Abstractions of Biochemical Reaction Networks

Andreea Beica

PSL
UNIVERSITÉ PARIS

THÈSE DE DOCTORAT
DE L'UNIVERSITÉ PSL

Préparée à l'École Normale Supérieure

Abstractions of Biochemical Reaction Networks

Soutenue par
Andreea BEICA
Le 12 juin 2019

École doctorale n°386
Sciences mathématiques de Paris centre

Spécialité
Informatique

Composition du jury :

M. Gilles BERNOT CNRS & Univ. Nice-Sophia Antipolis	*Rapporteur*
Mme. Anne SIEGEL INRIA, CNRS & Univ. Rennes	*Rapporteur*
M. Jérôme FERET INRIA & ENS	*Examinateur*
M. Wolfram LIEBERMEISTER Inst. Nat. de la Recherche Agronomique	*Examinateur*
Mme. Tatjana PETROV Konstanz University	*Examinateur*
M. David SAFRANEK Masaryk University	*Examinateur*
M. Vincent DANOS CNRS & ENS	*Directeur de thèse*

ENS | PSL

Résumé

Cette thèse vise á étudier deux aspects liés á la modélisation des Réseaux de Réactions Biochimiques.

Dans un premier temps, nous montrons comment la séparation des échelles de temps et de concentration dans les systèmes biologiques peut être utilisée pour la réduction de modèles. Nous proposons l'utilisation des modèles par régles de réécriture pour le prototypage de circuits génétiques, puis nous exploitons le caractère multi-échelle de tels systèmes pour construire une méthode générale d'approximation de modèles. La réduction est effectuée via une analyse statique du système de règles. Notre heuristique de réduction repose sur des justifications physiques solides. Cependant, tout comme pour d'autres techniques de réduction de modèles exploitant la séparation des échelles, on note la manque de méthodes précises pour quantifier l'erreur d'approximation, tout en évitant de résoudre le modèle original.

C'est pourquoi nous proposons ensuite une méthode d'approximation dans laquelle les garanties de réduction représentent l'exigence majeure. Cette seconde méthode combine abstraction et approximation numérique, et vise á fournir une meilleure compréhension des méthodes de réduction de modèles basées sur une séparation des échelles de temps et de concentration.

Dans la deuxième partie du manuscrit, nous proposons une nouvelle technique de reparamétrisation pour les modèles d'équations différentielles des réseaux biochimiques, afin d'étudier l'effet des stratégies de stockage de ressources intracellulaires sur la croissance, dans des modèles mécanistiques d'auto-réplication cellulaire. Enfin, nous posons des bases pour la caractérisation de la croissance cellulaire en tant que propriété émergeante d'une nouvelle sémantique des réseaux de Petri modélisant des réseaux de réactions biochimiques.

Abstract

This thesis aims at studying two aspects related to the modelling of Biochemical Reaction Networks, in the context of Systems Biology.

In the first part, we analyse how scale-separation in biological systems can be exploited for model reduction. We first argue for the use of rule-based models for prototyping genetic circuits, and then show how the inherent multi-scaleness of such systems can be used to devise a general model approximation method for rule-based models of genetic regulatory networks. The reduction proceeds via static analysis of the rule system.

Our method relies on solid physical justifications, however not unlike other scale-separation reduction techniques, it lacks precise methods for quantifying the approximation error, while avoiding to solve the original model. Consequently, we next propose an approximation method for deterministic models of biochemical networks, in which reduction guarantees represent the major requirement. This second method combines abstraction and numerical approximation, and aims at providing a better understanding of model reduction methods that are based on time- and concentration- scale separation.

In the second part of the thesis, we introduce a new re-parametrisation technique for differential equation models of biochemical networks, in order to study the effect of intracellular resource storage strategies on growth, in self-replicating mechanistic models. Finally, we aim towards the characterisation of cellular growth as an emergent property of a novel Petri Net model semantics of Biochemical Reaction Networks.

Acknowledgments

My first thanks go to Vincent Danos, for accepting to supervise my work during these three years. Thank you for advising me, and for cultivating a research environment based on freedom of choice and scientific curiosity.

I would like to thank my reviewers Anne Siegel and Gilles Bernot, for accepting the time consuming task of reading this manuscript and writing reports. Thank you for your time and for your valuable feedback. I also wish to express my gratitude towards the other members of my jury, for accepting to participate in my PhD defense.

My warmest thanks go to my co-authors, who provided me with invaluable help during all stages of the research and writing process: Jérôme Feret, Cãlin Guet, Tanja Petrov and Guillaume Terradot. Thank you Jérôme for your availability and your sustained indispensable feedback and help. Thank you Cãlin for the ever-stimulating discussions and for repeatedly welcoming me into your lab. Thank you Tanja for your constant support, both scientifically and personally. Thank you Guillaume for all the stimulating exchanges and for always being willing to share your impressive biological knowledge with me.

I am also thankful towards the rest of the permanent members of the Antique team: thank you Xavier for sustainedly helping me navigate the meanders of administrative processes, thank you Cezara for your much-needed advice and support. A warm thank you to the rest of the team: Caterina, Ferdinanda, Gaëlle, Guillaume, Huisong, Ilias, Jiangchiao, Lý Kim, Marc, Nicolas, Patric, Pierre, Stan.

I am very grateful towards my circle of friends, for their constant and unconditional support in my efforts towards this PhD: thank you Andreea, Alex, Coralie, Ioana, Ken, Laurent, Léa, Lola, Megumi, Raphaël. A special “thank you” goes to Kevin.

Most importantly, I would like to thank my parents, for giving me a taste for knowledge and for science, and for their constant support and encouragement throughout these years.

This thesis is dedicated to my grandmother.

Contents

Introduction

Computer Science and Systems Biology

When delving into the history of Computer Science, one finds that the first design for a modern computer is widely accepted to be the *Analytical Engine*: a *programmable* mechanical calculator invented in 1837, by Charles Babbage. Its invention date considerably pre-dates that of the modern digital computer, and hints at the initial goal of the field of study: *creating mechanical devices that automate mathematical calculations*. The validity of this objective is reinforced by the consensus that the earliest foundations of Computer Science can be found in ancient mechanical computation tools such as the abacus, the Antikythera mechanism, or the mechanical analog computer devices of the medieval Islamic world. However, in the last century, Computer Science has evolved from its initial calculation-related task of translating a mathematical language model into a computer program simulating it, to become a central component of both a vast range of scientific areas and of mundane existence.

It can be argued that the main reason for which the field has become ubiquitous to modern life lays in its evolution from being primarily a "science of calculation", to also becoming a "science of models". In this sense, one notes the recurring development of new domain-specific programming languages that are able to directly model a vast range of processes.

An indicator of the growing relevance of Computer Science in the area of scientific modelling is the emergence of interdisciplinary fields such as Systems Biology, computational biology, bioinformatics, computational linguistics, artificial intelligence, cognitive science, or computational social science, to name a few.

The work presented in this thesis aims at developing new modeling and model analysis tools for Systems Biology, in order to tackle two of the challenges specific to the field. The first such issue deals with model reduction techniques and their guarantees, while the second one addresses the mathematical modeling and analysis of specific biological behaviors such as describing cellular growth as an emergent system property, as well as how it is impacted by cellular resource storage strategies.

As the science that studies living systems, biology is concerned with their constituents at all scales: molecule, cell, tissue, organ, individual, organism, and ecosystem. As illustrated by existing biological classifications and taxonomies, the main activities of the discipline traditionally had a prominent *qualitative* component. However, the advent of novel experimental techniques that allowed researchers to simultaneously observe the behaviours of large numbers

of distinct molecular species, in the beginning of the 21^{st} century, has triggered a shift toward the *quantitative* side [36], and with it has seen the emergence of new approach to biology, in which Computer Science plays a key role. The resulting interdisciplinary research field, *Systems Biology*, aims at building an *integrated* physiology of systems and a *predictive* understanding of the whole, by focusing on the *data-centric quantitative modeling of biological processes and systems* [173]. In other words, intracellular processes are investigated as *dynamic systems*. This new outlook on biology is made possible by recent technological advances that enable both molecular observations on far more inclusive scales than previously achievable, but also allow for computational analysis of such observations.

Indeed, the last two decades have seen biological research become a data intensive science, due to the ever-increasing flow of data resulting from the ability to make comprehensive measurements on DNA sequence, gene expression profiles, or protein-protein interactions (to name a few). This, in turn, increases the role of computers and computer science in biology.

On the one hand, the considerable amount of vast information generated by new experimental techniques such as DNA microarrays and genome sequencers[1], as well as other large-scale technologies, exceeds human analysis capacity, and thus requires computation power for storing, processing, analyzing and understanding the data [98].

On the other hand, the rapidly growing amount of biological data in the public domain reinforces the importance of mathematical and computational models (as well as analysis techniques) of biological systems: measurements alone do not explain the underlying complex molecular mechanisms, therefore appropriate mechanistic theories are needed in order to understand them. In other words, inferring physical or structural interactions in the system from functional data alone is impossible, meaning that dynamical modeling becomes a part of biological reasoning.

As such, the central philosophy of *Systems Biology* is that the traditional approach to biological research, which consists in mapping out the physical components (and their individual interactions) of a biological system, is unsatisfactory for the understanding of emergent properties of a complex biological system, as the latter may be the result of the interplay of simpler, integrated parts of the network. Indeed, even though reductionism has been highly successful in explaining macroscopic phenomena, purely in terms of the constituent parts, the underlying assumptions (that there were few parts that interacted with each other in a simple manner, or that there were many parts but whose interactions could be neglected) are simply not true for many systems of present interest. What's more, the advent of computational techniques has revelead that even relatively small systems of interacting parts (*e.g.*, the Lorenz system) could exhibit very complex behaviour. Biological systems, which are *inherently* complex, must thus be modeled and studied using the *prediction-control-understanding* framework offered by *executable* mathematical models, which are also dubbed *dynamical models*.

Indeed, dynamical models prove to be a crucial component of the modern biologist's toolbox, as via their *simulation* and *predictive* powers, they enable biological investigation in a number of ways [99]:

[1] microarrays permit interrogation of more than one million single-nucleotide polymorphisms (SNPs) at the same time

1. Constructing such models demands a critical consideration of the underlying biological mechanisms, and can thus reveal inconsistencies or previously unnoticed gaps in knowledge.

2. They serve both as a recapitulation of system behavior, as well as a transparent, unequivocally-communicable description of the system.

3. They represent *"working hypotheses"*, which can be used to unambiguously investigate system behavior under conditions that are unachievable in a laboratory: model simulations can be carried out quickly and with no real cost, with every aspect of model behavior being observable at all time points.

4. Model analysis yields insights into why a system behaves the way it does, thus providing links between network structure and behavior.

5. Through simulation, dynamical models allow for hypothesis generation and testing, enabling one to rapidly analyze the effects of manipulating experimental conditions *in silico*. This iterative process leads to a continually improving understanding of the system, in what has been called a "virtuous cycle".

6. Model-based design is also a central component of *synthetic biology*, as models of cellular networks are useful for guiding the choice of components and suggesting the most effective experiments for testing system performance.

7. Other advantages of dynamical modeling include being able to model quantities that are experimentally hidden, and being able to stretch or compress timescales.

What's more, the very complexity of the living matter implies that biologists reason on models rather than on the objects themselves [36]. A "good" dynamical model should concomitantly reflect known behavior of the studied system, contain hypotheses that need to be verified, and be able to predict the system's behaviour in a precise, input-dependent manner.

All in all, Systems Biology aims at a system-level understanding and analysis of biological systems, under the assumption that "the whole is greater than the sum of the parts". As stated in [108], a system-level understanding "requires a set of principles and methodologies that links the behaviors of molecules to system characteristics and functions" and ultimately aims at describing and understanding living entities "at the system level grounded on a consistent framework of knowledge that is underpinned by the basic principles of physics".

One of the recurring terms of the field is that of *"network of networks"*. It is a term that provides meaningful insight into how the Systems Biology approach is different from, and more predictive than, the traditional approach to biology, as it implies that biological systems (*i.e.,* the bigger networks) are composed of many (smaller) networks that are integrated at and communicating on multiple scales. Under this "network of networks" framework, Systems Biology seeks to formulate hypotheses for biological functions, as well as to provide spatial and temporal insights into biological dynamics, by analyzing the component networks *across scales* and by integrating their behavior *at different levels*[100].

Indeed, by focusing on the study of single biomolecules and of the interactions between specific pairs of proteins, the traditional reductionistic approach to molecular biology operates on a single scale, thus imparting a limited understanding of the system[100]. By contrast, instead of merely identifying individual genes, proteins and cells, and studying their specific functions, Systems Biology investigates the behavior and relationships of all of the elements in a particular biological system while it is functioning[98].

Consequently, when compared to the traditional reductionistic approach, a *systems* approach towards biology enables the modeling of *global biological mechanisms* such as the circadian clock, for example, that are a result of complex interactions between various agents acting on heterogeneous time- and concentration- scales: genes, mRNAs, protein complexes, metabolites, tissues, organs, signalling networks, *etc.*...

The philosophy of Systems Biology thus lays in the *multiscale* exploitation of the huge amount of data yielded by the traditional approach, in order to study how the function of a biological system arises from dynamic interactions between its parts (*i.e.,* how low-level biological data translates into functioning cells, tissues and organisms). In this way, it aims at creating a consistent system of knowledge - and an understanding of biology at a system level- that is grounded in the molecular level [108]. By integrating models at different scales and allowing flow of information between them, multiscale models describe a system in its entirety, and as such, are intrinsic to the principles of Systems Biology[100].

The central tasks of Systems Biology can be divided into: [98] *(a)* comprehensively gathering information from each of the distinct levels of individual biological systems and *(b)* integrating the data to generate predictive mathematical models of the system.

According to [108], task *(b)* can be further partitioned into the following objectives:

1. **System structure identification:** identifying the system's components, as well as the interactions between them (*i.e.,* the system topology) and the interaction parameters;

2. **System behavior analysis:** once the model's structure is decided upon, its behavior needs to be understood, either by identifying the temporal evolution of the components' quantities (through *simulation*), or by using various analysis techniques to determine behaviours (*e.g.*, analyze the system's sensitivity against external perturbations, and how quickly it returns to its normal state after the stimuli [108]);

3. **System control:** aims at applying the insights obtained by structure identification and system behavior analysis towards establishing a method to control the state of biological systems, in order to address questions with immediate therapeutical benefits, such as: how to transform malfunctioning cells into functional ones, or how to drive cancer cells to apoptosis[2], or how to control the differentiation status of a specific cell into a stem cell, and then control it to differentiate into the desired cell type;

4. **System design/Synthetic Biology:** ultimately design biological systems *ab initio*, that exhibit a certain desired functionality.

[2]programmed cellular death

As such, Systems Biology is a hypothesis-driven research field, which is characterized by a synergistic combination of experimentation, theory and computation. The idealized process of Systems Biology research, as presented in [109] and illustrated in Fig.1, consists in a cycle that begins by selecting the contradictory issues of biological significance, and continues with the manual or automatic creation of a model representing the phenomenon. This model represents a computable set of assumptions and hypotheses that are to be tested experimentally, after having previously been revelead as adequate through simulation. The *in silico* experiments play a key role in the research cycle: when provided with inadequate models, simulation reveals inconsistencies with established experimental facts, thus informing the researcher that the model needs to be either modified or rejected. Models that pass this test are then subjected to thorough system analysis, where a number of predictions can be made. Among these, a set of predictions that are able to distinguish a correct model among competing models are selected for "wet" experiments - succesful experiments eliminate inadequate models. Models that survive this cycle are deemed to be consistent with existing experimental evidence, and can therefore be used in Steps 3 and 4 mentioned above.

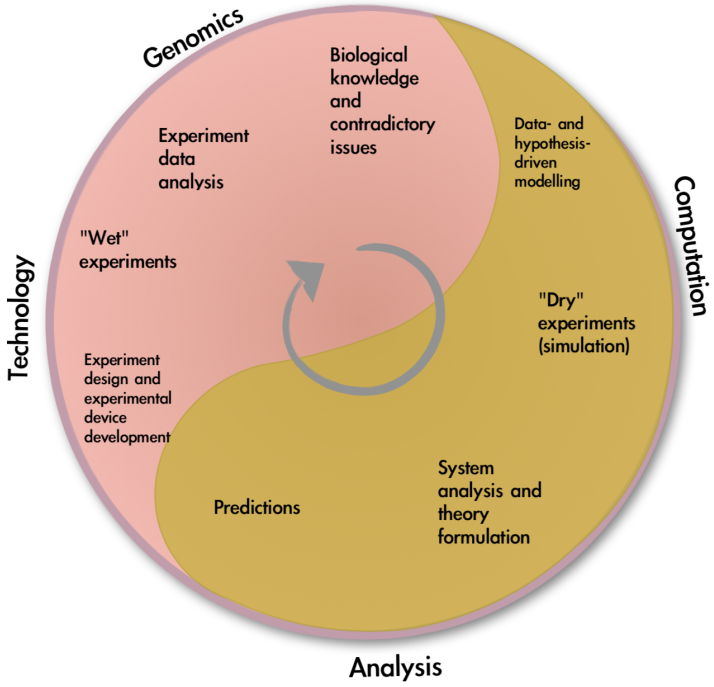


Figure 1: The idealized Systems Biology research cycle, according to [109]

In this manuscript, we address theoretical and computational issues in cellular modelling. More specifically, our contributions are in the areas of behavior analysis, model reduction, and the computational modeling of specific biological behaviours of interest.

Our work consequently relates to the *computation* and *analysis* branches of the Systems Biology research cycle of Figure 1, which aim at formulating data- and hypothesis-driven mod-

els of and theories for biological mechanisms, as well as developing efficient algorithms, data structures, visualization, data-analytical and communication tools for predictive computer modeling - a significant task of systems and mathematical biology. It involves the use of computer simulations of biological systems to both analyze and visualize the complex connections of the underlying cellular processes, and its ultimate goal is to create accurate real-time models of a system's response to environmental and internal stimuli: *e.g.,* a model of a cancer cell in order to find weaknesses in its signalling pathways.

Notable previous efforts in Computational Systems Biology include helping sequence the human genome, and creating accurate models of the human brain [95].

Challenges

All in all, with Systems Biology, a paradigm shift occurs in biology, away from a *descriptive* science, and toward a *predictive* one. Experimental observations cover a wide range of biological processes, that span from simple isolated enzymatic reactions, to complex systems such as patterns of gene expression and regulation. As even simple dynamic systems can exhibit a range of complex behavior, which cannot be intuitively predicted from experiences, the systems approach requires quantitative mathematical and statistical modelling of biological system dynamics [101]. Thus, modelling becomes a part of biological reasoning, however it turns out to be a particularly challenging task.

The first challenge arising in model construction is that dynamical models need to address the specificities of biological systems, which are as follows [36]:

- **Biological systems integrate multiple scales**, both with respect to time, and with respect to their constituent components. For example, chemical processes governing network dynamics span over many well separated timescales: while protein complex formation occurs on the seconds scale, post-translational protein modification takes minutes, and changing gene expression can take hours, or even days. As for the system components, multiscaleness applies both to the abundance of various species in biochemical networks (*e.g.,* the DNA molecule has one to a few copies, while mRNA copy numbers can vary from a few to tens of thousands), and to the scale of the components themselves, which can range from molecule and cell, to tissue, organ, individual, organism, and ecosystem.

- **Biological systems are governed by a mix of *deterministic* and *probabilistic* behaviors**: their behavior at the finest timescale is inherently stochastic, thus requiring probabilistic models, while integrating across scales typically yields deterministic behaviors.

- **The complexity of biological systems warrants phenomenological models.** Biological systems have evolved under the dual mechanism of mutation and selection. Consequently, they have selected for robustness, which is often translated by the presence of redundant features. For example, consider the existence of alternative competing metabolic pathways that are related to the same function; indeed, such a redundant feature can be interpreted as a backup strategy that renders the system robust to fluctuations in the

environment. The presence of redundancy in biological systems sheds doubt on the existence of simple laws governing the behavior of complex systems, which in turn partially explains the apparent duality of biological modeling: one the one hand, models are derived from first principles, but on the other hand, the development of phenomenological models is based parameter correlation investigation, thus calling for machine learning and inferential modeling methods.

- **The variability of biological systems calls for statistical assessments:** thus, there is a need for generic models that accommodate individual-specific variations, as well as for a statistical assessment of the parameters used to single out specific properties.

Dynamical model construction also raises several central problems from a knowledge perspective:

- finding an appropriate level of abstraction for a given analytic problem;
- finding a common basis to relate knowledge gained using different experimental techniques on the same system, or conversely to relate knowledge gained from the same experiment on different model systems;
- incorporating knowledge incrementally as new data is analyzed.

Finally, any mathematical or algorithmic development for biological sciences requires acknowledging the existence of a number of specificities of the latter, that are not common practice in mathematics and computer science. Model and technique design should thus require reconciling these somewhat different perspectives. These specificities are [36]:

- **an inherently system-centric development:** while mathematics and computer science aim at exhibiting general properties and algorithms (which can be instantiated in a number of settings), biology is often a system-centric activity that focuses on a specific cell, organ, or pathology.
- **an interest towards ill-posed problems**: while mathematics and computer science have traditionally been concerned with well-posed problems, modeling in biology is as concerned with designing models that serve to identify ill-posed problems, as it is with solving well-posed ones.
- **a need for validation:** every biological model should eventually be evaluated against experimental results, in order to be confirmed or invalidated.
- **models for complex biological systems are often multidisciplinary:**
  - biological knowledge/data serve as the starting point of the model, as well as providing its semantics, by embedding it in a biological context
  - mathematics enables the compromise between biological accuracy and conceptual simplicity, by allowing information to be abstracted at different levels.

  - physics and chemistry are used to endow the abstract model with mechanical or electrical properties
  - computer science allows for automatizing tasks, performing analyses, and, most-importantly, running model simulations, that serve as numerical experiments from which properties of the system can be inferred

- **the dual contribution of mathematics and computer science in biology**: firstly, the robustness and efficiency of existing methodological development can be improved by developments in the two fields (*e.g.,* by mastering the numerics of floating-point calculations, improving the convergence properties of algorithms for optimization purposes, and designing algorithms with enhanced asymptotic properties, that *scale* better); secondly, concepts and algorithms from mathematics and computer science are used to lay the groundwork for more advanced, more accurate models (*e.g.,* stochastic modeling, inverse problem solving, machine learning, statistical inference).

