of the graph indicates the location of the correct value of the parameter, i.e. the value used to create the experimental data. The estimated posterior is not concentrated around the correct parameter values, which are $k_1 = 0.07$, $k_2 = 0.6$, $k_3 = 0.05$, $k_4 = 0.3$, $V = 0.017$.

we show the average simulation of parameters of the 50th iteration of the algorithm. We can see that models 1 and 2 approximates the curve reasonably well while model 4, in the other hand, has more error, specially in the time steps of 200 and 400 seconds. Finally, on figure 4.13 we show the average simulations of models 1 and 2 on the 64th iteration of the algorithm. We can see that parameters of the last population bring both models simulations very close to the experimental data.

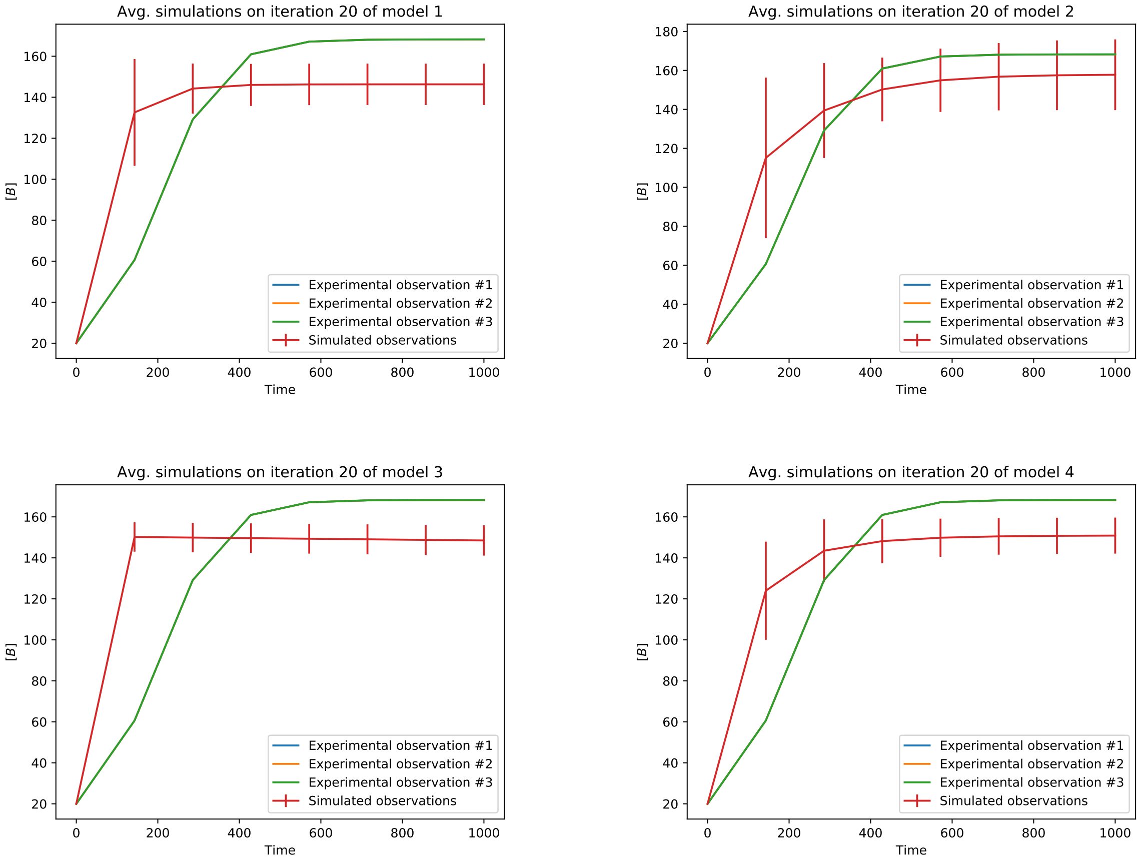


Figure 4.11: Average simulation of iteration 20 for all candidate models.