As mentioned above, the first question that arises when trying to construct models of biochemical networks deals with the level of abstraction to be employed. Dynamical models of biochemical networks, like all models, are abstractions of reality, which focus on certain aspects of the subject, and abstract away other aspects. A wide variety of modelling techniques have been proposed, each of different complexity and abstraction, ranging from models representing regulatory relations between genes as simplified discrete circuits (*e.g.* gene regulatory networks), through models containing a detailed description of the low-level mechanistic interactions between molecules (*e.g.* rule-based models), to models of spatio-temporal dynamics of processes considered at an individual particle level (*e.g.* stochastic models in which the diffusion, interaction with surfaces and participation in chemical reactions of each point-like particle is tracked individually [6],[17]). In the absence of a universal modelling framework, each approach abstracts the biological processes being considered in a specific way, thus bringing along its own set of assumptions as well as its own niche of validity. The overall challenge is related with the understanding of cellular processes and their modeling within the adequate level of abstraction.

After having settled on one of the available modelling frameworks, two issues await the modeler. The first challenge deals with *identifying the structure of the system*, a necessary step towards the understanding of the biological system. The difficulty is that a biochemical network cannot be inferred automatically from experimental data based on universal rules of principles, because biological systems evolve through stochastic processes and are not necessarily optimal [108]. What's more, one must identify the true network, among multiple candidates that exhibit similar behavior to the desired one.

Once the structure of the system has been fixed, the modeler needs to identify the set of corresponding parameters (*i.e.*, binding constant, transcription rate, translation rate, chemical reaction rate, degradation rate, diffusion rate, *etc...*), as all computational results have to be matched and tested against actual experimental data. What's more, simulation is needed in order to carry out a quantitative analysis of the system's response and behavioral profile. Except for certain well-studied cases, these constants are not readily available. Ideally, comprehensive measurements of major parameters should be carried out through wet-lab experiments, however, rate constants often vary drastically *in vivo*[108]. Consequently, various *in silico* parameter

optimization techniques can be used to find a set of parameters leading to simulation results consistent with experimental data (*e.g.*, brute force exhaustive search, genetic algorithms, simulated annealing), however most of them are computationally expensive.

Computational cost turns out to be a bottleneck of Systems Biology: returning to the challenge of choosing the modeling framework with the most adequate level of abstraction, one might think that an increased level of details is desirable for model comprehensiveness. This, however, is not always true, as it can lead to a model whose complexity becomes prohibitive. For models of biological systems, the term "complexity" refers not only to non-linearity and emergent behavior, but also to *(i)* their inherent *multiscaleness* (biochemical processes governing network dynamics typically span over many well separated timescales, and abundance of various chemical species can span over many well separated concentration scales) and to *(ii)* to the sheer size of the model (as measured by the number of components and the pattern of interactions among them). These complexity-inducing features make constructing an analytical perspective w.r.t. biological models a challenging task. Moreover, as computational modeling continues to make significant progress, the issue of scalability of techniques to models of realistic size remains a major challenge, as the growth in complexity is often exponential and arises independently of the representation [165].

For example, in models of biochemical networks, the number of possible chemical species is often subject to combinatorial explosion, due to the large number of species that may arise as a result of protein bindings and post-translational modifications [94]. As a consequence, mechanistic models of signaling pathways easily become very combinatorial. What's more, even if compact ways of describing models prone to combinatorial explosion exist (*i.e.*, rule-based models), the curse of dimensionality once again rises when trying to compute the system behavior. A strategy to cope with such complexity is *model reduction*, in which certain properties of biochemical models are exploited in order to obtain simpler versions of the original complex model; these simpler models should preserve the important behavioral aspects of the initial system. The major challenge with respect to model reduction techniques lies in providing explicit bounds on the accumulated reduction errors, or in other words, providing guarantees as to how the solution of the reduced model relates to that of the original model, while avoiding to solve the original model (whose size is often prohibitive).

Finally, the need for modeling biological *systems*, instead of biological mechanisms in isolation, has begotten the development of "whole-cell" computational models, which have been described both as "the ultimate goal" of Systems Biology, and as "a grand challenge for the 21st century" ([35],[177]). These models aim to predict cellular phenotypes from genotype, by an exhaustive representation of the cellular machinery: all of the chemical reactions in a cell, all of the physical processes that influence their rates, the entire genome, the structure and concentration of each molecular species, and the extracellular environment [105]. Given their potential of unifying the understanding of cell biology and of enabling *in silico* experiments to be performed with complete control, scope and detail ([143],[35],[177]), they are poised to have a dramatic impact on Systems Biology, bioengineering and medicine [105]. The reasons for this lay in their potential to guide experiments in molecular biology, as well as enable computer-aided design and simulation in synthetic biology, and inform personalized treatment in medicine [122]. To date, the most complete computational cell model is a mechanistic whole-cell model for the bac-

terium *Mycoplasma genitalium* [104], in which diverse mathematical techniques from multiple fields are combined, in order to enable mechanistic modeling at multiple levels of abstraction [3], in an integrated simulation. [35]

Despite having being hailed as a future transformative force of Systems Biology, several challenges (still) prevent the wide-scale conception of whole-cell models. These challenges are related, among others, to incomplete and disparate biological knowledge, to the integration of multiple scales from genotype to phenotype, to the exhaustive description of each molecule and each interaction, and to the scalability of computational tools and methods. In [122], the authors of the *Mycoplasma genitalium* model identified 7 areas containing the main challenges to building whole-cell models: experimental interrogation, data curation, model building and integration, accelerated computation, analysis and visualization, model validation, and collaboration and community development. However, Systems Biologists are leveraging recent progress in measurement technology, bioinformatics, data sharing, rulebased modeling, and multi-algorithmic simulation in order to construct whole-cell models, and it is anticipated that ongoing efforts towards developing whole-cell modeling tools "will enable dramatically more comprehensive and more accurate models, including models of human cells"[143]. Beyond the experimental and computational progress that is making whole-cell modeling possible, there is nonetheless a need for several technological advances, that would help accelerate the framework, in the areas of: metabolome-wide and proteome-wide measurement technologies, kinetic parameter measurement technologies, data aggregation tools, tools for collaboratively building large models directly from experimental data and for identifying gaps and inconsistencies in models, rule-based modelling languages that would support all of the biological processes that must be represented, scalable multi-algorithmic simulation, calibration and verification tools, and simulation analysis tools [143].

With all the challenges raised by whole-cell models that aim at exhaustively describing the entirety of the cell machinery, one can turn to a simpler class of models, dubbed "self-replicator" cellular models. Instead of accounting for all annotated gene functions of a cell, the "self-replicator" paradigm depicts minimal mechanistic models (*i.e.,* that contain just a few classes of proteins, grouped according to their function) of the molecular regulatory mechanisms inside the cell, and that are able to reproduce well-known experimental microbial growth laws. The strength of such models specifically lays in their ability to reproduce experimentally observed behaviors, with a minimalistic, coarse-grained modelling approach. The most well-known such model is presented in [130], where the authors construct a minimal model of a self-replicating bacterial cell. The model consists of only 4 enzymes and a membrane, and is optimized for growth rate, under the assumption of competition for a limited amount of ribosomes. The results of such optimization display regulation of properties (like the ribosomal content and the surface/volume ratio) similar to those observed in real cells.

In this manuscript, we will tackle such mechanistic self-replicator models (more specifically, the one presented in [182]), instead of whole-cell models in the classical sense, in order to address the issue of modelling intracellular resource storage strategies.

We note that by modeling an exhaustive range of cellular processes - albeit in a simplified

[3] thousands of heterogeneous experimental parameters are simultaneously included in the model, which captures a wide range of cellular behaviors

fashion-, from extracellular nutrient import and metabolization, to transcription, translation and degradation, such mechanistic models are indeed constructed in the spirit of Systems Biology.

Outline and Contributions of the Thesis

The works carried out during this thesis were precisely motivated by the challenges mentioned previously. This manuscript is composed of four independent - but complementary - projects, each one tackling one of the existing challenges in Systems and Computational Biology. The four works can be grouped in two orthogonal groups: the first group tackles model reduction heuristics and model reduction error estimation techniques that exploit the inherent multiscaleness of biological systems, while in the second one, we study the formal modelling of relevant biological behaviours (*i.e.*, intracellular resource storage and growth) in the deterministic modelling framework (*i.e.*, through ordinary differential equations). We note that our contributions span both modelling frameworks: deterministic and stochastic. Furthermore, our case studies range from classical, well-understood and exhaustively-studied examples, such as the *E.coli*'s λ-phage switch, to more recent models that fall under Systems' Biology goal of working toward a system-level understanding of biological systems.As such, in Chapter 7, our case study builds upon a recent mechanistic cellular growth model that respects the universal trade-offs that arise in cells due to resource limitations, and which quantitatively recovers the typical behavior of both an individual growing cell, and of a population of cells.

The structure of the manuscript is the following.

In **Part I**, **Chapter 1**, we introduce the notion of biochemical reaction network (BRN), and present the state of the art with respect to its mathematical and computational modelling. As commonly done for programming languages, we differentiate between *syntax* and *semantics*. The former, presented in **Chapter 2**, is represented by a network structure that models the biological system as set of species undergoing a series of chemical reactions. As for the semantics, we present the two major approaches to modeling the *dynamics* of a reaction network - the *deterministic* and the *stochastic* dynamics - , as well as their respective associated simulation and analysis tools, in **Chapter 3**. Also in Chapter 3, we show how the deterministic and stochastic models are related via a scaling limit. In **Chapter 4**, we present two computational models of Biochemical Reaction Networks, that will be addressed in this thesis: Petri Nets, in Section 4.1, and the the rule-based modeling language Kappa, in Section 4.2.

Part II of the manuscript is composed of two projects dealing with model reduction techniques and reduction error estimate heuristics.

The first contribution, presented in **Chapter 5**, deals with the *stochastic* modelling of genetic circuits, in which the evolution dynamics of a BRN is modeled as a Continuous Time Markov Chain (CTMC). Herein, we argue for the use of rule-based models in prototyping genetic circuits, and tackle the issue of multiscaleness-based model reduction in the context of Kappa, one of the existing rule-based modeling languages. As it will be explained in Part I, rule-based models allow, on the one hand, to circumvent the combinatorial explosion in the size of the models, by using a set of rules to indirectly specify a mathematical model. On the

other hand, rule-based modeling languages are designed to capture the mechanistic details of the protein-centric biological interactions, thus leading to transparent, modulable, extensible and easily understandable models of biological networks. This latter aspect will be the focus of our study; the complexity-reducing properties of using rules instead of reactions will *a priori* not be exploited in this manuscript.

When designing genetic circuits, the typical primitives used in major existing modelling formalisms are gene interaction graphs. This framework operates on a high level of abstraction, by modelling the circuit as a graph whose nodes denote genes and whose edges denote activation or inhibition relations between genes. Gene interaction graphs contain no information with respect to lower-level mechanistic details as to how such regulation relations are implemented. However, when designing experiments, it is important to be precise about this kind of details.

Fortunately, such protein-protein mechanistic interaction details can be modeled using Kappa - a rule-based language for modeling systems of interacting agents, which allows to unambiguously specify mechanistic details such as DNA binding sites, dimerisation of transcription factors, or co-operative interactions. Nonetheless, such a detailed description comes with complexity, as well as computationally costly executions. Consequently, we propose a general reduction method of rule-based models of genetic circuits, in which each rule is a reaction, based on eliminating intermediate species and adjusting the rate constants accordingly.

Our method is an adaptation of an existing algorithm, which was designed for reducing reaction-based programs[112]; our version of the algorithm scans the rule-based Kappa model in search for those interaction patterns known to be amenable to equilibrium approximations (e.g. Michaelis-Menten scheme, as well as a number of other stoichiometry-simplifying techniques). Additional checks are then performed in order to verify if the reduction is meaningful in the context of the full model. The reduced model is efficiently obtained by static inspection over the rule-set. We test our tool on a detailed rule-based model of a λ-phage switch, which lists 92 rules and 13 agents. The reduced model has 11 rules and 5 agents, and provides a dramatic reduction in simulation time of several orders of magnitude.

The Michaelis-Menten model of enzyme kinetics, which constitutes the basis of our reduction algorithm, is arguably the best known example of an approximation that exploits the multiscaleness property of biochemical networks, with respect to both time-scales and species abundance. It is commonly used to represent enzyme-catalysed reactions in biochemical models, and has been thoroughly studied in the context of traditional differential equation models [164]. Classically, ordinary differential equations represent an adequate approach to modelling biochemical systems in which species appear in abundance. However, the presence of small-concentration species in biochemical systems encourages the conversion of the Michaelis-Menten mechanism to a stochastic representation. It has been recently shown that this approximation is also applicable in discrete stochastic models, and that furthermore, its validity conditions are the same as in the deterministic case [164]. More specifically, in the deterministic case, its derivation is based on the *equilibrium approximation*, which is valid if the substrate species reaches equilibrium on a much faster time-scale than product formation, or alternatively on the *quasi steady-state approximation*, which requires that enzyme concentration be much less than the substrate concentration.

Our method is an illustration of the fact that in general, the multi-scaleness of biochemical

reaction networks represents a feature that can be exploited for model reduction purposes: it allows to approximate the complete mechanistic description with simpler rate expressions, retaining the essential features of the full problem on the time scale or in the concentration range of interest.

Nonetheless, providing guarantees as to how the solution of the reduced model relates to the original one remains a challenge. To the best of our knowledge, there exist no precise methods to quantify the error induced by time-scale separation approximations for biochemical reaction networks, while avoiding to solve the original model. The bottleneck lays in the complexity of the original system, whose behavior can be computationally costly to analyze - often times, its size is prohibitive enough to allow even running a single simulation trace.

The correctness of our approach relies on the fact that the approximate model is equal to the original one, in the artificial limit where certain reactions happen at a sufficiently larger time-scale than others, and they are seemingly equilibrated shortly upon the reactions initiation. However, not unlike other scale-separation reduction methods, our method relies on a solid physical justification, yet the numerical approximations lack both explicit bounds on the accumulated errors, and proof of soundness.

This is the reason for which, in **Chapter 6**, we propose an approximation method for deterministic models of biochemical networks, in which reduction guarantees represent the major requirement. Our method combines abstraction and numerical approximation, and aims at providing a better evalutation of model reduction methods that are based on time- and concentration- scale separation. The reduction guarantees of our method are a consequence of a carefully designed symbolic propagation of dominance constraints: given an ODE model of a BRN that exhibits time- and concentration- scale separation, we abstract the solution of the original system by a "box" that over-approximates the state of the original system and provides lower and upper bounds for the value of each variable of the system in its current state. The simpler equations (which we call *tropicalized*) that define the hyperfaces of the box are obtained by combining the dominance concept borrowed from tropical analysis [119] with symbolic bounds propagation. Mass invariants of the initial system of ODEs are used to refine the computed bounds, thus improving the accuracy of the method. The resulting (simplified) system provides *a posteriori* time-dependent lower and upper bounds for the concentrations of the initial model's species, and thus bounds on numerical errors stemming from tropicalization. This means that no information on the original system's trajectory is needed - the most important advantage of our approach. By contrast, the main difficulty of applying the classical QSS and QE reductions to biochemical models is that QE reactions and QSS species need to be specified *a priori*, which implies that some knowledge about the initial system's behavior is necessary. This, in turn, means that significantly high-dimensional, non-linear systems cannot benefit from these reductions, as their analysis can be prohibitive in practice.

Depending on the chosen granularity of mass-invariant-derived bounds, we show that our method can either be used to reduce models of biochemical networks, or to quantify the approximation error of tropicalization reduction methods that do not involve guarantees. As such, the guarantees of our method are obtained by formalizing the soundness relation between the original system of equations and the abstract system of ordinary differential equations operating on the coordinates of the hyper-faces of the box. The solution of a sound abstraction of an

original system of differential equations, starting from a box that contains the initial state of the original system, defines a sound abstraction of the solution(s) of the original system. We apply our method to several case studies (a simple DNA model constructed as an extension of the classical Michaelis-Menten reaction scheme, and to the minimal cell cycle proposed by Tyson [178], and finish by comparing it to existing interval numerical methods for enclosing the solution of an initial value problem (IVP) between rigorous bounds.

The works presented in **Part III** deal with issues regarding the mathematical modeling of biological behaviors of particular interest, in the deterministic modelling framework. Namely, we address the modelling of intracellular resource storage strategies in self-replicating mechanistic models, as well as cellular growth as an emergent property of Petri Net model semantics of BRNs.

As such, in **Chapter 7** we address the issue of *storage*. Cells grow by fueling internal processes with resources taken from the outside. Depending on the responsiveness of these biosynthetic processes with respect to the availability of intracellular resources, cells can build up different levels of resource storage. In this scenario, the questions we investigate are: how does storing resources impact cell growth? Namely, how much of these resources should a cell pile up internally, given the opportunity? And how does storage depend on resource availability and on other species competing for the same pool of resources?

To answer these questions, we introduce a new reparametrisation technique of ODE models, intended to model intracellular resource storage strategies. Our technique consists in defining a generic scaling transformation of BRNs that allows one to tune the concentration of certain chemical species, while preserving the network's behaviour at steady state. Consequently, it enables one to symbolically navigate natural lines of iso-cost in parameter space, provided cost functions only depend on steady-state constraints (and only on a subset of the model variables). In our specific case, this means that we can guarantee by construction that various storage strategies preserve approximatively goodness-of-fit to the original growth data, and therefore correctly match growth conditions to sectorial resource allocations.

We acknowledge that storage has a concrete maintenance cost, because idle resources are diluted by growth, but also that higher storage levels improve the dynamics of reallocation of resources among the various sectors of production (transporters, metabolism, translation) when sugar levels change sharply in the environment. This fundamental trade-off is best investigated using a mechanistic model of cellular growth, where costs are emergent and reflect architectural traits of the growth machinery. Consequently, we apply our method on such a recent "self-replicator" mathematical model of the coarse-grained mechanisms that drive cellular growth (*i.e.*, the *Weisse* model [182]), in order to investigate the effects of cellular resource storage on growth. We carry out our analysis not only for a single-cell model, but also in a competitive context.

At the single cell level, we start by comparing storage strategies against different patterns of environmental changes. We investigate the impact of scaling (that we identify as storage) on cellular growth during shifts of the sugar yield, and are able to make a number of observations: *(i)* storage capacity can be modulated over several orders of magnitude without significantly affecting growth rate, *(ii)* in constant environments, excessive storage of the protein precursor is detrimental to growth rate, *(iii)* the cost of storage, in terms of reduced growth, is condition-

dependent, and higher in rich growth conditions, *(iv)* storage results in smoother physiological transitions during environmental up-shifts and increases biomass during such transitions, as resource allocation is dependent on protein precursor concentration, and *(v)* evolutionary benefits of storage increase with the frequency and magnitude of environmental fluctuations.

Our results thus suggest that there is a cost associated with high levels of storage, which results from the loss of stored resources through dilution. On the other hand, high levels of storage can benefit cells in variable environments, by increasing biomass production during transitions from one medium to another. A potential explanation for this behavior is that a suitable amount of storage can decrease the cost of resource reallocation caused by changes in growth conditions, as reflected by the Weisse model. Our results thus suggest that cells may face trade-offs in their maintenance of resource storage based on the frequency of environmental change.

We continue by adopting an ecological perspective, in which we compare storage strategies in competitive situations (*i.e.,* in which species contend for the same resources). To do so, we test populations of low- and high-storage strategies against each other, in a variety of environments parametrized by the frequency of two superimposed probabilistic trains of high and low pulses of sugar. Our experiments demonstrate the existence of a convex boundary separating a domain of "bursty" regimes, characterized by high and infrequent sugar pulses, from the rest. In this domain, the low-storage strategy, which is faster growing, wins. A surprising result is that outside of this domain, the fast growers are driven to extinction by the high-storage, slow growers; lasting co-existence of the two storage strategies can only be observed on the boundary.

All in all, with our model in place, we are able to observe a rich interplay of storage levels, growth rates, growth yield (*i.e.*, the amount of biomass produced per unit of growth medium), and resource variability. Our results indicate that the specificities of environmental changes play a decisive role in deciding which storage strategy is deemed the most beneficial (with respect to accumulating biomass over time). This is even more so the case when species content for the same resource: the combined effect of storage strategies and competition lead to extracting less biomass out of the same amount of resources.

The last matter we address in this thesis deals with the modeling of cellular growth. In **Chapter 8**, we work towards a characterization of cellular growth as an emergent property of a novel Petri net execution semantics. Consequently, we aim to substitute to a "growth" biochemical reaction network (BRN) (i.e., for which an exponential stationary phase exists) a piecewise-synchronous approximation of the deterministic dynamics. To achieve this, we propose to model a BRN using a resource-allocation-centered Petri Net, with parallel maximal-step execution semantics. We argue that this semantics is better-suited for modeling biochemical reaction networks, when compared to the classical interleaving semantics, as it takes into account the inherently concurrent nature of biological processes. In the case of unimolecular chemical reactions, we prove the correctness of our method and show that it can be used either as an approximation of the dynamics, or as a method of constraining the reaction rate constants (an alternative to flux balance analysis, using an emergent formally defined notion of "growth rate" as the objective function), or a technique of refuting models.

Part I

Biochemical Reaction Networks: A Review

Chapter 1
Context and Motivation

1.1 Features of Dynamical Models in Systems Biology

We have previously detailed the crucial role that quantitative mathematical (or *dynamical*) models plays in modern biological reasoning. Herein, we elaborate on their required components.

The primary components of a dynamical mathematical model of a biological system correspond to the molecular species present in the system.The abundance of each species is assigned to a *state variable* of the model, the collection of which represents the *system state*. The system state provides a complete description of the system's condition at any given time, via its time course, as given by the model's *dynamic behavior*.

Besides state variables, models of biological systems also include *parameters*, which characterize environmental effects and interactions among system components, and whose values are fixed - meaning that the distinction between model parameters and state variables is clear-cut. Example of common parameters include association constants, maximal expression rates, and degradation rates. As a change in the value of a model parameter corresponds to a change in the environmental conditions or in the system itself, model parameters are typically held constant during a simulation. Varying parameter values between rounds of simulation allows one to explore system behavior under perturbations of the experimental conditions, or in altered environments.

As the main feature of a *dynamical model* of a biological system is its ability to describe the temporal evolution of the system components' quantities, building a model then involves two important choices: *(i)* how to represent the model structure (*i.e.,* determine its species, parameters, and molecular interactions), which is akin to defining its *syntax*, and *(ii)* how to 'interpret', or execute, the model, which is akin to defining its *semantics*. These two steps are indicative of a distinction that is commonly made when dealing with systems modeling: the *qualitative* approach, respectively the *quantitative* approach. While the latter essentially relies on mathematical equations describing the performance of the system for a large set of input functions and initial states, the former requires no such formal mathematical formulation, instead relying on visual representations (*e.g.,* diagrams) of the relationships between the system components, that is, it relies on the *structural* model.

For a large class of biological systems, what one usually seeks to model are the interactions between individual molecules which are only distinguishable by the class of species they belong to. Consequently, *population models* prove to be a well-suited, and popular, mathematical modelling framework for biological phenomena. Any population model can be described as a set of *reactions* operating on a set of *biochemical species*: this is what we will refer to as *Biochemical Reaction Networks*; they will represent the modeling framework of choice throughout this thesis. Below, we motivate our choice.

1.2 Network models of biological systems

While using mathematical models in the *natural* sciences (particularly in physics) is not a new idea[1], the novelty lays in the use of such models in *life* sciences, most notably in biology and biomedicine. As such, *quantitative mathematical* models have lain outside mainstream research approaches during the last decades of the purposely reductionist qualitative era of molecular biology, which instead heavily employed *qualitative* models. Indeed, biologists have long used these type of models as abstractions of reality, be it in the familiar form of the ball-and-stick model of chemical structure, or more generally in the form of diagrams that illustrate a set of components and their interactions, and which play a central role in representing our knowledge of cellular processes [99].

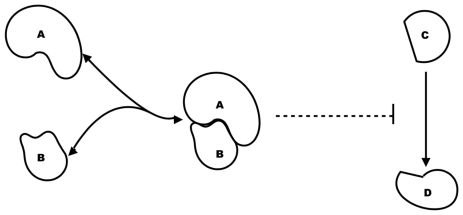


Figure 1.1: An interaction diagram, inspired from [99]. Species A and B bind reversibly, forming a complex that inhibits the rate at which species C is transformed into species D. The graphical conventions used are that the blunt arrows denote inhibition, normal arrows denote activation, while dashed lines denote regulatory interactions, *i.e.* the species is not consumed by the reaction.

Such interaction maps, also called *networks*, formalize the complex interactions between heterogeneous entities (within and between cells), which characterize biological systems. For example, Figure 1.1 shows molecular species A and B binding reversibly, to form a complex that inhibits the rate at which species C is transformed into species D.

Generally speaking, such network-based modeling approaches help structure and formalize existing knowledge, as well as predict system behaviour, and have been widely used for describing cellular regulation and metabolism [16]. Consequently, it makes sense that a *quantitative*

[1] Instead, it can be traced back to ancient Greek scholars, if not even further in the past

modeling framework be developed using this widely-employed *qualitative* description of biological systems: enter *Biochemical Reaction Networks* (BRNs). The central argument that justifies adding quantitative/mathematical information to simple interaction maps, in order to obtain *quantitative* BRN models, is that simple interaction diagrams leave ambiguities with respect to the system's behaviour: we know what components of the system interact with each other, but not necessarily *how*. By employing a *mathematical description* of the system, this uncertainty can be eliminated, at the cost of demanding a *quantitative* characterization of every interaction depicted in the diagram [99].

For example, in order to quantify the interaction between molecular species A and B, in Figure 1.1, a numerical description of the process must be provided, under the form of the binding and unbinding reaction rate constants. For cellular processes of which only a qualitative understanding of the underlying molecular interactions is available, such a quantitative description is not possible. However, for an important number of well-studied mechanisms, sufficient data have been collected as to allow a quantitative characterization [99], that, when coupled with the interaction diagram, can be used to formulate a *mathematical model of the network's dynamics*, which typically involves the physical and chemical laws governing the mechanisms that drive the observed behaviour (such models are dubbed *mechanistic*). Whenever the quantitative information regarding molecular interactions is available, the result is a collection of *biochemical reactions* involving the system's species. A *species* is a system entity that is *quantifiable*, and whose quantity is susceptible of evolving over time. A *reaction* is an elementary action of the system that consists of consuming (respectively producing) a finite and integer quantity of a finite subset of species, called *reactants*(respectively *products*).

For example, the reversible binding reaction between species A and B of the interaction diagram in Figure 1.1 writes as:

$$A + B \underset{\kappa_{-1}}{\overset{\kappa_1}{\rightleftharpoons}} A.B, \tag{1.1}$$

where $A.B$ denotes the resulting complex.

We note that, at this level of modeling, the notation $A.B$ for the complex species is simply a name, and should not be considered as indicative of the reaction's mechanistic details (*i.e.*, A binds B). In this sense, one could choose any other name for the complex species: *e.g.*, reaction 1.1 could be equivalently written as $A + B \underset{\kappa_{-1}}{\overset{\kappa_1}{\rightleftharpoons}} C$.

The model parameters are the binding, respectively unbinding, reaction constants κ_1 and κ_{-1}. As we will see in this chapter, both the interpretation of the reaction constants and the representation of the system state will depend on the chemical kinetics assumed to govern the dynamics of the system.

Example 1.2.1
A more biologically relevant example is the Michaelis-Menten mechanism, which consists of an enzyme, denoted E, that reversibly binds a substrate, S, to form a complex, $E:S$. The complex then releases a product P, while preserving the enzyme. The associated reaction network writes as:

$$\mathrm{E} + \mathrm{S} \underset{\kappa_{-1}}{\overset{\kappa_1}{\rightleftharpoons}} \mathrm{E:S} \xrightarrow{\kappa_2} \mathrm{E} + \mathrm{P} \tag{1.2}$$

Biochemical Reaction Networks will be the modeling framework of choice throughout this thesis. Consequently, in the next sections of this chapter we formally define both their syntax, as well as the two major applicable semantics: classical (deterministic) chemical kinetics, respectively stochastic chemical kinetics.

Before moving forward with the definition of BRNs, we note that a disadvantage of network models is that as the number of components and interactions in a biological system grows, it becomes increasingly difficult to maintain an intuitive understanding of the overall behaviour [99]. Thus, different levels of information abstraction can be employed when constructing network models, according to the size of the system, the available data, and the research question under consideration. In their simplest form, the interaction maps only depict the possibility of interaction between genes or proteins (Figure 1.2, top) - and can therefore cover larger systems -, while at the other end of the scale one finds detailed models based on mathematical descriptions (Figure 1.2, bottom), which provide significantly more detailed insight into the dynamics, but are usually only used to describe small, well-studied subsystems. Figure 1.2 shows a hierarchy of different, widely-employed, network models, each one abstracting information on a different level, and thus requiring different amounts of detail.

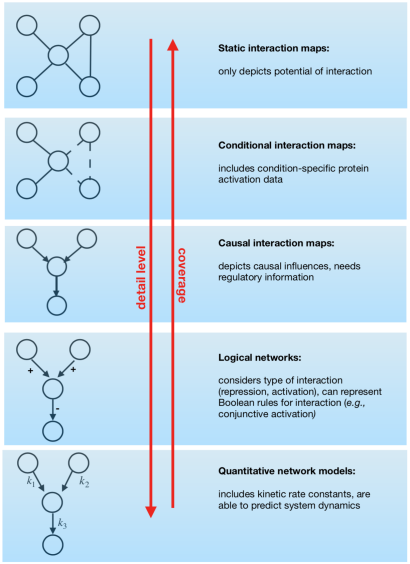


Figure 1.2: **A hierarchy of network models for biological systems [16].** Existing network models of biological systems abstract information on different levels, depending on the available data and the biological questions under consideration. *Top-down*: depicted model types are ranked in increasing detail level and data demand, and decreasing coverage power. Less detailed models cover larger systems (in number of genes/biomolecules), but do not allow for true simulation of network dynamics - simulation is possible only for the most detailed category of models, dubbed "quantitative models", that are labeled with kinetic rate constants. Intermediary models, *i.e.* causal and logical networks, allow at most for simulation of the event sequence or gene activation order, but quantifying the speed of reactions is impossible under these simplified frameworks.

Chapter 2
Biochemical Reaction Networks: Syntax

Models of cellular phenomena often take the form of interaction diagrams, as in Fig.1.1. For biochemical and genetic networks, interaction diagrams depict the *molecular species* in the system - which could be ions, small molecules, macromolecules, or molecular complexes - as nodes, and the interactions between them as arrows. The arrows can represent a range of processes: chemical binding or unbinding, reaction catalysis, or regulation of activity. The processes result in the production, inter-conversion, transport, or consumption of the species within the network.

A set of reactions constitutes a *biochemical reaction network*. The manner in which the biochemical species interact is referred to as the *network topology*, and its organization is apparent if one rearranges the reactions in the form of an interaction graph. For example, the interaction graph of the Michaelis-Menten mechanism of Ex.1.2.1 is shown in Fig.2.1.

In order to obtain a *quantitative description*, one needs to know the rates at which the reactions occur. *In vivo*, the rate of a reaction depends on its kinetic constant, on the concentration of the reactants, and on physico-chemical conditions, such as temperature and pH. However, for *in silico* models, the physico-chemical conditions are fixed, such that rate laws can be described solely in terms of reactant concentrations and the kinetic rate constants.

All in all, a biochemical reaction network can be defined as:

Definition 2.1
(Biochemical Reaction Network) *A biochemical reaction network (BRN) is a pair* $(\mathcal{S}, \mathcal{R}, \boldsymbol{\alpha}, \boldsymbol{\beta})$,

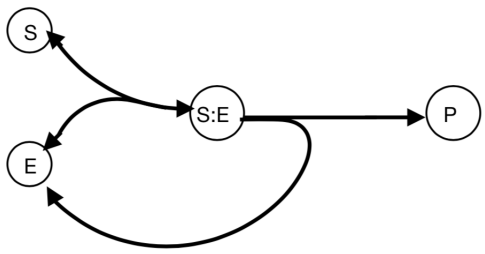


Figure 2.1: Interaction graph of the Michaelis-Menten enzymatic mechanism

such that:

(i) $\mathcal{S} = \{\mathrm{S}_1, \mathrm{S}_2, \ldots, \mathrm{S}_n\}$ *is a finite set of chemical species,*

(ii) $\mathcal{R} = \{\mathrm{r}_1, \mathrm{r}_2, \ldots, \mathrm{r}_m\}$ *is a finite set of reactions. Each reaction is a triple* $\mathrm{r}_j \equiv (\boldsymbol{\alpha}_j, \boldsymbol{\beta}_j, \kappa_j) \in \mathbb{N}^n \times \mathbb{N}^n \times \mathbb{R}_{\geq 0}$*, that writes as:*

$$\mathrm{r}_j : \alpha_{j1}\mathrm{S}_1 + \alpha_{j2}\mathrm{S}_2 + \ldots + \alpha_{jn}\mathrm{S}_n \xrightarrow{\kappa_j} \beta_{j1}\mathrm{S}_1 + \beta_{j2}\mathrm{S}_2 + \ldots + \beta_{jn}\mathrm{S}_n,$$

with vectors $\boldsymbol{\alpha}_j$ *and* $\boldsymbol{\beta}_j$ *being commonly referred to as the consumption, respectively the production vectors of reaction* r_j*, and* κ_j *denoting the kinetic constant of* r_j*,*

(iii) $\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{R}^{m \times n}$ *denote the network's consumption, respectively production, matrices; each element* α_{ji}*, respectively* β_{ji}*, denotes the quantity of species* S_i *begin consumed, respectively produced, by reaction* r_j*.*

Remark 2.1. The reaction network can be written compactly in matrix-vector form, as $\boldsymbol{\alpha}\mathrm{S} \xrightarrow{\boldsymbol{\kappa}} \boldsymbol{\beta}\mathrm{S}$, with S the species column vector, and $\boldsymbol{\kappa}$ the rates column vector.

Remark 2.2. The laws of thermodynamics state that all chemical reactions are reversible. Nevertheless, under certain environmental conditions (temperature, pressure, the availability of an enzyme), the reverse reaction proceeds at a negligible rate: nearly all of the reaction's reactants are used to form products, which makes it very difficult, even under extreme conditions, to reverse the reaction. In this case, it is reasonable to describe the reaction as being irreversible: this is why, in Definition 2.1, we assume elementary reactions to be irreversible. Reversible reactions will simply be represented by the couple of direct (forward) and reverse (backward) reactions, which we will note, for convenience, as:

$$\alpha_{j1}\mathrm{S}_1 + \alpha_{j2}\mathrm{S}_2 + \ldots + \alpha_{jn}\mathrm{S}_n \underset{k_j^-}{\overset{k_j^+}{\rightleftharpoons}} \beta_{ji}\mathrm{S}_1 + \beta_{j2}\mathrm{S}_2 + \ldots + \beta_{jn}\mathrm{S}_n.$$

Definition 2.2
(Reactants, products, modifiers) *In a reaction* r_j*, a species* S_i *is said to be:*

- *a* ***reactant****, if* $\alpha_{ji} > 0$
- *a* ***product****, if* $\beta_{ji} > 0$*.*

Additionally, species S_i *is usually considered to be a* ***modifier*** *in* r_j *if it participates in the reaction both as a reactant and as a product, but is neither produced nor consumed by it:* $\alpha_{ji} = \beta_{ji} > 0$*.*

Definition 2.3
(Stoichiometry matrix) *The stoichiometric matrix of a BRN* $(\mathcal{S}, \mathcal{R}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ *is formed by the stoichiometric coefficients of the reactions that constitute the network, and is defined as the* $m \times n$ *matrix* $\nabla = \boldsymbol{\beta} - \boldsymbol{\alpha}$.

Definition 2.4 (**State-change vector**)
Let $(\mathcal{S}, \mathcal{R}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ *be a biochemical reaction network. The* j^{th} *line of the stoichiometry matrix* ∇ *is called the* ***state-change vector*** *of reaction* $\mathrm{r}_j \in \mathcal{R}$*:* $\boldsymbol{\nu}_j \equiv (\nabla_{j1}, \nabla_{j2}, \ldots, \nabla_{jn})$*. The state-change vector denotes the change in the molecular population caused by one occurence of reaction* r_j*, i.e., if the system is in state* $\boldsymbol{x}$ *and one reaction* r_j *occurs, the system immediately jumps to state* $\boldsymbol{x} + \boldsymbol{\nu}_j$.

Remark 2.3. We note that the process of converting a reaction network into a stoichiometry matrix is lossy. Otherwise said, a reaction scheme is not *uniquely defined* by its stoichiometry matrix, which means it is not always possible to recover the original reaction scheme from a stoichiometry matrix. For example, while the reaction systems $X \to \emptyset$ and $2X \to X$ have the same associated matrix, their dynamical behavior is clearly different.

Example 2.1
Consider once again the Michaelis-Menten system:

$$\begin{cases} \mathrm{E} + \mathrm{S} \underset{\kappa_{-1}}{\overset{\kappa_1}{\rightleftharpoons}} \mathrm{E} : \mathrm{S} \\ \mathrm{E} : \mathrm{S} \xrightarrow{\kappa_2} \mathrm{E} + \mathrm{P} \end{cases}$$

Then $\mathcal{S} = \{S, E, E : S, P\}$, $\mathcal{R} = \{\mathrm{r}_1^+, \mathrm{r}_1^-, \mathrm{r}_2\}$, *with*

$$\begin{cases} \mathrm{r}_1^+ \equiv (\begin{pmatrix} 1 & 1 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 1 & 0 \end{pmatrix}, \kappa_1) \\ \mathrm{r}_1^- \equiv (\begin{pmatrix} 0 & 0 & 1 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 1 & 0 & 0 \end{pmatrix}, \kappa_{-1}) \\ \mathrm{r}_2 \equiv (\begin{pmatrix} 0 & 0 & 1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 0 & 1 \end{pmatrix}, \kappa_2), \end{cases}$$

meaning that $\boldsymbol{\alpha} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$, $\boldsymbol{\beta} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}$, *and* $\nabla := \boldsymbol{\beta} - \boldsymbol{\alpha} = \begin{pmatrix} -1 & -1 & 1 & 0 \\ 1 & 1 & -1 & 0 \\ 0 & 1 & -1 & 1 \end{pmatrix}$

Biochemical reaction networks represent a quantitative description of the system, which can be used to construct dynamical mathematical models. The *dynamic* behavior (*kinetics*) of a biochemical reaction network describes the changes in system states over time. A *system state*

is described by a set of variables at a given point in time. One state should contain enough information to predict all possible future behaviours. A state can be represented in different ways, and each model defines the contents of the system state, as well as the underlying process type governing the system's dynamics.

Generally speaking, the dynamics of a population model can be:

(i) either *discrete*, or *continuous* depending on whether the population quantity is modeled as discrete or continuous variable; discrete time is favored when the quantities of the model variables change only when specific events occur, while continuous time is favored when model variables are in constant flux;

(ii) either *deterministic*, if the output trajectory is fully determined by the initial state of the system, or *stochastic*, if starting from a given initial state, different trajectories can emerge, each trajectory having an associated probability of happening.

The notion of determinism, which is akin to *model behavior reproducibility*, is a foundation for much of scientific investigation. A model is called *deterministic* if its behavior is exactly reproducible. Although the behavior of a model depends on a specified set of conditions, no other factors influence it, so that repeated simulations under the same conditions are always in perfect agreement (*i.e.,* they are perfect replicates).

In contrast, *stochastic* models allow for randomness in the model behavior, which is influenced both by specified conditions and by unpredictable forces/noise. Consequently, each repetition of a stochastic simulation, under identical environmental conditions, will yield a distinct sample of system behavior.

As an illustrative example, consider the simple monomolecular reaction:

$$A \xrightarrow{\kappa} B \tag{2.1}$$

The process underlying this reaction can be assumed to be either *deterministic*, or *stochastic*. According to both the level of abstraction and to the hypothesized nature of change in the system state in time, one further distinguishes between *discrete* and *continuous* deterministic models.

Deterministic processes with a discrete change of system state can be represented using *Boolean models*, which approximate the dynamics of biological networks by considering each molecule (*e.g.,* gene or protein) in the network as either active/expressed (1) or inactive/not expressed (0). Under this representation, the questions that can asked about the system are of a *qualitative* nature: for example, instead of inquiring about the exact quantities of the chemical species present in the network, one is interested in their *presence* or *absence*. Despite their simplified underlying view of biological networks, and the fact that they introduce a coarse approximation by neglecting intermediate states, Boolean approaches give a meaningful insight into biological knowledge, having proven useful in analyzing system dynamics and reasoning about the stability and robustness of biological systems. For example, Boolean approaches enable the detection of singleton attractors (*i.e.,* fixed points), where the system is stable. Morevoer, Boolean networks have been successfully applied [180] in modeling gene regulatory and signaling networks in a variety of biological systems ([2],[73],[118],[133],[162],[163],[166],[181]), at both cellular and population levels ([27],[116]).

However, in this thesis, we will not address *discrete* deterministic models such as Boolean networks. Instead, we will focus on *continuous* deterministic models.

For deterministic models with *continuous* change in time, the state of the system no longer tracks the activation state of species, but rather their concentrations. Each species in the diagram is assigned a single *state variable*, $[S_i](t)$, denoting its concentration at time t. The collection of values of all single state variables, $\{[S_1](t), [S_2](t), [S_3](t), \ldots\}$, at any point in time t, constitutes the *state of the system*. Then, to each molecular species, corresponds an ordinary differential equation (ODE) that describes how its concentration changes over time, due to interactions with other species in the network. A reaction can have one of several effects on the molecular species partaking in it: besides the obvious phenomena of species consumption and production, there are also reactions that have a modifying effect on its reactants, such as autocatalytic production, (de)phosphorylation, or (un)binding. All of these reaction are thought of as *elementary*: they are reactions with a single mechanistic step. A general rule for constructing the differential equation of a species S, with respect to the type of reactions it participates in, is [43]:

$$\begin{aligned}\frac{d[S]}{dt} &= synthesis - degradation - phosphorylation \\ &\quad + dephosphorylation - binding + release, etc...\end{aligned} \tag{2.2}$$

Equation 2.2 indicates that each reaction in which species S is consumed or structurally changed (*e.g.,* phosphorylated, or bound to another species) contributes negatively to the evolution of its concentration, while the reactions that either produce S, or revert structural changes, contribute positively to its evolution.

The rate of each reaction must be represented by a *kinetic rate law*, which will have one or more *kinetic rate constants* associated with it. In the deterministic case, these rate constants denote the *speed* of the reaction: how frequently the reaction is expected to occur. As we will see in the next section, biochemical reactions are assumed to operate under the *law of mass action*, which states that the rate of a chemical reaction is directly proportional to the product of the activities or concentrations of the reactants.

For example, under the law of mass action, the changes in concentration for species A and B in Example 2.1, in the time interval dt, are given by the following system of ODEs:

$$\begin{cases} \frac{d[A]}{dt} = -\kappa[A] \\ \frac{d[B]}{dt} = \kappa[A] \end{cases}$$

which states that species B is produced at the same rate at which species A is consumed, and that that rate is proportional to both the concentration of the reaction's reactant A, and to its kinetic rate constant κ.

The third modeling choice assumes that the underlying process is non-deterministic, *i.e.*, that it exhibits a random component. Then, one can employ *stochastic models*, in which the state system is given by a vector of species' *molecule count* $(x_{S_1}(t), x_{S_2}(t), \ldots) \in \mathbb{N}_{\geq 0}$, instead of concentrations, and whose dynamics is governed by the *probability* of a reaction occurring

in a small time interval $(t, t + \delta t]$. In this case, the rate constant of a reaction r_j is no longer interpreted as the reaction's speed, but rather as an indicator of the *probability* that a tuple of molecules corresponding to r_j's reactant species will react according to r_j in the next infinitesimal time dt.

Then, the rate of reaction $A \xrightarrow{\kappa} B$ denotes the probability of a molecule A transforming into a molecule B in a time interval dt:

$$\mathrm{P}(x_a(t + dt) = x_a(t) - 1, x_b(t + dt) = x_b(t) + 1) = \kappa x_a(t),$$

with $(x_a(t), x_b(t))$ denoting the number of molecules of type A, respectively of type B, present in the system at time t.

As the latter two frameworks constitute the modelling approaches used in this manuscript (*i.e.,* we do not tackle Boolean models), we next formally elaborate on the concepts of *stochastic* and *deterministic* semantics of BRNs.

Chapter 3
Biochemical Reaction Networks: Semantics

3.1 Classical chemical kinetics

Conventional chemical kinetics operate under a continuous deterministic modelling framework, which involves no randomness in the development of states. That is, if a deterministic system is known at one time, then it is known at all subsequent times. It serves as the traditional way of representing the systems dynamic of complex biological systems that involve the interaction of many components: starting in the late 1970s, researchers began modeling cell physiology primarily using the continuous deterministic ODE approach, and creating increasingly detailed models over the next three decades.