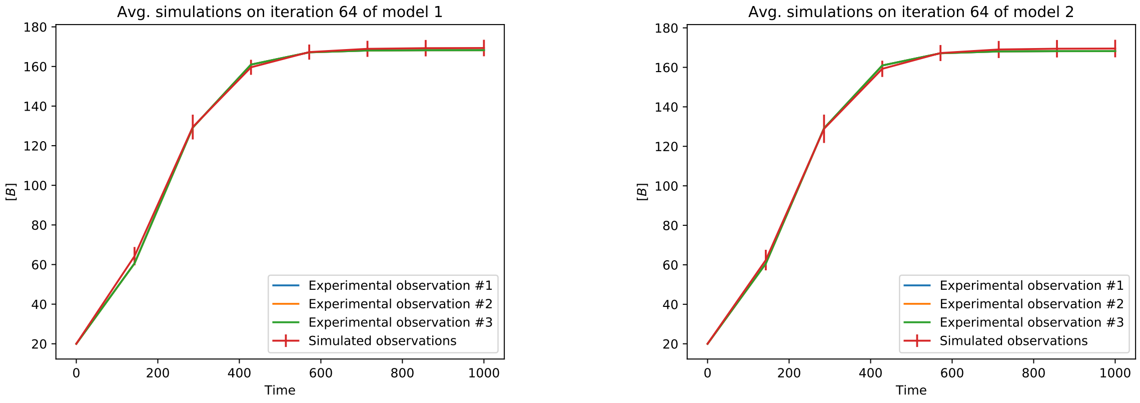


Figure 4.13: Average simulation of iteration 20 for all candidate models.

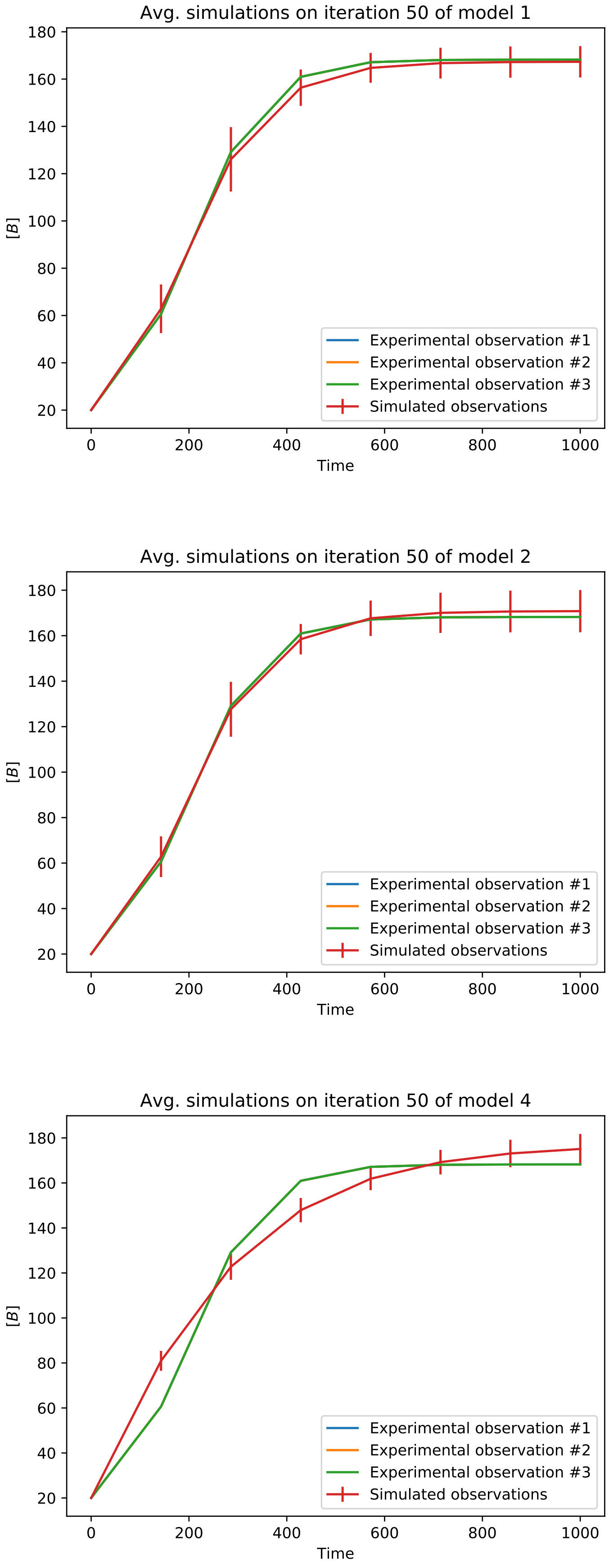


Figure 4.12: Average simulation of iteration 50 for all candidate models.