Under this framework, the biochemical reaction network is modeled as a *continuous system* with *continuous time dynamics*. For a given BRN $(\mathcal{S}, \mathcal{R}, \boldsymbol{\alpha}, \boldsymbol{\beta})$, the *system state* is represented by a n-vector $(x_1(t), x_2(t), \ldots, x_n(t))$ of *continuous* variables that keep track of *reactant concentrations*, *i.e.,* $x_i(t)$ denotes the concentration of species S_i in the system at time t. The chemical reactions/interactions are also represented by *continuous processes*, *i.e.,* it is assumed that reactions occur *continuously* and *simultaneously*.

Each process r_j has an associated *deterministic rate constants* k_j - whose value is identical to that of the dimensionless constant κ_j introduced in Definition 2.1. It gives a measure of how frequently each type of reaction is expected to occur. The *velocity* of each reaction is specified using a *rate equation*. Deterministic reaction rates are described under the following assumptions:

(i) stochastic fluctuations in the system are negligible;

(ii) molecules are considered to be *point-like* entities;

(iii) the reaction volume border, as well as the resulting frontier effects, are neglected;

(iv) *spatial homogeneity*: the reaction volume is *well stirred*, *i.e.*, the reactants are equally distributed throughout the volume, meaning that the rate of each reaction is independent of the reactant position in space; the reaction rate can then be unambiguously referred to in the volume;

(v) the *continuum hypothesis*: each species is present in the system in a *large* amount - molecular abundance can be then described in terms of their continuously varying *concentration*, as opposed to an integer-valued molecule count.

Then, in a fixed volume, and under the negligible stochastic fluctuation, spatial homogeneity and continuum hypothesis, the reaction rate equations typically assume *mass-action* kinetics - or an enzyme kinetic law based on mass-action, such as the Michaelis-Menten or Hill kinetics. The *law of mass-action* states that the rate of a chemical reaction is proportional to the product of the reactants' concentration.

Under these conditions, the dynamic evolution of the system state (represented as an array of species' concentrations) can be described mathematically by a set of coupled ordinary differential equations, called the *reaction-rate equation* (RRE):

Definition 3.1.1 (**Continuous deterministic model (RRE)**)
Let $(\mathcal{S}, \mathcal{R}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ *be a biochemical reaction system and* $\boldsymbol{x}_0 = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n_{\geq 0}$ *an initial state of the system. Then, the* ***continuous deterministic model*** *is the solution of the set of* n *coupled differential equations given by*[1]*:*

$$\frac{d\boldsymbol{x}(t)}{dt} = \nabla^T f(\boldsymbol{x}(t)), \qquad (t \in \mathbb{R}_{\geq 0}) \tag{3.1}$$

and satisfying the initial condition $\boldsymbol{x}_0$*. The function* $f : \mathbb{R}^n_{\geq 0} \mapsto \mathbb{R}^m_{\geq 0}$ *denotes the flux of each reaction, in a given state. The flux* f_j *of a reaction* r_j *depends only on the concentrations of* r_j*'s reactants. Assuming mass-action kinetics,* f *is given by:*

$$f_j(\boldsymbol{x}) \equiv \kappa_j \prod_{i=1}^{n} x_i^{\alpha_{ji}}, \qquad (j \in \mathcal{R}) \tag{3.2}$$

which in turn means that $\frac{d\boldsymbol{x}}{dt}$ *is a multivariate polynomial of the species' concentrations.*

Example 3.1.1
For the Michaelis-Menten mechanism, the mass-action reaction fluxes write as

$$\begin{cases} f_{\mathrm{r}_{1+}} = \kappa_1 [E][S] \\ f_{\mathrm{r}_{1-}} = \kappa_{-1} [E:S] \\ f_{\mathrm{r}_2} = \kappa_2 [E:S] \end{cases}$$

and the ODE system describing the evolution of species' concentrations is given by

[1] ∇^T denotes the transpose of the stoichiometry matrix ∇

$$\frac{d\boldsymbol{x}}{dt} = \begin{pmatrix} -1 & 1 & 0 \\ -1 & 1 & 1 \\ 1 & -1 & -1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \kappa_1[E][S] \\ \kappa_{-1}[E:S] \\ \kappa_2[E:S] \end{pmatrix} =$$

$$= \begin{pmatrix} -\kappa_1[E][S] + \kappa_{-1}[E:S] \\ -\kappa_1[E][S] + \kappa_{-1}[E:S] + \kappa_2[E:S] \\ \kappa_1[E][S] - \kappa_{-1}[E:S] - \kappa_2[E:S] \\ \kappa_2[E:S] \end{pmatrix}$$

or, component-wise:

$$\begin{cases} \frac{d[S]}{dt} = \kappa_{-1}[E:S] - \kappa_1[E][S] \\ \frac{d[E]}{dt} = \kappa_{-1}[E:S] + \kappa_2[E:S] - \kappa_1[E][S] \\ \frac{d[E:S]}{dt} = \kappa_1[E][S] - \kappa_{-1}[E:S] - \kappa_2[E:S] \\ \frac{d[P]}{dt} = \kappa_2[E:S] \end{cases} \tag{3.3}$$

Remark 3.1.1. As stated in Def.3.1.1, the deterministic model is the *solution* of the initial value problem (IVP) of 6.5, which is guaranteed to uniquely *exist* under very weak constraints regarding the smoothness of the reaction rate laws - constraints that are usually satisfied by realistic models of reaction networks. However, *finding* the mathematical expression of this solution is not always easy. As very few nonlinear systems of ODEs can be solved *explicitly*, the analytically solvable class of ODE systems is reduced to *linear systems of equations*, which unfortunately show very limited interesting dynamical behaviors, and cannot model complex reaction networks of interest (they can only model unimolecular reactions). This means that only certain simple, but usually not biologically realistic, systems of equations can be solved analytically, in order to obtain an explicit formula describing the time course trajectory. In more complicated scenarios, the common practice is to solve a well-posed IVP *numerically*, using numerical integration methods (*e.g.,* Euler, Runge-Kutta, *etc.*...) that provide approximations of the solution of the IVP [26].

Challenges

Despite it being the traditional approach to biological modelling, continuous deterministic models fail to capture several important details of biological processes and their related experimental data. Besides the ubiquitous model *scalability* issue that stems from kinetic parameter estimation difficulties, the hypotheses needed for the continuous deterministic chemical kinetics can prove to be too constraining for biochemical systems.

For example, the spatial homogeneity assumption typically holds in stirred laboratory reaction vessels, and can be a good approximation in the cell, where the rapid diffusion process enables the mixing of molecular components. However, biological systems in general can contain a significant amount of spatial structure, meaning that in many cases the assumption of spatial homogeneity does not hold.

As for the continuum hypothesis, it serves towards allowing discrete changes in molecule number to be approximated by continuous changes in concentration, as individual reaction events cause infinitesimal changes in abundance. This assumption is valid when molar quantities of reactants are involved (recall that the number of Avogadro is 6.02×10^{23}), and is thus appropriate for cellular species with molecular counts of at least thousands. However, if the system is small enough that the molecular populations of at least *some* of the reactants do not exceed a unitary order of magnitude, discreteness and stochasticity may play important roles, in which case Eq.6.5 does not accurately describe the system's true behavior. As it turns out, it is often the case that reactants are not abundant: a number of cellular processes are governed by small populations of molecules numbering dozens or even less. For example, many genes, RNAs and proteins are typically present in low copy numbers. Consequently, deterministic modeling can prove to be inappropriate, as in some cases changes in molecule abundance should be treated as discrete steps in population size.

What's more, numerous wet-lab experiments enabled by recent advances in experimental methods have shown that stochastic effects generate phenotypic heterogeneity in cell behavior, and that even more important, cells can functionally exploit variability for increased fitness. They also demonstrate that the dynamics at the single cell (or even single molecule) level are intrinsically stochastic, or "noisy", and that that noise can have large implications for the qualitative dynamics - meaning that stochastic fluctuations are not negligible: biology seems to be inherently stochastic.

Therefore, we next introduce the *stochastic model* of biochemical reaction networks, as an alternative to continuous deterministic models.

3.2 Stochastic chemical kinetics

In the previous section, we argue that numerous experiments have revealed the inherently stochastic nature of biological phenomena, which is explained by the fact that molecules exhibit a certain degree of randomness in their dynamical behavior. Indeed, deterministic models of biochemical systems ignore the physical aspect involved in the dynamic behavior of biological systems, by assuming that reactions occur continuously. However, in a typical biological system, reaction events do not occur at regular intervals, which is why, for example, molecular motion can be represented as Brownian motion, even in a "well-mixed" solution. In the same way, bi-molecular reactions result from collisions of individual reactant molecules that approach closely enough (and in the correct orientation). What's more, most molecular collisions do not cause reactions. All these factors mean that the timing of reaction events is stochastic rather than deterministic: on a molecular scale, reactions are rare, hard-to-predict events. In many cellular processes, this randomness is averaged out over large numbers of reaction events, resulting in predictable system behavior. Deterministic models take advantage of exactly this averaging phenomenon: a reaction network that comprises significant amounts of reactant molecules will involve many simultaneous reaction events, in which case the network behavior corresponds to the average over these events and will be accurately described by deterministic differential equation models. However, models of processes that involve low-copy molecular species, such as gene expression, should account for this random variance/stochasticity.

Stochastic chemical kinetics applies to models centred on individual reaction events, and aims to describe the time evolution of a well-stirred chemically reacting system in a way that incorporates the system's discreteness and stochasticity [79]. In a stochastic model of a BRN, the additional random element that determines the development of subsequent processes lies precisely in the random manner in which collisions in a system of molecules take place - this in turn leads to a probability distribution of system states.

In the stochastic framework, for a given BRN $(\mathcal{S}, \mathcal{R}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ that is assumed to be confined to a constant volume Ω and in thermal equilibrium, the abundance of each chemical species will be described by *the number of molecules* in the reaction volume at time t, instead of using their concentrations. Thus, the system state is represented by a n-vector $(X_1(t), X_2(t), \ldots, X_n(t))$ of *discrete* variables that keep track of *molecular count*, *i.e.,* $X_i(t)$ denotes the number of molecules of species S_i in the system at time t. Chemical reactions are also assumed to occur *discretely*, *instantaneously* and at *separate times*.

Once again, changes in species populations are a consequence of the chemical reactions of the network. Each reaction r_j is mathematically characterized by two quantities: its state-change vector, as presented in Definition 2.4 and its propensity function.

The definition of the propensity function is regarded as *the fundamental premise of stochastic chemical kinetics* [79], as all the subsequent developments in stochastic chemical kinetics theory follow from it, via the laws of probability (a review of the basic definitions and concepts of probability theory needed for defininining the stochastic model can be found in Appendix A).

Definition 3.2.1 (**Propensity function**)
Let $(\mathcal{S}, \mathcal{R}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ be a biochemical reaction network modeling a well-stirred, constant-temperature reaction system, which is considered in a finite volume Ω. The ***propensity*** *of a reaction $\mathrm{r}_j \in \mathcal{R}$, denoted a_j, is defined such that:*

$$\begin{aligned} a_j(\boldsymbol{x})dt \equiv\ & \textit{the probability that, given the state of the system} \\ & \textit{at time } t,\ \boldsymbol{X}(t) = \boldsymbol{x},\ \textit{one reaction } \mathrm{r}_j \textit{ occurs inside} \\ & \Omega \textit{ in the next infinitesimal time interval } [t, t+dt). \end{aligned} \tag{3.4}$$

From a formal point of view, the dynamic hypothesis behind the propensity function is that the system trajectories are realizations of a continuous-time Markov chain (CTMC), which leads us to the definition of the *stochastic model* of a biochemical reaction network:

Definition 3.2.2 (**Stochastic model**)
Let $(\mathcal{S}, \mathcal{R}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ be a biochemical reaction network, and $\boldsymbol{x}_0 = (X_1, \ldots, X_n) \in \mathbb{N}^n$ an initial state of the system. Then, the ***discrete, stochastic model*** *is a continuous-time Markov chain $\{\boldsymbol{X}(t)\}$ with Markov graph (S, w, p_0), such that:*

(i) $S = \{\boldsymbol{x} \mid$ *is reachable from* $\boldsymbol{x}_0$ *in* $\mathcal{R}\}$,

(ii) $p_0(\boldsymbol{x}_0) = 1$,

(iii) $w(\boldsymbol{x}, \boldsymbol{y}) = \sum\{a_j(\boldsymbol{x}) 1_{\boldsymbol{y}=\boldsymbol{x}+\boldsymbol{\nu}_j} \mid j = 1, \ldots, m\}$.

Equivalently, the stochastic semantics of the network is the homogeneous CTMC that is defined by its transition laws:

$$\begin{cases} \forall t \geq 0, \\ \forall \boldsymbol{x} \in \mathbb{N}^n, \\ \forall 1 \leq j \leq m \ s.t.\ \boldsymbol{x} + \boldsymbol{\nu}_j \in \mathbb{R}^n \end{cases} \Rightarrow \begin{cases} \mathrm{P}(\boldsymbol{X}(t+dt) = \boldsymbol{x} + \boldsymbol{\nu}_j \mid \boldsymbol{X}(t) = \boldsymbol{x}) = a_j(\boldsymbol{\nu})dt, \\ \mathrm{P}(\boldsymbol{X}(t+dt) = \boldsymbol{x} \mid \boldsymbol{X}(t) = \boldsymbol{x}) = 1 - a_j(\boldsymbol{\nu})dt \end{cases}$$

Propensity functions are assumed to be of mass-action type: $a_j(\boldsymbol{x}) = c_j \prod_{i=1}^{n} \binom{X_i}{\alpha_{ji}}$, *where c_j is the stochastic kinetic rate constant - the existence of which is guaranteed by kinetic theory arguments and the well-stirred system hypothesis, and which is such that $c_j dt$ denotes the probability that a random tuple of molecules of the reactant species will react according to r_j in the next infinitesimal time dt. The binomial coefficient $\binom{X_i}{\alpha_{ji}}$ indicates the total number of possible tuples of reactant molecules, with the needed stoichiometry α_{ji} for each S_i, amongst the X_i available ones at time t.*

Remark 3.2.1. According to [79], even if the mathematical forms of the propensity functions are mass-action, like in the deterministic case, this should not be interpreted to imply that propensities are heuristic extrapolations of the reactions rates of deterministic chemical kinetics. Instead, the propensity functions are grounded in molecular physics, and it is rather the deterministic chemical kinetics formulas that are approximate consequences of the stochastic ones, than the other way around.

The probabilistic nature of the Equation 3.4 rules out making an exact prediction of a stochastic model's trajectory $\boldsymbol{X}(t)$. Instead, the usual stochastic approach focuses on the *grand probability function*, *i.e.*, the probability of the system being in a certain state $\boldsymbol{x} \in \mathbb{N}^n$ at a time t $\boldsymbol{x} \in \mathbb{N}^n$, and its moments.

Definition 3.2.3 (**Grand probability function (GPF)**)
For a stochastic BRN $(\mathcal{S}, \mathcal{R}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ *with underlying CTMC* $\{\boldsymbol{X}(t)\}$. *For any possible system state* $\boldsymbol{x} = (X_1, X_2, \ldots, X_n) \in \mathbb{N}^n$, *the* ***grand probability function*** *is defined as:*

$$\mathrm{P}(\boldsymbol{x}, t) \equiv \mathrm{P}(\boldsymbol{X}(t) = \boldsymbol{x}), \tag{3.5}$$

i.e., the function denoting the probability that at time t *the system will containt* X_1 *molecules of* S_1, X_2 *molecules of* $\mathrm{S}_2, \ldots$ *and* X_n *molecules of* S_n.

Definition 3.2.4 (**Moments of the GPF**)
The k^{th}*-order* ***moment*** *of a GPF* $\mathrm{P}(\boldsymbol{x}, t)$ *is defined as:*

$$\boldsymbol{x}_i^{(k)} \equiv \sum_{\boldsymbol{x} \in \mathbb{N}^n} \mathrm{P}(\boldsymbol{x}, t) \boldsymbol{x}_i^k. \tag{3.6}$$

Remark 3.2.2. $\boldsymbol{x}_i^{(k)}(t)$ denotes "the average (number)^k of S_i molecules in the system at time t", taken over many repeated runs starting in the same initial state. As the system is stochastic, the number $X_i(t)$ of S_i molecules at time t will vary between realizations, but the average of their k^{th} powers will approach $\boldsymbol{x}_i^{(k)}(t)$ in the limit of infinitely many runs[77]. Moments of order $k = 1$ and $k = 2$ are of particular interest: $\boldsymbol{x}_i^{(1)}(t)$ denotes the average number of S_i molecules in the system at time t, while the quantity $\Delta_i(t) \equiv (\boldsymbol{x}_i^{(2)}(t) - [\boldsymbol{x}_i^{(1)}(t)]^2)^{1/2}$ denotes the magnitude of the root-mean-square fluctuation magnitude about this average. That is to say, one can reasonably expect $X_i(t) \in [\boldsymbol{x}_i^{(1)}(t) - \Delta_i(t), \boldsymbol{x}_i^{(1)}(t) + \Delta_i(t)]$.

Applying the laws of probability to the fundamental premise of Def. 3.2.1 results in the chemical master equation (CME) [126], that gives the time evolution of the grand probability function $\mathrm{P}(\boldsymbol{x}, t)$. The CME encodes a continuous time discrete state Markov process, and acts as the probabilistic counterpart of the mass-action principle:

Definition 3.2.5 (**The Chemical Master Equation**)
The probability law of the Markov chain $\{\boldsymbol{X}(t)\}$ *of a biochemical reaction network* $(\mathcal{S}, \mathcal{R}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ *satisfies the equation:*

$$\frac{\partial \mathrm{P}(\boldsymbol{x}, t)}{\partial t} = \sum_{j=1}^{m} \left[a_j(\boldsymbol{x} - \boldsymbol{\nu}_j)\mathrm{P}(\boldsymbol{x} - \boldsymbol{\nu}_j, t) - a_j(\boldsymbol{x})\mathrm{P}(\boldsymbol{x}, t) \right]. \tag{3.7}$$

As it completely determines the function $\mathrm{P}(\boldsymbol{x}, t)$, solving the CME means obtaining the probability law of the system's trajectory, as well as analytical formulas describing their first and second order moments. However, the CME consists of a set of ODEs that has one equation for every possible combination of reactant molecules. Consequently, analytical solutions of the CME for the probability density function of $\boldsymbol{X}(t)$ can only be obtained for a very few very simple systems (*e.g.,* for linear systems [102], or for systems consisting of one (irreversible or reversible) nonlinear reaction [114]). What's more, even numerical solutions can prove to be prohibitively difficult in some cases [79].

As both the CME and the GPF moments prove to be virtually intractable, both analytically and numerically, the common practice is to construct numerical realizations of $\boldsymbol{X}(t)$ instead, *i.e.* to simulate the master equations' stochastic realizations of the evolution of the state (its "paths") via elementary reactions that occur with probabilities according to the CME. The resulting method employs rigurously derived Monte Carlo techniques to *numerically simulate* the very Markov process that the master equation describes analytically, meaning the simulation algorithm is fully equivalent to the CME, even if the latter is never explicitly used [77].

In order to do so, instead of using the grand probability function $\mathrm{P}(\boldsymbol{x}, t)$, a new function is defined, called *the reaction probability density*:

$$\begin{aligned} p(\tau, j \mid \boldsymbol{x}, t)d\tau \equiv\ & \text{the probability, given } \boldsymbol{X}(t) = \boldsymbol{x}, \text{ that the next} \\ & \text{reaction in the system to occur will be } \mathrm{r}_j, \text{ and} \\ & \text{that it will occur in the infinitesimal time interval} \\ & [t + \tau, t + \tau + dt), \end{aligned} \tag{3.8}$$

that describes the joint probability density function of two random variables: *(i)* the time to the next reaction, τ, and *(ii)* the index of the next reaction, j.

The intuition behind employing these two random variables lies in the fact that changes in the state of the underlying CTMC of a stochastic model occur at time instants $0 = t_0 < t_1 < \ldots$, where each t_i corresponds to a reaction r_j taking place.

Then, the a trajectory of the CTMC is entirely determined by the two quantities mentioned above:

- the time elapsed between two consecutive reaction events, $\tau_k \equiv t_{k+1} - t_k$, for $k \in \mathbb{N}$,
- the index $(j_k)k \in \mathbb{N}^*$ of the reaction taking place at time t_k).

These two random variables enable the construction of a system trajectory, via the equation:

$$\forall k \in \mathbb{N}, X(t_{k+1}) = X(t_k) + \nabla \boldsymbol{e}_{j_k}, \tag{3.9}$$

with $\boldsymbol{e}_i = (0, \ldots, 0, 1^{(i)}, 0, \ldots, 0)^T$ the basis vectors of $\mathbb{R}^n$.

By once again applying the laws of probability to the fundamental premise of 3.2.1, one can derive [2] an exact formula for the reaction probability density function $p(\tau, j \mid \boldsymbol{x}, t)$:

Theorem 3.2.1
Given a stochastic model of a BRN $(\mathcal{S}, \mathcal{R}, \boldsymbol{\alpha}, \boldsymbol{\beta})$, its reaction probability density function writes as:

$$p(\tau, j \mid \boldsymbol{x}, t) = a_j(\boldsymbol{x}) exp(-a_0(\boldsymbol{x})\tau), \tag{3.10}$$

with

$$a_0(\boldsymbol{x}) \equiv \sum_{j=1}^{m} a_j(\boldsymbol{x}),$$

which, as expected from a physical point of view, indicates that the probability of a particular reaction occurring in the next time step is proportional to its propensity.

The reaction probability density function of 3.10 and Equation 3.9 are then used to construct a rigorous algorithm for simulating the temporal development of the stochastic chemical system. Indeed, Equation 3.10 is the mathematical basis of the stochastic simulation, as it implies that τ is an exponential random variable with mean and standard deviation $\frac{1}{a_0(\boldsymbol{x})}$, and that j is a statistically independent integer random variable with point probabilities $\frac{a_j(\boldsymbol{x})}{a_0(\boldsymbol{x})}$ [79]. Samples of τ and j according to these distributions can be generated using Monte Carlo methods, and then plugged into Equation 3.9 in order to generate an individual stochastic trajectory of the system.

The simplest method of generating samples of τ and j is dubbed *the direct method*, in which two random numbers n_1 and n_2 are drawn from the uniform distribution in the unit interval, and then used to construct [79] τ and j[79], as follows:

$$\begin{aligned} \tau &= \frac{1}{a_0(\boldsymbol{x})} ln \frac{1}{n_1} \\ j &= \text{ the smallest integer satisfying } \sum_{k=1}^{j} a_k(\boldsymbol{x}) > n_2 a_0(\boldsymbol{x}). \end{aligned} \tag{3.11}$$

Such a generating method (or a mathematically equivalent one) is then used in Gillespie's stochastic simulation algorithm (SSA), in order to construct an exact numerical realization of the process $X(t)$. The outline of the SSA algorithm is as follows. First, a maximal simulation time t_{max} is set, and the initial state of the system is fixed. In Step 1, the algorithm proceeds by drawing the reaction probabilities and the next reaction time from their respective distributions, after which the reaction to implement is chosen according to the fractional rates of 3.11. In Step

[2] for the exact derivation, the reader is refered to the original Gillespie paper [77]

2, the time and system state are updated according to Equation 3.9. If the maximal simulation time has been exceeded, the algorithm stops, otherwise it returns to Step 1.

Below is an implementation of the SSA algorithm, as described in ([79], [77]). The function $draw\_uniform$ is used to generate a random number from the uniform distribution in the unit interval, and $gen_\tau(a_0, n_1)$ and $gen_j(\boldsymbol{a}, a_0, n_2)$ generate values for τ and j according to Equation 3.11.

Algorithm 1: Gillespie's SSA algorithm, direct version

Input: A BRN $(\mathcal{S}, \mathcal{R}, \boldsymbol{\alpha}, \boldsymbol{\beta})$, with initial state $\boldsymbol{x}_0$, and a maximal simulation time t_{max}
Output: An exact numerical realization $(X(t_k), t_k)_{k \in \mathbb{N}^*}$

`/* Step 0: initialize the time and the system state */`
$k \leftarrow 0$;
$t_k \leftarrow 0$;
$X(t_k) \leftarrow \boldsymbol{x}_0$;
`/* Step 1: while the maximal simulation time has not been exceeded, simulate the trajectory one reaction at a time */`
while $t_k < t_{max}$ **do**
  `/* evaluate the propensity functions in the current state */`
  for j = *1 to* m **do**
    $a_j \leftarrow c_j \prod_{i=1}^{n} \binom{X_i(t_k)}{\alpha_{ji}}$;
  end
  `/* evaluate` a_0 `in the current state */`
  $a_0 \leftarrow \sum_{j=1}^{m} a_j$;
  `/* draw` n_1 `and` n_2 `*/`
  $n_1 \leftarrow draw\_uniform(0, 1)$;
  $n_2 \leftarrow draw\_uniform(0, 1)$;
  `/* generate value for` τ `*/`
  $\tau_k \leftarrow gen_\tau(a_0, n_1)$;
  `/* generate value for` j `*/`
  $j_k \leftarrow gen_j(\boldsymbol{a}, a_0, n_2)$;
  `/* effect the next reaction by updating the time and the system state according to Equation 3.9 */`
  $t_{k+1} \leftarrow t_k + \tau_k$;
  `/* go back to Step 1 or finish simulation */`
  $k \leftarrow k + 1$;
end
return $(X(t_k), t_k)$

Challenges

The validity of the hypotheses needed for continuous deterministic models remains an issue even for discrete stochastic models. Indeed, in the stochastic modeling framework for BRNs one continues to operate under the spatial homogeneity and fixed volume hypotheses, as they allow the system state to be described by specifiying only the *molecular populations*, while ignoring the positions and velocities of the individual molecules, because the system is assumed to be well-stirred.

When compared to the deterministic approach, stochastic models are often less tractable: the number of states of the underlying Markov chain can be prohibitively large (even infinite) when compared to the size of the network, which in turn makes solving the CME analytically (and even numerically) virtually impossible, save for a few simple cases. The Monte-Carlo approach, that is used in Gillespie's algorithm, somewhat aleviates this issue, by generating system trajectories that are correct w.r.t. the system's stochastic semantics, and subsequently allowing to estimate its moments, by averaging a significant number of such trajectories. However, difficulties also arise with respect to Gillespie's algorithm. As the CME and SSA are both derived exactly from the fundamental premise of Equation 3.4, they are equivalent to each other. This means that the SSA imposes no approximations on the stochastic formulation of chemical kinetics, thus taking full account of the inherent fluctuations ignored by the deterministic formulation. The SSA's main advantage lays in its being a simple and compact way of simulating exact trajectories for systems with an intractable CME. It can also be argued that the construction of the SSA exploits the fundamental premise (3.4) in a more direct way than the CME. Indeed, Equation 3.4 describes the probability of the system being in a state x'' at time t'', *knowing that it was in state* x' *at time* t'. While this *step-like* description of the dynamics is inherent to the SSA by construction, it cannot be expressed using the CME (unless one re-initializes time at each step, in order to consider (x', t') to be the initial state, *i.e.*, $(x_0, 0) \longleftarrow (x', t')$).

However, the SSA is not without its flaws, the main one being that it is often slow, as it relies on simulating every individual reaction event. This means that for realistic cellular system, which contain large quantities of molecular species, or for significant simulation durations, its slowness becomes prohibitive. Consequently, a number of refinements of the SSA have been proposed, aiming at reducing the simulation computational requirements. These refinements can be split into two categories: *exact variations* and *approximate methods*.

Exact variations of SSA are essentially later elaborations on the most expensive computation step of the original algorithm of ([77],[79]): locating the next reaction to fire, cf. Equation 3.11. The classical SSA employed a linear search on the cumulative array of reaction propensities, in order to determine the next reaction event. Consequently, the first improvements to the original algorithm included replacing the linear search by a binary-tree search ([117]), and sorting the cumulative array (either using a pre-simulation step [31], or on-the-fly [124]).

Other exact variation methods aimed at reducing the average number of operations required to obtain the index of the next reaction event, employed modifications such as reusing the unused reaction times computed during one SSA step [74], or using factored-out, partial reaction propensities [156].

Exact variation methods significantly increase the SSA's speed for large networks (both in terms of species and reactions), albeit they prove to be limited improvements of the original SSA, as they still simulate reaction events one at a time.

On the other hand, *approximate simulation* strategies aim at finding a trade-off between the exactness of SSA and simulation speed. Among these methods, we mention:

- The τ-*leaping approximation* algorithm ([76],[29],[30],[4],[39],[132]), which instead of taking incremental steps in time, approximates the number of reaction events taking place in an interval of length τ, and performs all the reactions in that interval before updating the propensity functions. The value of τ is assumed to be small enough that there is no significant change in the value of the transition rates along during the time interval $[t, t + \tau]$;

- The *conditional difference* method [171], which approximates reversible processes by their corresponding effective net reactions, before simulating the newly obtained system using existing stochastic procedures (such as the SSA).

Finally, we note that for stiff systems - evolving on both fast and slow timescales, with the fastes modes being stable -, because accuracy requires τ to be small on the fastest timescales, even the τ-leaping performs poorly in terms of speed. Consequently, accelerations procedures have been developed for stiff systems: implicit τ-leaping ([159],[32]), which mirrors the implicit Euler method for systems of ODEs, and the *slow-scale SSA* ($ssSSA$) ([34],[33]), that directly simulates the slow reactions (using specially modified propensity functions), while ignoring fast reactions.

3.3 Stochastic vs deterministic models

When comparing the deterministic and stochastic formulations of chemical kinetics, it is common practice to view the former as describing the average behavior of the latter, *i.e.,* to interpret the solution of the RRE as the mean of the species population size over a large number of stochastic simulations. Consequently, one would expect equality between the solution of the RRE, $\boldsymbol{\chi}$ and the expectation [3] of the GPF, $E[\boldsymbol{X}(t)]$. However, in this section we will show that this equality holds only in one of two cases: in the *thermodynamical limit* ([80],[111]), or if all reactions in the network are unimolecular [126].

To do so, we derive the expectation of the marginal distribution of $\{\boldsymbol{X}(t)\}$, using the CME:

$$\begin{aligned}
\frac{d}{dt}E[\boldsymbol{X}(t)] &= \frac{\partial}{\partial t}\sum_{\boldsymbol{x}} \boldsymbol{x}\mathrm{P}(\boldsymbol{x},t) \\
&= \sum_{\boldsymbol{x}}\sum_{j=1}^{m}\left[a_j(\boldsymbol{x}-\boldsymbol{\nu}_j)\mathrm{P}(\boldsymbol{x}-\boldsymbol{\nu}_j,t)\boldsymbol{x} - a_j(\boldsymbol{x})\mathrm{P}(\boldsymbol{x},t)\boldsymbol{x}\right] \\
&= \sum_{\boldsymbol{x}}\sum_{j=1}^{m}\left[a_j(\boldsymbol{x}-\boldsymbol{\nu}_j)\mathrm{P}(\boldsymbol{x}-\boldsymbol{\nu}_j,t)(\boldsymbol{x}-\boldsymbol{\nu}_j)+\right. \\
&\qquad\qquad a_j(\boldsymbol{x}-\boldsymbol{\nu}_j)\mathrm{P}(\boldsymbol{x}-\boldsymbol{\nu}_j,t)\boldsymbol{\nu}_j- \\
&\qquad\qquad \left. a_j(\boldsymbol{x})\mathrm{P}(\boldsymbol{x},t)\boldsymbol{x}\right] \\
&= \sum_{j=1}^{m}\sum_{\boldsymbol{x}}\left[a_j(\boldsymbol{x}-\boldsymbol{\nu}_j)\mathrm{P}(\boldsymbol{x}-\boldsymbol{\nu}_j,t)(\boldsymbol{x}-\boldsymbol{\nu}_j)+\right. \\
&\qquad\qquad a_j(\boldsymbol{x}-\boldsymbol{\nu}_j)\mathrm{P}(\boldsymbol{x}-\boldsymbol{\nu}_j,t)\boldsymbol{\nu}_j- \\
&\qquad\qquad \left. a_j(\boldsymbol{x})\mathrm{P}(\boldsymbol{x},t)\boldsymbol{x}\right] \\
&= \sum_{j=1}^{m}\left[E[a_j(\boldsymbol{x})]\boldsymbol{x} + E[a_j(\boldsymbol{x})]\boldsymbol{\nu}_j - E[a_j(\boldsymbol{x})]\boldsymbol{x}\right]
\end{aligned} \quad , \tag{3.12}$$

that is, one obtains that:

$$\frac{dE[\boldsymbol{X}(t)]}{dt} = \sum_{j=1}^{m} E[a_j(\boldsymbol{X}(t))]\boldsymbol{\nu}_j. \tag{3.13}$$

In order to compare $\frac{dE[\boldsymbol{X}(t)]}{dt}$ to the corresponding deterministic expression for $\frac{d\boldsymbol{\chi}}{dt}$, we recall that the relation between $\boldsymbol{\chi}$, the molecular concentration vector, and $\boldsymbol{X}$, the molecular copy number vector writes as $\boldsymbol{\chi} \equiv \frac{\boldsymbol{X}}{\Omega}$, where Ω denotes the volume.

Deterministic and stochastic rate constants

At this point, it is worth examining the relation between *stochastic* and *deterministic* reaction constants. As previously mentioned, in the deterministic formulation of chemical kinetics, a

[3] For readability purposes, we will hereafter denote the first-order moment using the classical probability notation for expectation, $E[\boldsymbol{X}(t)]$, instead of $\boldsymbol{x}^{(1)}$.

reaction constant k_j is viewed as reaction *rate/speed*, whereas in the stochastic formulation, the constant c_j is indicative of the reaction's *probability per unit of time* [77]. Consequently, when switching between the stochastic and deterministic models, a conversion of rates must be performed. Intuitively, this transformation is justified by the stochastic rate's dependency on the volume and the arity of the underlying reaction.

The recipe for this scaling procedure lies in the physical rationale of Definition 3.2.1, stating that the propensity $a_j(\boldsymbol{x})dt$, which in itself is a function of the stochastic kinetic rate c_j, denotes the probability of reaction r_j taking place.

To start with, assume a unimolecular reaction

$$\mathrm{r}_j : S_1 \to \dots \tag{3.14}$$

It denotes the spontaneous conversion of a molecule S_1, and implies no molecular collision: consequently, its reaction rate is independent of the system volume Ω. This means that for unimolecular reactions, the stochastic kinetic constant c_j is numerically equal to the reaction-rate constant k_j of conventional deterministic chemical kinetics:

$$c_j = k_j \tag{3.15}$$

On the other hand, for a bimolecular reaction

$$S_1 + S_2 \to \dots, \tag{3.16}$$

the stochastic rate c_j will be inversely proportional to Ω, reflecting the fact that two reactant molecules will have a harder time finding each other inside a larger volume. Indeed, if there are x_1 molecules of S_1 and x_2 molecules of S_2 inside a volume Ω, then there will be $x_1 x_2$ distinct combinations of reactant molecules in Ω, meaning that the probability that reaction r_j will occur somewhere inside Ω in the next infinitesimal time interval dt is given by $x_1 x_2 c_j dt$. From this, the *average* rate at which r_j occurs inside Ω writes as

$$\langle x_1 x_2 \rangle c_j, \tag{3.17}$$

where $\langle\rangle$ denotes the average taken over an ensemble of stochastically identical systems. Then, dividing by Ω, one obtains *the average reaction rate per unit volume*:

$$\frac{\langle x_1 x_2 \rangle c_j}{\Omega}, \tag{3.18}$$

or, in terms of molecular concentrations, $\chi_i \equiv \frac{x_i}{\Omega}$:

$$\langle \chi_1 \chi_2 \rangle \cdot \Omega \cdot c_j \tag{3.19}$$

The deterministic reaction constant k_j is conventionally [77] defined as the average reaction rate per unit volume divided by the product of the average densities of the reactants:

$$k_j = \frac{\langle \chi_1 \chi_2 \rangle \cdot \Omega \cdot c_j}{\langle \chi_1 \rangle \langle \chi_2 \rangle}. \tag{3.20}$$

However, in the deterministic formulation no distinction is made between the average of a product and the product of the average, *i.e.,* $\langle \chi_1 \chi_2 \rangle = \langle \chi_1 \rangle \langle \chi_2 \rangle$, which in turn leads to the simplified expression relating the stochastic and deterministic reaction rate constants:

$$c_j = \frac{k_j}{\Omega}. \tag{3.21}$$

Similarly, for a tri-molecular reaction, one would have the transformation:

$$c_j = \frac{k_j}{\Omega^2}. \tag{3.22}$$

In general, for a reaction r_j of arity p, in which all the reactants have stoichiometry one, the conversion rule is given by:

$$c_j = \frac{k_j}{\Omega^{p-1}}. \tag{3.23}$$

However, we note that for a bimolecular reaction involving the same species, $S_1 + S_1 \to \ldots$, the number of distinct possible reactant pairs would have been $\frac{x_1(x_1-1)}{2} \approx x_1^2$, and we would have obtained the relation:

$$c_j = \frac{2k_j}{\Omega} \tag{3.24}$$

instead of the expression of Equation 3.21.

As it turns out, the conversion rule for any chemical reaction, no matter the reactant stoichiometry, is given by:

$$\frac{k_j}{\Omega^{\sum_{i=1}^{n} \alpha_{ji}-1}} = \frac{c_j}{\prod_{i=1}^{n} \alpha_{ji}!} \tag{3.25}$$

Equation 3.25 is justified by the fact that in general, the conversion between the stochastic and deterministic rate constants is such that for any reaction r_j, its stochastic rate function applied in a state $\boldsymbol{X}$ and the deterministic law of its conversion to a volume unit will relate through [78]:

$$f_j(\boldsymbol{\chi}) = \frac{a_j(\boldsymbol{X})}{\Omega} \tag{3.26}$$

However, the quantities $\Omega f_j(\boldsymbol{\chi})$ and $a_j(\boldsymbol{X})$ write as:

$$\begin{aligned} \Omega f_j(\boldsymbol{\chi}) &= \Omega \cdot k_j \cdot \prod_{i=1}^{n} \chi_i^{\alpha_{ji}} = k_j \cdot \Omega^{1-\sum_{i=1}^{n} \alpha_{ji}} \cdot \prod_{i=1}^{n} X_i^{\alpha_{ji}} \\ a_j(\boldsymbol{X}) &= c_j \cdot \prod_{i=1}^{n} \binom{X_i}{\alpha_{ji}} \end{aligned} \tag{3.27}$$

When all species in the system are abundant, *i.e.,* if $X_i \to \infty$, the propensity function can be approximated by:

$$a_j(\boldsymbol{X}) \approx \frac{c_j}{\prod_{i=1}^{n} \alpha_{ji}!} \cdot \prod_{i=1}^{n} X_i^{\alpha_{ji}} \tag{3.28}$$

By combining Equations 3.28 and 3.27, and applying the constraint of Equation 3.26, one indeed obtains the general conversion rule of Equation 3.25.

The thermodynamic limit

When deriving the conversion rule of Equation 3.25, we hypothesized that all species are abundant, *i.e.,* $X_i \to \infty$. It turns out that this hypothesis is linked to the first scenario in which the deterministic law of mass-action is valid w.r.t. to stochastic dynamics: the *thermodynamic limit* [5].

The thermodynamic limit is defined as the limit in which both the species populations X_i, and the system volume Ω approach infinity, but in such a way that the species concentrations $\frac{X_i}{\Omega}$ remain constant. Under this idealized state that provides a convenient approximation to macroscopic systems, it was shown that the deterministic model represents a correct approximation of the stochastic one. Below, we sketch the proof presented in the original paper [5].

For a reaction r_j, denote by $R_j(t)$ the number of times r_j occurs until time t. The value of R_j is a random variable described by the counting process satisfying:

$$R_j(t) = Y_j(\int_0^t a_j(\boldsymbol{X}(s))ds), \tag{3.29}$$

with Y_j being independent unit Poisson processes.

Then, the evolution of the state of the system satisfies:

$$\begin{aligned} \boldsymbol{X}(t) &= \boldsymbol{X}(0) + \sum_{j=1}^{m} R_j(t)\boldsymbol{\nu}_j \\ &= X(0) + \sum_{j=1}^{m} Y_j(\int_0^t a_j(\boldsymbol{X}(s))ds)\boldsymbol{\nu}_j \end{aligned} \tag{3.30}$$

If Ω denotes the size of the volume in which the reactions take place, one can introduce the scaled quantity denoting molecular *concentration* $\boldsymbol{\chi}(t) \equiv \frac{\boldsymbol{X}(t)}{\Omega}$. Using the scaling constraint $f_j(\boldsymbol{\chi}) = \frac{a_j(\boldsymbol{X})}{\Omega}$ of Equation 3.26, and the centered Poisson process $\tilde{Y}_j(t) = Y_j(t) - t$, one has that:

$$\begin{aligned} \boldsymbol{\chi}(t) &= \boldsymbol{\chi}(0) + \sum_{j=1}^{m} \Omega^{-1} Y_j(\Omega \int_0^t f_j(\boldsymbol{\chi}(s))ds)\boldsymbol{\nu}_j \\ &= \boldsymbol{\chi}(0) + \sum_{j=1}^{m} \Omega^{-1} \tilde{Y}_j(\Omega \int_0^t f_j(\boldsymbol{\chi}(s))ds)\boldsymbol{\nu}_j + \sum_{j=1}^{m} \boldsymbol{\nu}_j \int_0^t f_j(\boldsymbol{\chi}(s))ds \end{aligned} \tag{3.31}$$

However, the law of large numbers for the Poisson process implies that $\Omega^{-1}\tilde{Y}_j(\Omega u) \approx 0$, such that:

$$\boldsymbol{\chi}(t) \approx \boldsymbol{\chi}(0) + \sum_{j=1}^{m} \boldsymbol{\nu}_j \int_0^t f_j(\boldsymbol{\chi}(s))ds), \tag{3.32}$$

which, in the limit $\Omega \to \infty$, means that $\boldsymbol{\chi}$ follows the classical deterministic law of mass action of Definition 3.1.1:

$$\frac{d\boldsymbol{\chi}(t)}{dt} = \sum_{j=1}^{m} \boldsymbol{\nu}_j f_j(\boldsymbol{\chi}(t)). \tag{3.33}$$

Linear reaction networks

The second scenario in which classical chemical kinetics accurately describes the mean population sizes is when the underlying reaction system is *linear*, *i.e.,* when all of its reactions have arity 0 (reactions of type $\emptyset \to \ldots$) or 1 (unimolecular reactions).

Proposition 3.3.1
Let $(\mathcal{S}, \mathcal{R}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ be a linear BRN. In its corresponding stochastic model, let $\boldsymbol{X}(t)$ be the stochastic process described by the CME, for an initial condition denoted by $\boldsymbol{X}_0$. Let $\boldsymbol{\chi}(t)$ be the deterministic time-evolution of species' concentration described by the corresponding RRE, and starting in initial state $\boldsymbol{\chi}_0 = \frac{\boldsymbol{X}_0}{\Omega}$. Then, for every time instant t, $E[\boldsymbol{X}(t)] = \boldsymbol{\chi}(t)\Omega$.

Proof. ([148]) The propensity of a 0-ary reaction is a constant, while the propensity of a unary reaction is a linear function, meaning that $E[a_j(\boldsymbol{X})] = a_j(E[\boldsymbol{X}])$. What's more, according to Equation 3.25, stochastic and deterministic constants of linear reactions are identical, meaning that $a_j = f_j$. Then, according to Equation 3.12, for every time point t, one has that:

$$\begin{aligned} \frac{d}{dt}E[\boldsymbol{X}(t)/\Omega] &= \sum_{j=1}^{m} E[a_j(\boldsymbol{X}(t)/\Omega)]\boldsymbol{\nu}_j \\ &= \sum_{j=1}^{m} f_j(E[\boldsymbol{X}(t)/\Omega])\boldsymbol{\nu}_j \end{aligned}, \tag{3.34}$$

i.e., the expectation of $\boldsymbol{X}(t)/\Omega$ evolves according to a differential equation that assumes the RRE law of mass action. ■

Examples

To illustrate the above-mentioned relation between stochastic and deterministic models, we proceed with the analysis of some simple reaction networks.

First, assume a *linear* reaction network, comprised of a single reversible reaction:

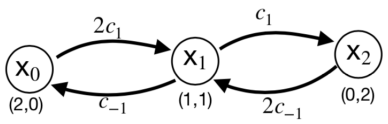


Figure 3.1: Markov graph for the linear reaction network of (3.35), with initial condition $\boldsymbol{x}_0 = (2, 0)$

$$A \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} B, \tag{3.35}$$

with initial condition $\boldsymbol{x}_0 = (2, 0)$, *i.e.,* there are initially 2 molecules of species A present in the system.

Its deterministic model is given by the ODE system:

$$\begin{cases} \frac{d[A]}{dt} = -k_1[A] + k_{-1}[B] \\ \frac{d[B]}{dt} = k_1[A] - k_{-1}[B] \end{cases} \tag{3.36}$$

.