Since we achieved good results on the simulations of the model 1, we also plotted the estimation of the posterior distribution of the parameters given the data. Figure 4.14 shows the estimated posterior distribution for each parameter. We can note that the posterior distributions estimated have high density around the true parameter values, i.e. parameter values used to create the artificial experimental data.

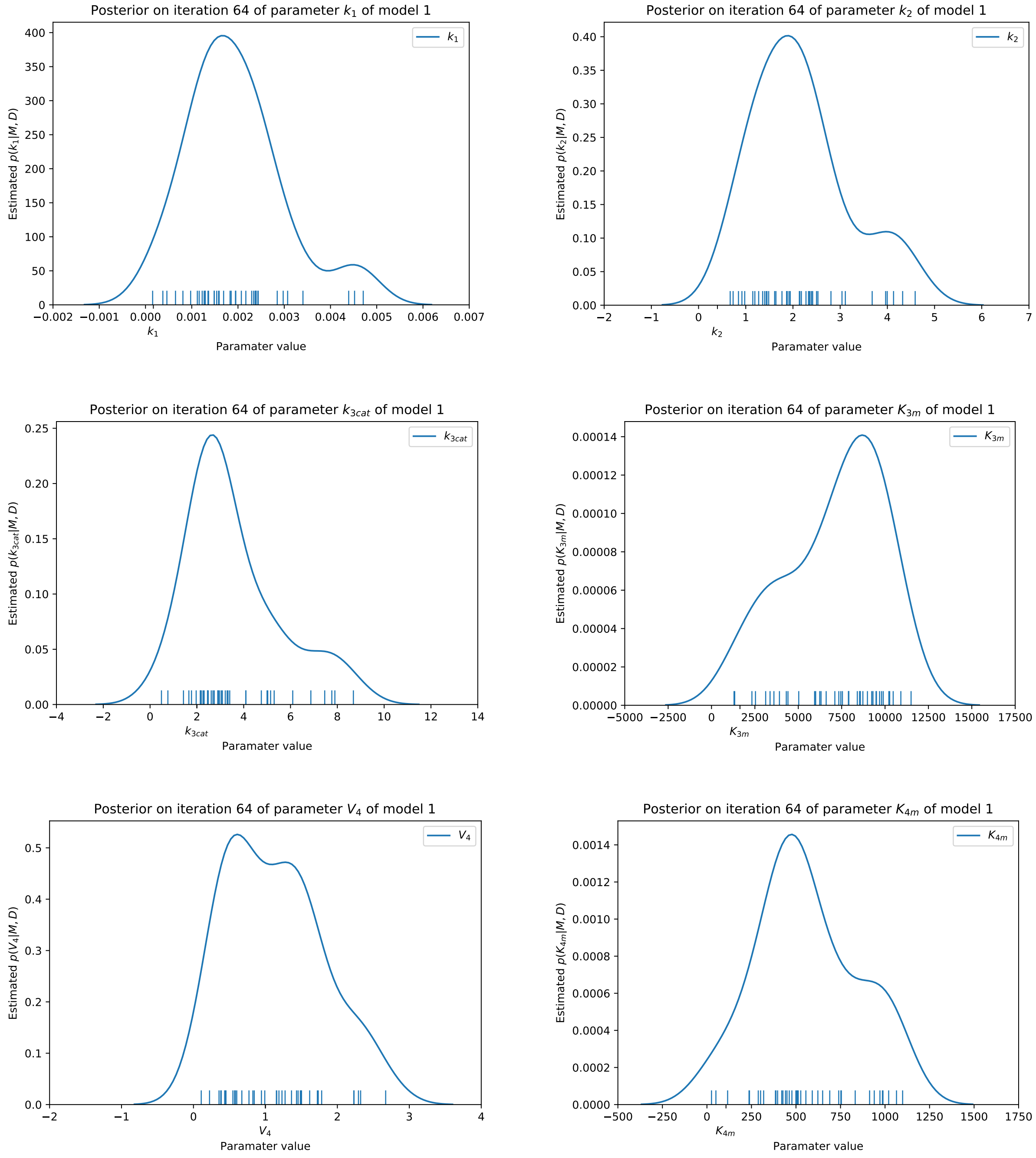


Figure 4.14: Estimates of the posterior distribution of model 1 parameters. These estimates were created using the kernel density estimation method from the seaborn Python package. The parameter name on the abscissa of the graph indicates the location of the correct value of the parameter, i.e. the value used to create the experimental data. The estimated posterior is not concentrated around the correct parameter values, which are $k_1 = 1.7e-4$, $k_2 = 0.4$, $k_{cat3} = 2$, $K_{3m} = 1.43e3$, $V_4 = 1$ and $K_{4m} = 1.07e2$.

Chapter 5

Future Activities

Bibliography

[BH25] George Edward Briggs and John Burdon Sanderson Haldane. "A Note on the Kinetics of Enzyme Action". In: *Biochemical Journal* 19.2 (1925), pp. 338–339. DOI: 10.1042/bj0190338. URL: https://doi.org/10.1042/bj0190338.

[FP08] N. Friel and A. N. Pettitt. "Marginal likelihood estimation via power posteriors". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 70.3 (July 2008), pp. 589–607. DOI: 10.1111/j.1467-9868.2007.00650.x. URL: https://doi.org/10.1111/j.1467-9868.2007.00650.x.

[Gel+13] Andrew Gelman, John B. Carlin, Hal S. Stern, David B. Dunson, Aki Vehtari, and Donald B. Rubin. *Bayesian Data Analysis, Third Edition*. Chapman and Hall/CRC, Nov. 2013. DOI: 10.1201/b16018. URL: https://doi.org/10.1201/b16018.