This system can be solved analytically; its solution is:

$$\begin{cases} A(t) = \frac{2k_{-1}+2k_1 e^{-t(k_1+k_{-1})}}{k_1+k_{-1}} \\ B(t) = \frac{2k_1-2k_1 e^{-t(k_1+k_{-1})}}{k_1+k_{-1}} \end{cases} \tag{3.37}$$

The stochastic model of (3.35) is a CTMC $\{\boldsymbol{X}(t)\}$ with a Markov graph (S, ω, p_0), such that:

- $p_0(\boldsymbol{x}_0) = 1$;
- $S = \{\boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{x}_2\}$, with $\boldsymbol{x}_1 = (1, 1)$, $\boldsymbol{x}_2 = (0, 2)$;
- transition weights as depicted in Figure 3.1.

Using the notation $\mathrm{P}(\boldsymbol{x}, t) \equiv \mathrm{P}(\boldsymbol{X}(t) = \boldsymbol{x})$ of Definition 3.2.3, the CME is represented by the following system of equations:

$$\begin{cases} \frac{d\mathrm{P}(\boldsymbol{x}_0,t)}{dt} = -2c_1\mathrm{P}(\boldsymbol{x}_0, t) + c_{-1}\mathrm{P}(\boldsymbol{x}_1, t) \\ \frac{d\mathrm{P}(\boldsymbol{x}_1,t)}{dt} = 2c_1\mathrm{P}(\boldsymbol{x}_0, t) + 2c_{-1}\mathrm{P}(\boldsymbol{x}_2, t) - c_1\mathrm{P}(\boldsymbol{x}_1, t) - c_{-1}\mathrm{P}(\boldsymbol{x}_1, 0) \\ \frac{d\mathrm{P}(\boldsymbol{x}_2,t)}{dt} = -2c_{-1}\mathrm{P}(\boldsymbol{x}_2, t) + c_1\mathrm{P}(\boldsymbol{x}_1, t) \end{cases} \tag{3.38}$$

with initial condition $(\boldsymbol{x_0}, \boldsymbol{x_1}, \boldsymbol{x_2}) = (1, 0, 0)$.

Applying the reaction rate scaling formula of Equation 3.25 for unimolecular reactions means that the stochastic and deterministic constants have the same values: $c_1 = k_1$ and $c_{-1} = k_{-1}$.

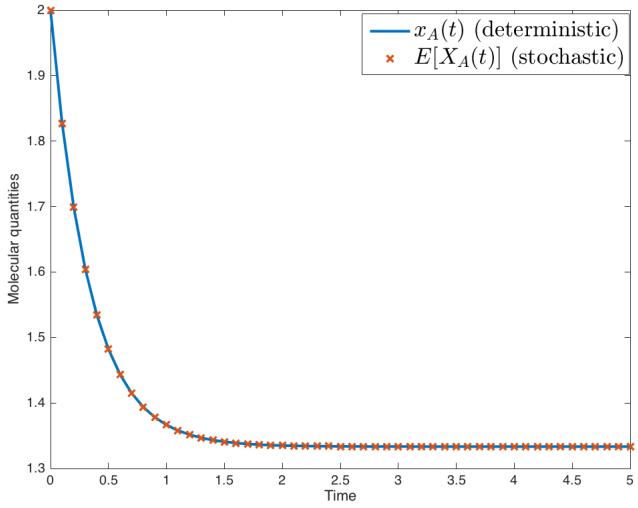


Figure 3.2: The deterministic solution and the stochastic mean population for species A in the linear reaction network (3.35) coincide. The parameter values used for simulation are $c_1 = k_1 = 1, c_{-1} = k_{-1} = 2$.

The ODE system of Equation 3.38 contains only linear equations, meaning it can be solved analytically. The solution writes as:

$$\begin{cases} \mathrm{P}(\boldsymbol{x}_0, t) = \frac{c_{-1}^2 + c_1^2 e^{-2t(c_1+c_{-1})} + 2c_1 c_{-1} e^{-t(c_1+c_{-1})}}{(c_1+c_{-1})^2} \\ \mathrm{P}(\boldsymbol{x}_1, t) = \frac{2c_1 c_{-1} - 2c_1^2 e^{-2t(c_1+c_{-1})} + 2c_1 e^{-t(c_1+c_{-1})}(c_1 - c_{-1})}{(c_1+c_{-1})^2} \\ \mathrm{P}(\boldsymbol{x}_2, t) = \frac{c_1^2 + 2c_1^2 e^{-t(c_1+c_{-1})} + c_1^2 e^{-2t(c_1+c_{-1})}}{(c_1+c_{-1})^2} \end{cases} \tag{3.39}$$

Then, the stochastic mean population of species A is indeed equal to the solution (3.37) of the deterministic model (3.36):

$$\begin{aligned} \mathrm{E}[\boldsymbol{X}_A(t)] &= 2 \cdot \mathrm{P}(\boldsymbol{x}_0, t) + 1 \cdot \mathrm{P}(\boldsymbol{x}_1, t) + 0 \cdot \mathrm{P}(\boldsymbol{x}_2, t) \\ &= \frac{2c_{-1} + 2c_1 e^{-t(c_1+c_{-1})}}{c_1 + c_{-1}} \\ &= A(t) \end{aligned} \tag{3.40}$$

In Figure 3.2 one can see that, as expected from Proposition 3.3.1, the stochastic mean population of A coincides with the deterministic trajectory of its concentration, as obtained by numerically simulating the mass-action equations of 3.36.

Now consider the bimolecular reversible reaction:

$$A + B \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} C, \tag{3.41}$$

with initial state $\boldsymbol{x}_0 = (1, 3, 0)$, ODE system:

$$\begin{cases} \frac{d[A]}{dt} = -k_1[A][B] + k_{-1}[C] \\ \frac{d[B]}{dt} = -k_1[A][B] + k_{-1}[C] \\ \frac{d[C]}{dt} = k_1[A][B] - k_{-1}[C] \end{cases} \tag{3.42}$$

and underlying CTMC as in Figure 3.3.

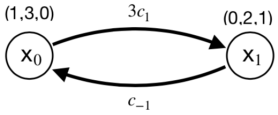


Figure 3.3: Markov graph for the bimolecular reaction network (3.41)

The resulting CME is given by the system of equations:

$$\begin{cases} \frac{d\mathrm{P}(\boldsymbol{x}_0,t)}{dt} = -3c_1\mathrm{P}(\boldsymbol{x}_0,t) + c_{-1}\mathrm{P}(\boldsymbol{x}_1,t) \\ \frac{d\mathrm{P}(\boldsymbol{x}_1,t)}{dt} = 3c_{-1}\mathrm{P}(\boldsymbol{x}_0,t) - c_{-1}\mathrm{P}(\boldsymbol{x}_1,t) \end{cases} \tag{3.43}$$

with initial condition $(\boldsymbol{x}_0, \boldsymbol{x}_1) = (1, 0)$. Its solution is given by:

$$\begin{cases} \mathrm{P}(\boldsymbol{x}_0,t) = \frac{c_{-1}}{3c_1+c_{-1}} + \frac{3c_1 e^{-t(3c_1+c_{-1})}}{3c_1+c_{-1}} \\ \\ \mathrm{P}(\boldsymbol{x}_1,t) = \frac{3c_1}{3c_1+c_{-1}} - \frac{3c_1 e^{-t(3c_1+c_{-1})}}{3c_1+c_{-1}} \end{cases} \tag{3.44}$$

Consequently, the stochastic mean population of species A is given by:

$$\mathrm{E}[\boldsymbol{X}_A(t)] = 1 \cdot \mathrm{P}(\boldsymbol{x}_0,t) + 0 \cdot \mathrm{P}(\boldsymbol{x}_1,t) \tag{3.45}$$

We note that, as the forward reaction in 3.41 is bimolecular, the rate constants need to be scaled according to the rules $c_1 = \frac{k_1}{\Omega}$, $c_{-1} = k_{-1}$. In order to compare the deterministic and stochastic models, the volume Ω is assumed to scale with the total molecule number. We assume that one volume unit v corresponds to 4 molecules. Therefore, for an initial state of the stochastic model $\boldsymbol{x}_0 = (1, 3, 0)$, the volume $\Omega = 4$ would be equivalent one unit, while for an initial state of $\boldsymbol{x}_0 = (10, 30, 0)$, the volume $\Omega = 40$ would take 10 units, *i.e.,* $\Omega = 10v$, and both the bimolecular reaction rate constant and the trajectory of a resulting stochastic simulation will be scaled by a factor of 10: $c_1 = \frac{k_1}{10}$, $\frac{\boldsymbol{X}(t)}{10}$.

The results of Figure 3.4 confirm that for bimolecular reactions, the mean population size does not coincide with the deterministic solution. However, as shown in Figure 3.5, as the system approaches the thermodynamical limit, the deterministic solution indeed approximates the mean population size.

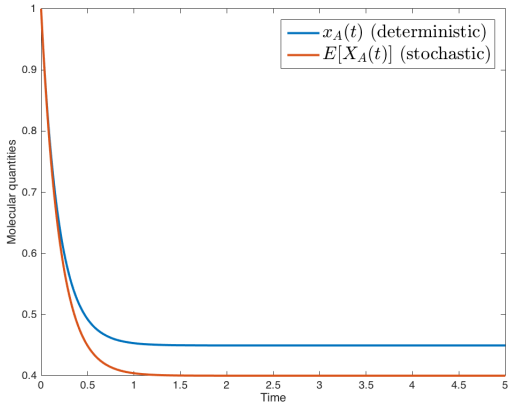


Figure 3.4: As species A is a reactant of the bimolecular reaction (3.41), its mean population size does not coincide with its deterministic solution.

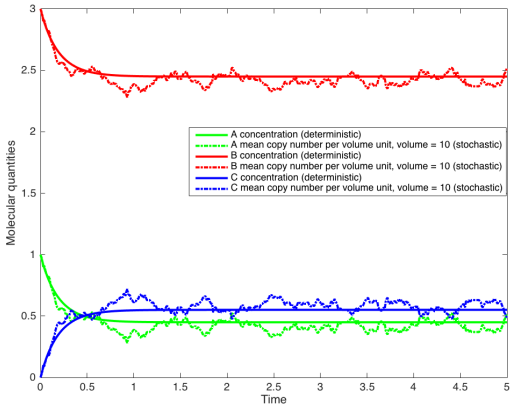
Conclusion

To summarize, in this chapter we have outlined the two fundamental formalisms for modeling biochemical reaction network dynamics: the classical framework, which is based on a deterministic chemical kinetics, and the stochastic framework, based on stochastic chemical kinetics. The underlying assumptions and formalisms of the stochastic and deterministic models are listed for comparison in Table 3.1.

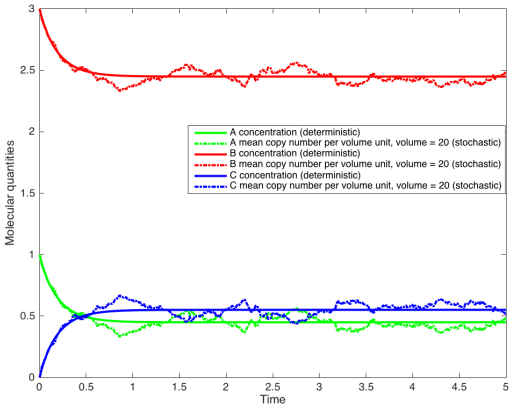
From a physical point of view, the stochastic formulation is superior to the deterministic one, as it captures and exploits the variability inherent to biological processes. The stochastic models are valid whenever the deterministic formulation is, but also when it is not.

However, from a mathematical point of view, the differential equations employed by the deterministic formulation are far more tractable than its stochastic counterpart, the chemical master equation. Nonetheless, when the modeled system involves large quantities of chemical species and numerous reactions, neither formulation is tractable purely through analytical methods; instead, one resorts to numerical simulation methods in order to "solve" the model.

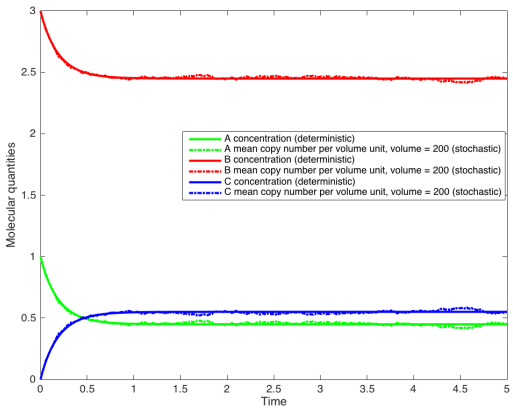
Finally, we have seen that, despite the fact that the deterministic formulation is considered to model the average behavior of molecular populations, classical chemical kinetics faithfully describes this mean in only one of two cases: when all reactions are unimolecular, or when the system is in the thermodynamical limit.



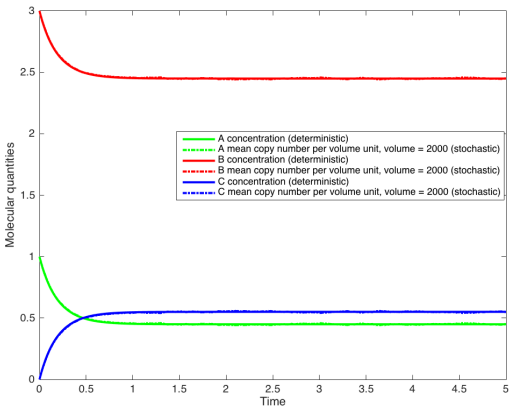
(a) $\Omega = 10v$



(b) $\Omega = 20v$



(c) $\Omega = 200v$



(d) $\Omega = 2000v$

Figure 3.5: Deterministic and stochastic models for Example 3.41, for different volume values Ω. For each value of $\Omega = \lambda v$, we plot the solution $\boldsymbol{\chi}(t)$ of the deterministic model with initial condition $\boldsymbol{\chi}_0 = (1, 3, 0)$, and the scaled mean population corresponding to the mean of 10 scaled trajectories $\frac{\boldsymbol{X}(t)}{\lambda}$ of a stochastic simulation, for initial state $\boldsymbol{X}_0 = (\lambda, 3\lambda, 0)$. Rate values are set to $k_1 = 1, k_{-1} = 2, c_1 = \frac{k_1}{\lambda}, c_{-1} = k_{-1}$. One notices that as the value of the scale factor λ increases, the system approaches the theoretical thermodynamical limit, in which the deterministic solution and the mean population size coincide.

Model	Deterministic	Stochastic
Mathematical formalism	ODEs	CTMC and CME
System State	species' concentration	number of molecules
System evolution	the future state of the system is uniquely "predictable", given present knowledge	includes randomness, every simulation is different
Analysis techniques	wide range	often rely on simulation
Validity assumptions	large population of species is involved	small numbers of molecules involved
Output data represents	population average	population variability

Table 3.1: Comparison between deterministic and stochastic modelling of Biochemical Reaction Networks

Chapter 4
Executable Biology: Computational models of Biochemical Reaction Networks

As previously stated, in order to benefit from the computational power and analysis techniques originating in Computer Science, Biochemical Reaction Networks need to be represented using computational, or *executable*, models. The dichotomies between mathematical and computational models have recently been subject to debate ([64], [10]). Therein, the authors argue that the differences between the two types of models stem from their primary *semantics*. Consequently, the primary semantics of mathematical models (such as ordinary differential equations) is considered to be *denotational*: such models describe the relationships between quantities of the system's components in terms of equations, but these equations do not determine an algorithm for solving them; in general, there may be different solution algorithms. In contrast, the primary semantics of computational models is *operational*: the model prescribes a sequence of instructions that can be executed on an abstract state machine, that can be implemented on a computer. Thus, they can be *directly executed*, as well as *formally analyzed* (*e.g.*, by model-checking). Another advantage of computational models resides in their ability of offering both a *quantitative* and a *qualitative* modeling of biological systems. For comparison, mathematical models generally represent *quantitative* relationships between entities.

For example, Boolean Networks are considered to be *computational* models, rather than *mathematical* ones. Other examples include process algebras [128], interacting state machines [89], hybrid systems [3], spatio-temporal models (which can be compartment-based [160], agent-based [161], or lattice-based [141]), Petri nets [147], and rule-based systems ([18],[52], [53], [50]). The authors of [64] dub the approach of constructing computational models of biological systems *"executable biology"*.

In this sense, we next present two of these computational modeling approaches applicable to Biochemical Reaction Networks, which we will use throughout this manuscript: Petri nets, and the rule-based modeling language Kappa.

4.1 Petri Nets

4.1.1 Motivation

By their very nature, Biochemical Reaction Networks have three distinctive characteristics [92]:

1. they have a *bipartite* structure: they consists of two different types of entities, *species* and their *interactions*
2. they are *concurrent*: several interactions can happen at the same time, independently;
3. they are *stochastic*: the timing behavior of the interactions is governed by stochastic laws.

Bipartiteness and concurrency are inherent characteristics of Petri Nets - a modeling methodology especially tailored for representing and simulating concurrent dynamic systems, and as such *stochastic Petri nets* represent a natural choice for the modeling of biochemical reaction systems. Nonetheless, due to computational efforts required to analyse stochastic models, two other Petri net abstractions are more popular: *qualitative models*, and *continuous models*. While the choice of the latter abstraction is justified by the fact that continuous models are commonly used to approximate stochastic behavior by a deterministic one, the former has the advantage of abstracting away from any time dependencies.

All in all, the advantages of employing the Petri net formalism for biochemical network modeling are as follows [92]:

- they offer a formal modeling and analysis technique for systems that display behaviors such as parallelism, concurrency, synchronization and resource sharing - all of which are inherent to biological processes;
- they have an intuitive and executable modeling style;
- they dispose of both true concurrency (partial order) semantics, and interleaving semantics - which allows to simplify analysis;
- they have an exact mathematical definition of their execution semantics, and dispose of well-developed, mathematically founded qualitative and quantitative analysis techniques based on formal semantics;
- theoretical results concerning them are plentiful, and their properties have been and still are extensively studied; most notably, they allow for coverage of both structural and behavioral properties, as well as their relations;
- last, but not least, they dispose of reliable tool support.

4.1.2 Qualitative Petri Nets

Qualitative Petri nets abstract away timed information: they are stoichiometric, or purely causal, and are based on a graph-theoretical description of the underlying system's topology. Consequently, the most abstract representation of a biochemical network is *qualitative* and is minimally described by its topology: herein, network structure represents static knowledge about interactions within a cell.

An ordinary Petri Net is a weighted directed bipartite graph, in which the nodes represent either *places* or *transitions*. The distinction between places and transitions reflects the difference between passive and active system components: while places are usually used to model passive components such as conditions, transitions stand for active system components such as atomic actions. In the case of biochemical reaction networks, conditions are represented by chemical compounds (*e.g.* proteins, protein compounds) acting as precursors, products or enzymes, while (atomic) actions denote chemical reactions, for example (dis)association, (de)phosphorylation, or transforming precursors into products:

Definition 4.1.1 (**Petri Net ($\mathcal{QPN}$), Syntax**)
A Petri Net is a quadruple $\mathcal{N} = (P, T, f, m_0)$, where:

- *P and T are disjoint finite sets: $P \cup T \neq \emptyset, P \cap T = \emptyset$; the elements of P are called places, and the elements of T are called transitions*
- *the function $f : (P \times T) \cup (T \times P) \to \mathbb{N}_0$ specifies the non-negative integer weights of the arcs connecting places to transitions and vice-versa*
- *$m_0 : P \to \mathbb{N}_0$ denotes the initial marking of the net; $m(p)$ specifies the number of tokens in place p, for the marking m*

The directed arcs connect precursors to reactions (incoming arcs), or reactions to products (outgoing arcs). Arc weights are read as the multiplicity of the arc, and reflect known stoichiometries. Thus, the arc weight 0 marks the absence of an arc, while arc weight 1 is the default value, and is usually omitted when drawing a Petri net.

To the original *discrete* Petri net model described in Def.4.1.1, David and Alla [55, 56] added a *continuous* model ($\mathcal{CPN}$), in which both the marking of a place and the weight of an arc are no longer integers, but real positive numbers: $f : ((P \times T) \bigcup (T \times P)) \to \mathbb{R}_0^+$, $m_0 : P \to \mathbb{R}_0^+$.

By construction, in the case of Petri Nets modeling biochemical reaction networks, the pre-places of a transitions correspond to the reaction's precursors, and its post-places to the reaction's products. For example, in Figure 4.1, we show the Petri nets modeling the reactions $A \to B$ (left), respectively $A \leftrightarrow B$ (right).

The following notations are used to formalize the notion of pre- and post-places:

Definition 4.1.2 (**Pre- and post- set**)
For a node $x \in P \cup T$:

- *${}^\bullet x := \{y \in P \cup T \mid f(y, x) \neq 0\}$ is the pre-set of x*

Figure 4.1: Qualitative Petri nets modeling a simple reaction $A \rightarrow B$ (Left) and a reversible reaction $A \leftrightarrow B$ (Right)

- $x^\bullet := \{y \in P \cup T \mid f(x,y) \neq 0\}$ *is the post-set of* x

All in all, from the definition above, one can distinguish four types of sets:

- ${}^\bullet t$, the *pre-places* of a transition t, consisting in the underlying reaction's precursors
- $t^\bullet$, the *post-places* of a transition t, consisting in the reaction's products
- ${}^\bullet p$, the *pre-transitions* of a place p, consisting of all reactions producing this species
- $p^\bullet$, the *post-transitions* of a place p, consisting of all reactions consuming this species

In order to describe the dynamics of a Petri net, another object needs to be introduced - the token, denoted by a solid dot ($\bullet$) inside the circles representing places. Tokens that inhabit the system at a given time t constitute the marking of the net at t, and describe the *state of the system* at that time. In ordinary Petri nets, tokens are indistinguishable; their role is to indicate the presence (or absence) of a condition, resource or signal. For Petri nets that model biochemical systems, the quantity of tokens in a place may be interpreted according to the type of net being used: in the case of a $\mathcal{QPN}$, the integer number of tokens in a place is a proxy for the discrete amount of the substance corresponding to that place, *i.e.* how many molecules of the substance are present in the system at the current time.

The behavior/semantics of a Petri net is defined by a firing rule, which consists of two parts: the precondition and the firing itself. The firing of a transition moves tokens from its pre-places to its post-places, while possibly changing the number of tokens: it "consumes" tokens from the pre-places, and "produces" tokens in its post-places, just as a chemical reaction consumes reactants in order to create products.