[KG00] Minoru Kanehisa and Susumu Goto. "KEGG: kyoto encyclopedia of genes and genomes". In: *Nucleic Acids Research* 28.1 (2000), pp. 27–30.

[Lie+10] J. Liepe, C. Barnes, E. Cule, K. Erguler, P. Kirk, T. Toni, and M. P. H. Stumpf. "ABC-SysBio–approximate Bayesian computation in Python with GPU support". In: *Bioinformatics* 26.14 (June 2010), pp. 1797–1799. DOI: 10.1093/bioinformatics/btq278. URL: https://doi.org/10.1093/bioinformatics/btq278.

[Lie+14] Juliane Liepe, Paul Kirk, Sarah Filippi, Tina Toni, Chris P Barnes, and Michael P H Stumpf. "A framework for parameter estimation and model selection from experimental data in systems biology using approximate Bayesian computation". In: *Nature Protocols* 9.2 (Jan. 2014), pp. 439–456. DOI: 10.1038/nprot.2014.025. URL: https://doi.org/10.1038/nprot.2014.025.

[LN+06] Nicolas Le Novere, Benjamin Bornstein, Alexander Broicher, Melanie Courtot, Marco Donizelli, Harish Dharuri, Lu Li, Herbert Sauro, Maria Schilstra, Bruce Shapiro, et al. "BioModels Database: a free, centralized database of curated, published, quantitative kinetic models of biochemical and cellular systems". In: *Nucleic Acids Research* 34.suppl_1 (2006), pp. D689–D691.

[Mar+03] P. Marjoram, J. Molitor, V. Plagnol, and S. Tavare. "Markov chain Monte Carlo without likelihoods". In: *Proceedings of the National Academy of Sciences* 100.26 (Dec. 2003), pp. 15324–15328. DOI: 10.1073/pnas.0306899100. URL: https://doi.org/10.1073/pnas.0306899100.

[Mil+09] Ron Milo, Paul Jorgensen, Uri Moran, Griffin Weber, and Michael Springer. "BioNumbers—the database of key numbers in molecular and cell biology". In: *Nucleic Acids Research* 38.suppl_1 (Oct. 2009), pp. D750–D753. DOI: 10.1093/nar/gkp889. URL: https://doi.org/10.1093/nar/gkp889.