In the case of qualitative Petri nets ($\mathcal{QPN}$), as presented in Def.4.1.1, the standard semantic does not associate a time with neither transitions nor with the sojourn of tokens at places.

Definition 4.1.3 (**($\mathcal{QPN}$, Firing rule)**)
Let $\mathcal{N} = (P, T, f, m_0)$ be a Petri net. A transition $t \in T$ is enabled in a marking m, written as $m[t\rangle$, if $\forall p \in {}^\bullet t, m(p) \geq f(p,t)$. A transition t may (but is not forced to) fire in marking m, if it is enabled. This firing yields a new marking m', with $\forall p \in P, m'(p) = m(p) - f(p,t) + f(t,p)$. For quantitative (untimed) Petri nets, the firing happens atomically/instantaneously: it does not consume any time.

If the firing of (the enabled) transition t in marking m changes the system state into marking m', it is common practice to write $m \xrightarrow{t} m'$. This notation is naturally extended to *sequences of transitions*: one writes $m \xrightarrow{t_1 t_2 \ldots t_n} m'$ as an abbreviation of $m \xrightarrow{t_1} m_1 \xrightarrow{t_2} m_2 \ldots \xrightarrow{t_n} m'$.

The repeated firing of transitions establishes the behavior of the net. In the *interleaving execution semantics*, only one transition fires at a time: in each step, one of the enabled transitions is selected non-deterministically, then fired. If there are no more enabled transitions, the net deadlocks; otherwise, the procedure repeats. This semantics contains all possible interleavings of transitions, and as such describes the **totally asynchronous behavior** of the net. The interleaving semantics of Petri nets can be formally described a *Kripke structure*: a graph whose nodes represent the reachable states of the system, whose edges represent state transitions, and which is equipped with a labelling function mapping each node to a set of properties that hold in the corresponding state.

Definition 4.1.4 (**Transition system/Kripke structure**)
A Kripke structure is a 6-tuple $\mathcal{M} = (S, \Sigma, T, I, AP, l)$*, with:*

- S *a set of states (finite or infinite)*
- Σ *a set of actions*
- $T \subseteq S \times \Sigma \times S$ *a set of transitions*
- $I \subseteq S$ *a set of initial states*
- AP *a set of atomic propositions*
- $l : S \to 2^{AP}$ *a labeling function*

Definition 4.1.5 (**Interleaving semantics**)
Let $\mathcal{N} = (P, T, f, m_0)$ *be a Petri net. We associate with it a transition system* $\mathcal{M} = (S, \Sigma, \Delta, I, AP, l)$*,where:*

- $S = \{m \mid m : P \to \mathbb{N}_0\}$
- $\Sigma = T$
- $\Delta = \{(m, t, m') \mid \forall p \in P, m(p) \geq f(p,t) \bigwedge m'(p) = m(p) - f(p,t) + f(t,p)\}$
- $I = m_0$
- $AP = P$
- $l(m) = \{p \in P \mid m(p) > 0\}$

When $(m, t, m') \in \Delta$*, transition* t *is enabled in marking* m*, and its firing produces the marking* m'*. We also write* $m \xrightarrow{t} m'$*.*

The dynamic behavior of Petri nets can also be described (and completely analyzed) using algebraic equations - a feature which enables analysis techniques based on linear algebra.

The main idea is to represent the net structure by its *incidence matrix*, and then derive matrix equations that govern the dynamic behavior of concurrent systems modeled by Petri nets. We note that the solvability of the incidence matrix equations remains somewhat limited, partly because of the inherent non-deterministic nature of Petri nets, and partly because of the non-negativity integer constraint imposed on the solutions.

Definition 4.1.6 (**Incidence matrix**)
The incidence matrix of a Petri net $\mathcal{N} = (P, T, f, m_0)$, with n transitions (i.e., $\mid T \mid = n$) and m places (i.e., $\mid P \mid = m$) is a $m \times n$ integer matrix $\nabla : P \times T \to \mathbb{Z}$, whose entries are given by $\nabla_{ij} = f(j,i) - f(i,j)$.

We note that the incidence matrix of a Petri net that models a Biochemical Reaction Network is the same as the transpose of the stoichiometry matrix of the BRN.

Definition 4.1.7 (**Parikh-vector of a sequence of transitions**)
Let $\mathcal{N} = (P, T, f, m_0)$ be a net and $t = t_1 t_2 \dots t_n$ a sequence of transitions. The Parikh-vector is given by the function $\sigma_t : T \to \mathbb{N}$, which associates to each transition $t_i \in T$, its number of occurences in t.

Definition 4.1.8 (**State Equation**)
Consider a Petri net $\mathcal{N} = (P, T, f, m_0)$, for which a marking is written as a P-vector. Then, the net marking/system state m obtained after firing an (enabled) transition sequence $t = (t_1 t_2 \dots t_n)$ writes as:

$$m = m_0 + \nabla \cdot \sigma_t,$$

with σ_t the Parikh-vector of transition sequence t.

Qualitative Analysis

The main advantage of using Petri nets to model biochemical networks resides in their support for the qualitative analysis of many properties and problems commonly associated with concurrent systems. The available mathematically founded analysis techniques enable the decision of an exhaustive range of properties, by relying (almost) exclusively on the topology of the underlying net.

Such structural analysis enables identification of properties that are conserved during the execution of the modeled system, such as:

1. *Liveness*: A Petri net is said to be live, if no matter what marking has been reached from m_0, it is possible to ultimately fire any transition of the net by firing some further sequence. This means that a live Petri net is guaranteed to be deadlock-free, no matter what firing sequence is chosen.

2. *Boundedness:* Checking there is no infinite accumulation of tokens in a place.

3. *P-invariants:* Identifying an ensemble of places in which the total amount of tokens is invariant.

4. *T-invariants:* Identifying a set of transitions that have to fire from some initial marking m, to enable the Petri net to return to m.

5. *Reachability:* Deciding whether a certain marking is reachable from another marking. Reachability can be used to determine whether certain outcomes are possible for a given Petri net and a given initial marking.

Such structural properties can provide meaningful insight to biologists[146]. The most immediate use of structural analysis is that it enables the identification of conflicting transitions and concurrent processes. In addition, by checking for the structural properties listed above, one can answer an array of relevant questions the modeled biological system:

1. Checking the *liveness* property can answer questions related to the causality of biological reactions, *e.g.,"Does inhibition of a reaction cause a state in which some other reactions cannot be executed?"*[146].

2. By definition, *reachability* analysis allows the biologist to decide on whether a certain state of the system is reachable from the initial state. It can also help answers w.r.t. the effects of inhibiting certain activities: does inhibiting a certain activity render certain states of the system inaccessible, *e.g.*, *"If we block the immune system, can we still reach a state where the parasite is cleared from the blood system?"*[146].

3. Verifying the *boundedness* of Petri nets can inform the biologist about the accumulation of metabolites at a toxic level.

4. Locating *P-invariants* can help identify parts of metabolic pathways, or sets of metabolites that are likely to be affected by the inhibition of a reaction (by locating P-invariants containing input/output places of the inhibited reaction).

5. Identifying *T-invariants* can help detect continuous operations and cycles in the BRN.

Since any real system behavior happens in time, qualitative Petri nets can be seen as a supplementary intermediate step that allows for model validation, "at least from the viewpoint of the biochemist accustomed to quantitative modeling only"[92]. Generally, qualitative models provide a complementary approach to the analysis of the system, when compared to the quantitative ones: whereas the latter uses execution/simulation/"the token game" to experience and establish confidence in (or dismiss) the model behavior, the former allows for formal exhaustive analysis and validation of the network. For example, in [75], the authors propose a discrete qualitative Petri net model of the influence of the Raf Kinase Inhibitor Protein (RKIP) on the Extracellular signal Regulated Kinase (ERK) signalling pathway, the analysis of which is then used to derive the sets of initial concentrations required by the corresponding continuous ordinary differential equation model.

We note that it is common practice to distinguish quantitative and qualitative models (and handle them separately). As such, qualitative models are employed when kinetic parameters

are either deficient or unavailable, and are widely accepted as an intermediary step for larger models. Conversely, quantitative models become prevalent as soon as a substantial fraction of the necessary kinetic parameters is known - for example, kinetic reaction rate constants, species' concentrations and equilibrium constants. The remaining kinetic parameters are then usually either estimated, or, if available, taken from the taxonomically nearest neighbour species. The quantitative approach seems to be the favorite choice: the vast majority of published biochemical models are quantitative.

In [150], the authors address the issue of bridging the gap between qualitative and quantitative models, by demonstrating how to develop quantitative models of biochemical networks in a systematic manner, starting from the underlying qualitative ones. To do so, they exploit the well-established structural Petri net analysis technique of transition invariants, which may be interpreted as a characterization of the system's steady state behaviour.

4.1.3 Quantitative Petri nets

Having succesfully validated a qualitative Petri net model, its quantitative counterpart can be derived, by adding timed information. This can be achieved in one of two ways: by assigning either deterministic or exponentially distributed stochastic timings to the net's transitions.

For *biochemically interpreted* continuous Petri nets, this amounts to assigning a *firing rate* function to each transition:

$$v : T \mapsto H, \text{ with } H = \bigcup_{t \in T} \{h_t \mid h_t : \mathbb{R}^{|^\bullet t|} \mapsto \mathbb{R}\},$$

the set of all firing rate functions, and $v(t) = h_t, \forall t \in T$.

For *biochemically interpreted* stochastic Petri nets, $H = \bigcup_{t \in T} \{h_t \mid h_t : \mathbb{N}_0^{|^\bullet t|} \mapsto \mathbb{R}^+\}$, and the functions h_t are dubbed *stochastic transition rates*.

In both cases, the rate function of a transition depends on its underlying reaction's rate constant, and is defined according to mass-action kinetics:

- for continuous Petri nets: $h_t = k_t \prod_{p \in {}^\bullet t} m(p)$;
- for stochastic Petri nets: $h_t = c_t \prod_{p \in {}^\bullet t} \binom{m(p)}{f(p,t)}$.

Then, the execution semantics of stochastic Petri nets follows the standard firing rule of the interleaving semantics of qualitative Petri nets, while that of continuous Petri nets is defined by the system of ODEs imposed on the net by the firing rate function (*i.e.*, $\frac{dm(p)}{t} = \sum_{t \in {}^\bullet p} f(t,p)v(t) - \sum_{t \in p^\bullet} f(p,t)v(t)$).

As such, these quantitative Petri nets are nothing more than a *structured description* of either the ODE system or the CTMC that describes the time evolution of the species' quantities.

In Chapter 8, we will aim at defining a *general* piecewise-synchronous execution of Petri nets, in which the dynamic behaviour of a Biochemical Reaction Network is not simply imposed on the net (as is the case above), but is rather recreated by the execution semantics.

4.2 Rule-based modeling and the Kappa language

4.2.1 Rule-based modeling

A second class of executable representations of Biochemical Reaction Networks that we will use in this manuscript is given by *rule-based models*, which were originally developed to address the limitations of traditional approaches for modeling chemical kinetics in cell signaling systems. Namely, that very large network models are needed in order to capture all possible consequences of the multiple molecular interactions that occur in such systems. In rule-based models, this issue is circumvented by representing protein-centric interactions in terms of local rules.

Rule-based modeling languages borrow the concept of "rule" from chemistry: a rule represents the *mechanism* of an interaction, and as such emphasizes the distinction between the transformation of a structure fragment and the reaction instance that results when that fragment is transformed within the context of specific entities that contain it [149]. In other words, rules are *context-free*: there is no need to describe the whole state of the entities participating in a rule in order to apply it; instead, only the partial information needed to trigger the rule is needed.

Consequently, an advantage of using rules to concisely capture the dynamics of molecular interactions, is that it avoids having to exhaustively enumerate the reachable chemical species of a system - a necessity in traditional modeling approaches.

As such, they enable reasoning on the behavior of systems that can suffer from a combinatorially explosive complexity, as is the case for models of biochemical reaction networks. What's more, their usefulness in biochemical modeling is accentuated by the fact that they are a concise, transparent, and easily extensible modeling framework.

Remark 4.2.1. In this study, however, we choose to focus on the latter distinguishing feature of rule-based models, instead of the former. In Chapter 5, we will argue for the use of rule-based models for prototyping genetic circuits, not due to the *context-free* property of rules, but rather due to the intuitive, chemical-like syntax of rule-based models, that results from reasoning in terms of *protein-protein* interactions, and which leads to the construction of transparent and easily understandable models of biochemical reaction networks.
For example, consider once again the simple reaction of Example (1.1):

$$A + B \underset{\kappa_{-1}}{\overset{\kappa_1}{\rightleftharpoons}} A.B,$$

We recall the observation made with respect to the notation of the complex species $A.B$, namely that in the classical modeling frameworks (both deterministic and stochastic), the notation $A.B$ for the complex species is simply a name, and should not be considered as indicative of the reaction's mechanistic details (*i.e.*, A binds B) - meaning that one

could choose any other name for the complex species: *e.g.*, $A + B \underset{\kappa_{-1}}{\overset{\kappa_1}{\rightleftharpoons}} C$.

However, in Kappa (the rule-based modeling language we will use in this manuscript), the same reaction writes as:

$$A(x), B(x) \underset{\kappa_{-1}}{\overset{\kappa_1}{\rightleftharpoons}} A(x[1]), B(x[1]),$$

where A and B are defined as two agents denoting their respective species, both containing a site x, on which binding can occur.

The syntax of Kappa models mirrors the well-known manner of writing chemical reactions, with the added advantage of including the mechanistic details of the interaction between proteins. Indeed, the Kappa notation for the product species is indicative of the reaction's binding mechanism: $A(x[1]), B(x[1])$ indicates that there is bond between site x of protein A and site x of protein B, which is denoted by its identifier, 1.

What's more, the rule-based representation does not introduce a supplementary name for the product species, a feature which proves to be essential when dealing with very large biological networks, which would otherwise demand the use of one notation per occurring species, thus rendering the model opaque, non-intuitive and difficult to modify or extend. By contrast, rule-based models of even large networks offer an intuitive and easily understandable (and modifiable) representation of the underlying chemical mechanisms.

The common feature of all rule-based modeling languages is that the structure of their composing entities is represented as a graph, and that rules consist in graph-rewriting directives. We next present the rule-based modeling language Kappa ([52],[53],[50]), which will be our rule-based language of choice throughout Chapter 5. It is a graph-rewrite language for representing, reasoning about, and simulating systems of interacting structured entities (graphs) [149], which we will use to specify biochemical reaction networks, by explicitly describing chemical species in form of site-graphs. Kappa was originally developed to reason about systems of protein-protein interaction: the "structured entities" mentioned above were meant to denote complexes of non-covalently bound proteins, as they arise in signaling and assembly processes.

However, in its most general definition, the Kappa language provides a versatile framework for thinking about the statistical dynamics induced by the mass-action of interacting heterogeneous agents, regardless of the chosen agent interpretation [149].

4.2.2 The Kappa language: Syntax and Operational Semantics

At the heart of the Kappa language lies the conceptualization of proteins as agents with an interface of sites representing distinct interaction capabilities, such as binding and post-translational modifications. An agent can be thought of as an *atomic* entity: it can not be decomposed into further agents. A *complex* is then a connected graph of agents. The analogy between Kappa and chemistry then becomes clear: in chemistry, an atom would correspond to a Kappa agent, and a molecule, to a Kappa complex.

Formally, Kappa agents connect into *site graphs* via their sites. A site graph can be either fully specified, in terms of the interface and state of its agents, in which case it represents a *molecular species*, or partially specified - in which case it is considered a *pattern*. A rule $r : E_1 \to E_2$ consists of two site graphs. The state of a system is given by a *mixture*: a graph consisting of an ensemble of disconnected graph sites, each representing an instance of a molecular species. A rule r is applied to a mixture by *embedding* its left-hand side, E_1, into the mixture, *i.e.*, by finding a match in the mixture of all agent types, site names and states (including binding states) mentioned in E_1. Executing a rule $r : E_1 \to E_2$ consists in replacing the part of the mixture matched by E_1 with E_2.

We introduce the syntax and operational semantics of Kappa, in a process-like notation. We assume a finite set of agent names $\mathcal{A}$, representing different kinds of proteins; a finite set of sites $\mathcal{S}$, corresponding to protein domains; a finite set of internal states $\mathbb{I}$, and Σ_ι, Σ_β two signature maps from $\mathcal{A}$ to $\mathcal{P}(\mathcal{S})$, listing the domains of a protein which can bear an internal state, respectively a binding state. We denote by Σ the signature map that associates to each agent name $A \in \mathcal{A}$ the combined interface $\Sigma_\iota(A) \cup \Sigma_\beta(A)$.

Definition 4.2.1 (**Kappa agent**)

A *Kappa agent* $A(\sigma)$ is defined by its type $A \in \mathcal{A}$ and its *interface* σ. In $A(\sigma)$, the interface σ is a sequence of sites s in $\Sigma(A)$, with internal states (as subscript) and binding states (as superscript). The internal state of the site s may be written as s_ϵ, which means that either it does not have internal states (when $s \in \Sigma(A) \setminus \Sigma_\iota(A)$), or that it is not specified. A site that bears an internal state $m \in \mathbb{I}$ is written s_m (in such a case $s \in \Sigma_\iota(A)$). The binding state of a site s can be specified as s^ϵ, if it is *free*, otherwise it is bound (which is possible only when $s \in \Sigma_\beta(A)$). There are several levels of information about the binding partner: we use a binding label $i \in \mathbb{N}$ when we know the binding partner, or a wildcard bond $-$ when we only know that the site is bound. The detailed description of the syntax of a Kappa agent is given by the following grammar:

$$
\begin{array}{rcll}
a & ::= & N(\sigma) & \text{(agent)} \\
N & ::= & A \in \mathcal{A} & \text{(agent name)} \\
\sigma & ::= & \varepsilon \mid s, \sigma & \text{(interface)} \\
s & ::= & n_\iota^\lambda & \text{(site)} \\
n & ::= & x \in \mathcal{S} & \text{(site name)} \\
\iota & ::= & \epsilon \mid m \in \mathbb{I} & \text{(internal state)} \\
\lambda & ::= & \epsilon \mid - \mid i \in \mathbb{N} & \text{(binding state)}
\end{array}
$$

The symbol ϵ is generally omitted.

Definition 4.2.2 (**Kappa expression**)

A *Kappa expression* E is a set of agents $\mathbf{A}(\sigma)$ and fictitious agents $\emptyset$. Thus, the syntax of a Kappa expression is defined as follows:

$$E ::= \varepsilon \mid a, E \mid \emptyset, E.$$

The structural equivalence $\equiv$, defined as the smallest binary equivalence relation between expressions that satisfies the rules:

$$E, A(\sigma, s, s', \sigma'), E' \equiv E, A(\sigma, s', s, \sigma'), E'$$

$$E, a, a', E' \equiv E, a', a, E'$$

$$E \equiv E, \emptyset$$

$$\frac{i, j \in \mathbb{N} \text{ and } i \text{ does not occur in } E}{E[i/j] \equiv E}$$

$$\frac{i \in \mathbb{N} \text{ and } i \text{ occurs only once in } E}{E[\epsilon/i] \equiv E}$$

stipulates that neither the order of sites in interfaces, nor the order of agents in expressions matters, that a fictitious agent might as well not be there, that binding labels can be injectively renamed and that *dangling bonds* can be removed.

Definition 4.2.3 (**Kappa pattern, mixture and species**)

A *Kappa pattern* is a Kappa expression which satisfies the following five conditions: (i) no site name occurs more than once in a given interface; (ii) each site name s in the interface of the agent A occurs in $\Sigma(A)$; (iii) each site s which occurs in the interface of the agent A with a non empty internal state occurs in $\Sigma_\iota(A)$; (iv) each site s which occurs in the interface of the agent A with a non empty binding state occurs in $\Sigma_\lambda(A)$, and (v) each binding label $i \in \mathbb{N}$ occurs exactly twice if it does at all — there are no dangling bonds.

A *mixture* is a pattern that is fully specified, meaning that each agent A documents its full interface $\Sigma(A)$, that a site can only be free or tagged with a binding label $i \in \mathbb{N}$, that a site in $\Sigma_\iota(A)$ bears an internal state in $\mathbb{I}$, and that no fictitious agent occurs.

A *species* is a connected mixture: for each two agents A_0 and A, there is a finite sequence of agents $A_1, \ldots, A_k$ s.t. there is a bond between a site of A_k and a site of A, and for each $i = 0, 1, \ldots, k-1$, there is bond between a site of agent A_i and a site of agent A_{i+1}. The state of the system can then be considered as a multiset of species.

Definition 4.2.4 (**Species occurring in a pattern**)

Given Kappa patterns E_s and E_p, if E_s defines a Kappa species, and E_s is a substring of E_p, we say that a species E_s *occurs* in a pattern E_p.

Definition 4.2.5 (**Kappa rule**)

A Kappa rule r is defined by two Kappa site-graphs (*i.e.*, patterns or species) E_l and E_r, and a rate $k \in \mathbb{R}_{\geq 0}$, and is written: $r = E_l \rightarrow E_r @ k$.

Given a Kappa rule $r = E_l \to E_r@k$, we may assume without any loss of generality that both E_l and E_r can be written as $C_1, \ldots, C_k$, respectively $P_1, \ldots, P_j$, where each C_i and each P_i is a connected pattern. Then, a species E occurs in the pattern E_l (respectively E_r) of a rule r if there is a C_i (respectively a P_i) sucht that $C_i = E$ (respectively $P_i = E$).

A rule r is well-defined, if the expression E_r is obtained from E_l by finite application of the following operations:

(i) *creation*: some fictitious agents $\emptyset$ are replaced with some fully defined agents of the form $\mathrm{A}(\sigma)$; moreover, σ documents all the sites occurring in $\Sigma(A)$, and each site in $\Sigma_\iota(A)$ bears an internal state in $\mathbb{I}$;

(ii) *unbinding*: some occurrences of the wild card and binding labels are removed;

(iii) *deletion*: some agents, bearing only free sites, are replaced with the fictitious agent $\emptyset$;

(iv) *modification*: some non-empty internal states are replaced with some non-empty internal states;

(v) *binding*: some free sites are bound pair-wise by using binding labels in $\mathbb{N}$.

In order to apply a rule $r = E_l \to E_r@k$ to a mixture E, the structural equivalence $\equiv$ is used in order to bring the participating agents to the front of E (with their sites in the same order as in E_l), to rename binding labels if necessary, and to introduce a fictitious agent for each agent that is created by r.