[NR93] Michael A. Newton and Adrian E. Raftery. “Approximate Bayesian Inference with the Weighted Likelihood Bootstrap”. In: *Journal of the Royal Statistical Society. Series B (Methodological)* 56.1 (1993), pp. 3–48.

[Pri+99] J. K. Pritchard, M. T. Seielstad, A. Perez-Lezaun, and M. W. Feldman. “Population growth of human Y chromosomes: a study of Y chromosome microsatellites”. In: *Molecular Biology and Evolution* 16.12 (Dec. 1999), pp. 1791–1798. DOI: 10.1093/oxfordjournals.molbev.a026091. URL: https://doi.org/10.1093/oxfordjournals.molbev.a026091.

[Rei+17a] Marcelo S. Reis, Vincent Noël, Matheus H. Dias, Layra L. Albuquerque, Amanda S. Guimarães, Lulu Wu, Junior Barrera, and Hugo A. Armelin. “An Interdisciplinary Approach for Designing Kinetic Models of the Ras/MAPK Signaling Pathway”. In: *Methods in Molecular Biology*. Springer New York, 2017, pp. 455–474. DOI: 10.1007/978-1-4939-7154-1_28. URL: https://doi.org/10.1007/978-1-4939-7154-1_28.

[Rei+17b] Marcelo S. Reis, Gustavo Estrela, Carlos Eduardo Ferreira, and Junior Barrera. “featsel: A framework for benchmarking of feature selection algorithms and cost functions”. In: *SoftwareX* 6 (2017), pp. 193 –197. ISSN: 2352-7110. DOI: https://doi.org/10.1016/j.softx.2017.07.005. URL: http://www.sciencedirect.com/science/article/pii/S2352711017300286.

[Sch04] I. Schomburg. “BRENDA, the enzyme database: updates and major new developments”. In: *Nucleic Acids Research* 32.90001 (Jan. 2004), pp. 431D–433. DOI: 10.1093/nar/gkh081. URL: https://doi.org/10.1093/nar/gkh081.

[Szk+10] D. Szklarczyk et al. “The STRING database in 2011: functional interaction networks of proteins, globally integrated and scored”. In: *Nucleic Acids Research* 39.Database (Nov. 2010), pp. D561–D568. DOI: 10.1093/nar/gkq973. URL: https://doi.org/10.1093/nar/gkq973.

[Ton+09] Tina Toni, David Welch, Natalja Strelkowa, Andreas Ipsen, and Michael P.H Stumpf. “Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems”. In: *Journal of The Royal Society Interface* 6.31 (Feb. 2009), pp. 187–202. DOI: 10.1098/rsif.2008.0172. URL: https://doi.org/10.1098/rsif.2008.0172.

[VG07] Vladislav Vyshemirsky and Mark A. Girolami. “Bayesian ranking of biochemical system models”. In: *Bioinformatics* 24.6 (Dec. 2007), pp. 833–839. DOI: 10.1093/bioinformatics/btm607. URL: https://doi.org/10.1093/bioinformatics/btm607.

[Whi71] A.W. Whitney. “A Direct Method of Nonparametric Measurement Selection”. In: *IEEE Transactions on Computers* C-20.9 (Sept. 1971), pp. 1100–1103. DOI: 10.1109/t-c.1971.223410. URL: https://doi.org/10.1109/t-c.1971.223410.

[Wit+11] U. Wittig et al. “SABIO-RK–database for biochemical reaction kinetics”. In: *Nucleic Acids Research* 40.D1 (Nov. 2011), pp. D790–D796. DOI: 10.1093/nar/gkr1046. URL: https://doi.org/10.1093/nar/gkr1046.

[Xu+10] Tian-Rui Xu et al. “Inferring Signaling Pathway Topologies from Multiple Perturbation Measurements of Specific Biochemical Species”. In: *Science Signaling* 3.113 (2010), ra20–ra20. ISSN: 1945-0877. DOI: `10.1126/scisignal.2000517`. eprint: `http://stke.sciencemag.org/content/3/113/ra20.full.pdf`. URL: `http://stke.sciencemag.org/content/3/113/ra20`.

[Wu15] Lulu Wu. “Um método para modificar vias de sinalização molecular por meio de análise de banco de dados de interatomas”. MA thesis. University of Sao Paulo, 2015.