The result is an equivalent expression E' that matches the left hand side term E_l. The matching relation writes as $E \models E_l$, and is defined inductively as follows:

$$
\begin{aligned}
E &\models \epsilon \\
a \models a_l \wedge E \models E_l &\Rightarrow a, E \models a_l, E_l \\
\emptyset &\models \emptyset \\
\sigma \models \sigma_l &\Rightarrow N(\sigma) \models N(\sigma_l) \\
\sigma &\models \varepsilon \\
s \models s_l \wedge \sigma \models \sigma_l &\Rightarrow s, \sigma \models s_l, \sigma_l \\
\iota \models \iota_l \wedge \lambda \models \lambda_l &\Rightarrow n_\iota^\lambda \models n_{\iota_l}^{\lambda_l} \\
\iota_l \in \{\varepsilon, \iota\} &\Rightarrow \iota \models \iota_l \\
\lambda_l \in \{-, \lambda\} \wedge \lambda \neq \epsilon &\Rightarrow \lambda \models \lambda_l
\end{aligned}
$$

E' is then replaced by $E'[E_r]$, which is defined as follows:

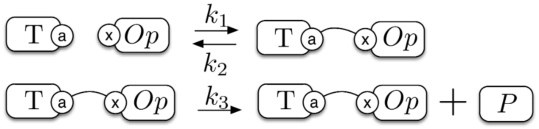
$$
\begin{aligned}
E[\varepsilon] &= E \\
(a, E)[a_r, E_r] &= a[a_r], E[E_r] \\
\emptyset[a_r] &= a_r \\
a_r[\emptyset] &= \emptyset \\
N(\sigma)[N(\sigma_r)] &= N(\sigma[\sigma_r]) \\
\sigma[\varepsilon] &= \sigma \\
(s, \sigma)[s_r, \sigma_r] &= s[s_r], \sigma[\sigma_r] \\
n_\iota^\lambda[n_{\iota_r}^{\lambda_r}] &= n_{\iota[\iota_r]}^{\lambda[\lambda_r]} \\
\iota_r \in \mathbb{I} &\Rightarrow \iota[\iota_r] = \iota_r \\
\lambda_r \in \mathbb{N} \cup \{\varepsilon\} &\Rightarrow \lambda[\lambda_r] = \lambda_r \\
\lambda[-] &= \lambda
\end{aligned}
$$

Definition 4.2.6 (**Kappa system**)
A *Kappa system* $\mathcal{R} = (\boldsymbol{x}_0, \mathcal{O}, \{r_1, \ldots, r_n\})$ is given by an initial mixture $\boldsymbol{x}_0$, a set of Kappa patterns $\mathcal{O}$ called *observables*, and a finite set of rules $\{r_1, \ldots, r_n\}$.

We give below an equivalent definition of a Kappa system, that we will use to define the stochastic semantics in the next section:

Definition 4.2.7 (**Kappa system**)
A *Kappa system* $\mathcal{R} = (\pi_0^{\mathcal{R}}, \{r_1, \ldots, r_n\})$ is given by a finite distribution over intial mixtures $\pi_0^{\mathcal{R}} : \{M_{01}, \ldots, M_{0k}\} \mapsto [0, 1]$, and a finite set of rules $\{r_1, \ldots, r_n\}$.

A simple example of a Kappa system is presented in Figure 4.2, while Figure 4.3 depicts a rule being applied to a Kappa mixture. In the next section, we formally define the stochastic semantics of Kappa.




```
'T.Op'   T(a[.]),Op(x[.])  <-> T(a[1]),Op(x[1])      @ 'k_1','k_2'
'Pprod'  T(a[1]),Op(x[1]),. -> T(a[1]),Op(x[1]),P()  @ 'k_3'
```

Figure 4.2: An example of a rule-based model. The transcription factor T binds to the operator's site x via site a and, when bound, it initiates the production of protein P.

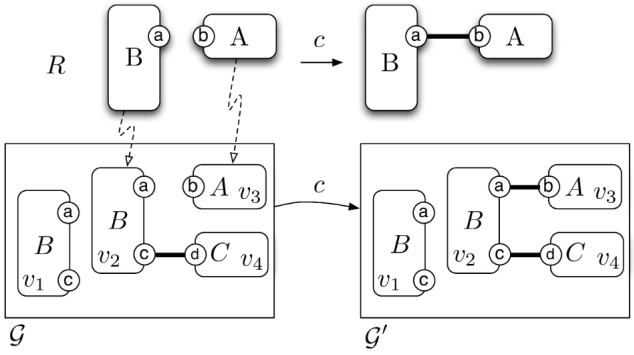


Figure 4.3: In Kappa, rule application is akin to pattern rewriting: applying a certain rule is carried out by finding an instance of the rule's left-hand side pattern in the reaction mixture (*pattern searching*) and then replacing it with the rule's right-hand side species (*rewriting*), according to the rule's kinetic rate.

4.2.3 The Kappa language: Stochastic Semantics

We now present the stochastic semantics of a Kappa system, as given in [63]. This definition involves observing the continuous time Markov chain (CTMC) generated by a weighted-labeled transition system on a countable space. Intuitively, any rule-based system can be expanded to an equivalent reaction system, albeit with potentially infinitely many species and reactions. The stochastic semantics of a Kappa system is then the CTMC $\{X_t\}$ assigned to that equivalent reaction system. We note that even though the semantics of a Kappa system is defined as that of the equivalent reaction system, in practice, using Kappa models can be advantageous for several reasons: they are easy to read, write, edit or compose, they can compactly represent potentially infinite sets of reactions or species, and, perhaps most importantly, they can be symbolically executed.

4.2.3.1 Preliminaries

Definition 4.2.8 (**Weighted-labeled transition system (WLTS)**)
A weighted-labeled transition system is a tuple (X, L, w, π_0), with:

- X a countable set of states (the state space);
- L a set of labels;
- $w : X \times L \times X \mapsto \mathbb{R}_0^+$ the weighing function, mapping two states and a label to a real positive value;
- $\pi_0 : X \mapsto [0, 1]$ an initial probability distribution

The assumptions made in Definition 4.2.8 are as follows:

- a label fully defines a transition: $\forall x \in X, l \in L$, there is at most one $x' \in X$ such that $w(x, l, x') > 0$;
- the system is finitely branching: the sets $\{x \in X \mid \pi_0 > 0\}$ and $T_{\hat{x}} = \{(l, x') \in L \times X \mid w(\hat{x}, l, x') > 0\}$ are finite, $\forall \hat{x} \in X$.

Then, the activity of a state x_i is defined as:

Definition 4.2.9 (**State activity**)
The activity of a state $x_i \in X$, denoted $a : X \mapsto \mathbb{R}_0^+$, is the sum of all weights originating at x_i:

$$a(x_i) \equiv \sum_{x_j \in X, l \in L} w(x_i, l, x_j)$$

The WLTS definition implicitly defines a *transition relation* $\rightarrow \subseteq X \times X$, such that:

$$(x_i, x_j) \in \rightarrow \text{ if and only if } \sum_{l \in L} w(x_i, l, x_j) > 0 \tag{4.1}$$

One can differentiate the initial set of states $I \subseteq X$:

$$I = \{x \in X \mid \pi_0(x) > 0\}$$

Definition 4.2.10 (**Rate matrix of a WLTS**)
Given a WLTS $W = (X, L, w, \pi_0)$, one can assign it a CTMC matrix, $R : X \times X \mapsto \mathbb{R}$, defined by:

$$R(x_i, x_j) \equiv \sum_{l \in L} w(x_i, l, x_j)$$

We can now refer to the generated stochastic Markov process, written as a continuous-time random variable $\{X_t\}$ over the countable state space X. If $\mathrm{P}(X_t = x_i)$ denotes the probability that the process takes the value x_i at time t, then the following equalities hold:

$$\begin{aligned}
&\mathrm{P}(X_0 = x_i) = \pi_0(x_i) \\
&\mathrm{P}(X_{t+dt} = x_j \mid X_t = x_i) = R(x_i, x_j)dt, \text{ when } i \neq j \\
&\mathrm{P}(X_{t+dt} = x_i \mid X_t = x_i) = 1 - \sum_{j \neq i} R(x_i, x_j)dt
\end{aligned}$$

For an execution of a WLTS, a *trace* denotes an observation of the sequence of visited states, of labels of the executed transitions, and of time points in which the transition happened:

Definition 4.2.11 (**A trace of a WLTS**)
Assume a WLTS $W = (X, L, w, \pi_0)$, its CTMC, and a number $k \in \mathbb{N}$. A trace of length k is defined as $\tau \in (X \times L \times \mathbb{R}_0^+)^k \times X$,and writes as:

$$\tau = x_0 \xrightarrow{l1, t_1} x_1 \ldots x_{k-1} \xrightarrow{l_k, t1+\ldots+t_k} x_k$$

If a trace τ is such that $\pi_0(x_0) > 0$, and $\forall 0 \leq i \leq k-1, w(x_i, l_i, x_{i+1}) > 0$, then τ is said to belong to the set of traces of W ($\tau \in T(W)$).

As we are dealing with continuous random variables, the probability of any single trace is 0, meaning one cannot assign a probability distribution to the traces in $T(W)$. Consequently, the notion of *cylinder set of traces over intervals of time* is introduced:

Definition 4.2.12 (**Cylinder set of traces**)
If $\mathbb{IR}$ is the set of all nonempty intervals in $\mathbb{R}_0^+$, the cylinder set of traces $\tau_{\mathbb{IR}} \in (X \times L \times \mathbb{IR})^k \times X$, is defined such that

$$\tau_{\mathbb{IR}} = x_0 \xrightarrow{l_1,I_1} x_1 \dots x_{k-1} \xrightarrow{l_k,I_k} x_k \tag{4.2}$$

denotes the set of all traces $\tau = x_0 \xrightarrow{l1,t_1} x_1 \dots x_{k-1} \xrightarrow{l_k,t1+\dots+t_k} x_k$, such that $t\, i \in I_i, 1 \leq i \leq k$.

If the cylinder of traces $\tau_{\mathbb{IR}}$ is such that $\pi_0(x_0) > 0$, and $\forall 0 \leq i \leq k-1, w(x_i, l_i, x_{i+1}) > 0$, then we say that $\tau_{\mathbb{IR}}$ belongs to the cylinder set of traces of W ($\tau_{\mathbb{IR}} \in T_{\mathbb{IR}}(W)$).

With all this in place, we define a *probability measure* over $\Omega(T_{\mathbb{IR}}(W))$, where $\Omega(T_{\mathbb{IR}}(W))$ is the smallest Borel σ-algebra that contains all the cylinder sets of traces in $T_{\mathbb{IR}}(W)$:

Definition 4.2.13 (**Trace density semantics over a WLTS**)
Given a WLTS $W = (X, L, w, \pi_0)$, and a number $k \in \mathbb{N}$, the probability of the cylinder set of traces $\tau_{\mathbb{IR}} \in T_{\mathbb{IR}}(W)$, specified as in (4.2), is given by:

$$\begin{aligned}\pi(\tau_{\mathbb{IR}}) &= \pi(x_0 \xrightarrow{l_1,I_1} x_1 \dots x_{k-1} \xrightarrow{l_k,I_k} x_k) \\ &= \pi_0(x_0) \prod_{i=1}^{k} \frac{w(x_{i-1}, l_i, x_i)}{a(x_{i-1})} \left(e^{-a(x_{-1} \cdot \inf(I_i))} - e^{-a(x_{i-1}) \cdot \sup(I_i)} \right)\end{aligned}$$

We note that, since the probability density function of the residence time of x_{i-1} is equal to $a(x_{i-1})e^{-a(x_{i-1})}$, $\int_{I_i} a(x_{i-1})e^{-a(x_{i-1})\cdot t}dt = e^{-a(x_{-1}\cdot \inf(I_i))} - e^{-a(x_{i-1})\cdot \sup(I_i)}$ denotes the probability of exiting state x_{i-1} in a time interval I_{i-1}

4.2.3.2 Population-based stochastic semantics of Kappa

A correct definition of the system's quantitative dynamics requires knowing the values of reaction rate constants k_j, but also a way of counting the number of times each rule can be applied to a mixture - this is done using the notion of *embedding* between a mixture and an expression:

Definition 4.2.14 (**Embedding**)
Let $Z = a_1, \dots, a_m$ and $Z_l = c_1, \dots, c_n$ be two patterns with no occurrence of the fictitious agent, and such that there exists a pattern $Z' = b_1, \dots, b_m$ that satisfies both $Z \equiv Z'$ and $Z' \models Z_l$.

The agent permutations required while proving that $Z \equiv Z'$ allow for the derivation of a permutation p such that $a_{p(i)} \equiv b_i$. Then, the restriction ϕ of p to the integers between 1 and n is called and embedding between Z and Z_l, and writes as $Z_l \lhd_\phi Z$.

Remark 4.2.2. Several embeddings may exist between Z_l and Z, for the same Z'. If such is the case, the relative weight of the reaction in the stochastic semantics is influenced. Let $[Z, Z']$ denote the set of embeddings between Z and Z'. The notion of embedding can be extended to patterns with fictitious agents, by defining $Z_l \lhd_\phi Z$ if and only if $(\downarrow_\emptyset Z_l) \lhd_\phi \downarrow_\emptyset Z$, where $\downarrow_\emptyset$ removes all occurrences of the fictitious agent in patterns.

We also note that if E_l is the left-hand-side term of a rule $r = E_l \to E_r @ k$ and Z is a mixture such that $E_l \lhd_\phi Z$, then the embedding ϕ between E_l and Z fully defines the action of rule r on Z, up to structural equivalence. In order to define Kappa's stochastic semantics we note that *computation steps* have to occur over $\equiv$-equivalent classes of mixtures. Consequently, an equivalence relation $\equiv_L$ is defined over triples (r, E, ϕ), with ϕ an embedding of the left-hand-side of r into E:

$$(r_1, \phi_1, E_1) \equiv_L (r_2, \phi_2, E_2) \Leftrightarrow r_1 = r_2 \text{ and } \exists \text{ an embedding } \Psi \in [E_1, E_2] \text{ s.t. } \phi_2 = \Psi \circ \phi_1$$

The last notion needed in this section is that of the weight-labeled transition system associated to a given Kappa system:

Definition 4.2.15 (**WLTS of a Kappa system**)
Let $\mathcal{R} = (\pi_0^{\mathcal{R}}, \{r_1, \ldots, r_n\})$ be a Kappa system. Then, its corresponding WLTS $\mathcal{W}_{\mathcal{R}} = (X, L, w, \pi_0)$ consists of:

(i) *X, the set of all $\equiv$-equivalent classes of mixtures;*

(ii) *L, the set of of all $\equiv_L$-equivalence classes of triples (r, E, ϕ) such that ϕ is an embedding between the left-hand-side of r and E;*

(iii) *$w(x, l, x') = \frac{k}{|[E_l, E_l]|}$, whenever there exists a rule $r = E_l \to E_r @ k$, two mixtures E and E', and an embedding $\phi \in [E_l, E]$, such that $x = [E]_\equiv$, $l = [r, E, \phi]_{\equiv_L}$, and E' is the result (up to $\equiv$) of the application of r along ϕ to the mixture E; otherwise, $w(x, l, x') = 0$;*

(iv) $\pi_0(x) = \sum_{E' \in Dom(\pi_0^{\mathcal{R}}) \cap x} \pi_0^{\mathcal{R}}(E')$.

It is assumed that the system state is an *"agent soup"*, *i.e.* the order of agents in the mixture is irrelevant, or otherwise said the states of the system are the class of $\equiv$-mixture. With all this in place, the stochastic semantics of a Kappa system $\mathcal{R}$ is defined as the trace distribution semantics of its weighted-labeled transition system $\mathcal{W}_{\mathcal{R}}$.

Part II

Model reduction

Motivation

As mentioned in the introduction of this thesis, one of the main challenges when modelling biochemical reaction networks is related to the complexity of the resulting models. In particular, the number of possible chemical species of a model is often subject to combinatorial explosion, due to the large number of species that may arise as a result of protein bindings and post-translational modifications [94]. As a consequence, for example, mechanistic models of signaling pathways easily become very combinatorial.

More generally, when considering deterministic models of BRNs, this species' numbers' combinatorial explosion caused by their rich pattern of interactions results in high-dimensional, non-linear systems of ODEs describing the evolution of said species' concentrations. The analysis of such systems of equations is often computationally expensive and even prohibitive in practice. What's more, as we have previously shown, deterministic models generally offer an *approximation* of their stochastic counter-part [111]. Moreover, the differential equations themselves do not transparently reflect the underlying mechanisms. Addressing these latter issues, formalisms allowing to write mechanistic hypothesis in form of discrete transition steps have been proposed: Boolean networks[180], logical networks[185], Petri Nets[38], cellular automata[83], and rule-based languages[49], to name the most common. Languages such as Kappa[49, 53] and BNGL[18] provide compact ways of describing models prone to combinatorial explosion, of simulating them [50], and even transforming them into ODEs [18]. However, the curse of dimensionality once again rises when trying to compute the system behavior.

A strategy to cope with such complexity is model reduction, in which certain properties of biochemical models are exploited in order to obtain simpler versions of the original complex model; these simpler models should preserve the important behavioral aspects of the initial system. An example of such a property is the *multiscaleness* of biochemical networks, with respect to both time-scales and species' abundance. In the case of the former, it is known that biochemical processes governing network dynamics span over many well separated timescales: while protein complex formation occurs on the seconds scale, post-translational protein modification takes minutes, and changing gene expression can take hours, or even days. As for the latter, multiscaleness also applies to the abundance of various species in biochemical networks: the DNA molecule has one to a few copies, while mRNA copy numbers can vary from a few to tens of thousands. On the one hand, these widely different time- and concentration scales represent challenges for the estimation of rate constants, for the measurement of low-concentration species, and even for numerical integration. On the other hand, they represent a feature that can be exploited for model reduction purposes, allowing to approximate the complete mech-

anistic description with simpler rate expressions, that retain the essential features of the full problem on the time scale or in the concentration range of interest.

The Michaelis-Menten (MM) enzymatic model illustrates the idea of *dominance* can be useful for model reduction of nonlinear models with multiple timescales.

As previously described in Example 1.2.1, the MM model consists of an enzyme that reversibly binds a substrate to form a complex, which in turn releases a product, while preserving the enzyme, $\mathrm{E} + \mathrm{S} \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} \mathrm{E:S} \xrightarrow{k_2} \mathrm{E} + \mathrm{P}$.

Its ODE system writes as:

$$\begin{cases} \frac{d[S]}{dt} = k_{-1}[E:S] - k_1[E][S] \\ \frac{d[E]}{dt} = k_{-1}[E:S] + k_2[E:S] - k_1[E][S] \\ \frac{d[E:S]}{dt} = k_1[E][S] - k_{-1}[E:S] - k_2[E:S] \\ \frac{d[P]}{dt} = k_2[E:S] \end{cases} \tag{4.3}$$

The Michaelis-Menten equation relates the rate of product formation to the substrate concentration:

$$v = \frac{d[P]}{dt} = k_2 \frac{[E]_T[S]}{K_M + [S]}, \tag{4.4}$$

where $[E]_T = [E]+[E:S]$ is the total enzyme concentration, and $K_M = \frac{k_2+k_{-1}}{k_1}$ is the Michaelis-Menten constant. Equation (4.4) can be interpreted as the reaction rate of a reduced reactive system, equivalent to the original reaction system (1.2.1), in which the intermediary complex $[E:S]$ has been eliminated:

$$\mathrm{S} \xrightarrow{k_2 \frac{[E]_T}{K_M+[S]}} \mathrm{P} \tag{4.5}$$

The approximation of the original enzymatic reaction scheme (1.2.1) by (4.4) is generally considered to be sufficiently good if the *quasi steady-state* (QSS) assumption holds, that is if the total initial enzyme concentration is much lower than the total initial concentration of substrate: $[S]_0 \gg E_T$, with $E_T = [E]_0 + [E:S]_0$. In this case, the complex $[E:S]$ is a low concentration fast species, whose concentration is dominated by that of the substrate; the value of $[E:S]$ almost instantly relaxes to a value determined by $[S]$. Thus, one can set $\frac{d[E:S]}{dt} = 0$, and exploit this relation to pool the two reactions of the initial system (4.3) into an unique irreversible reaction (4.5).

The original MM analysis used the complementary *quasi equilibrium* (QE) approximation, which considers the complex formation reaction to be *fast* and *reversible*: $k_{-1} \gg k_2$. Thus, the term $k_{-1}[E:S]$ can be considered to *dominate* the term $k_2[E:S]$ in Eq.(4.3), meaning the latter can be discarded from the ODE system, allowing for pooling of species, and resulting once again in a single step approximation that reads:

$$\mathrm{S} \xrightarrow{k_2 \frac{[E]_T}{K_d+[S]}} P, \tag{4.6}$$

with $K_d = \frac{k_{-1}}{k_1}$, if indeed $[S] \gg E_T$. We note that if the QE assumption is indeed valid, then $K_M \approx K_d$.

The authors of [168] provide a similar validity criterion of the deterministic Michaelis–Menten approximation: $[S]_0 + K_M \gg E_T$. Whenever this condition holds, it is shown that a separation exists between a fast "pre-steady state" timescale and a slower "steady state" timescale. The solution of the Michaelis-Menten approximation closely tracks the solution of the initial enzymatic reaction system on the slow timescale.

This simple example illustrates how the dynamics of multiscale, large biochemical systems can be reduced to those of simpler models, called *dominant* subsystems [138], which contain less parameters and are easier to analyze. Dominant subsystems are chosen by comparing the timescales of the large system. For example, the classical quasi steady-state (QSS) [23] and quasi-equilibrium (QE) approximations [127, 106] are conditions that lead to dominance, and represent popular methods for the computation of "first approximations" to the slow invariant manifold. Classical QSS is based on the small concentrations of highly reactive intermediate species (*i.e.*, atoms, ions, enzymes and substrate-enzyme complexes)[41], while in the QE approximation the reduction of the full mechanism is done based on the existence of *fast* and *slow* reactions.

The multiscaleness property of biochemical network is by definition closely linked to the mathematical notion of dominance, captured in the framework of tropical analysis[9, 120]. Recently, a class of semi-formal methods for reducing and hybridizing models of biochemical networks has been developed, based on ideas from tropical analysis [138, 139, 154, 155]. These methods exploit the multiscaleness of biochemical networks, in order to deduce *dominance* relations among parameters and/or reaction rates, which can then be used to obtain a system of truncated ODEs (by eliminating the dominated terms). One of the advantages of using dominance relations in multi-scale networks is that it helps cope with parameter uncertainty: parameter values are replaced with their orders of magnitude, which are easier to determine. However, providing guarantees as to how the solution of the reduced model relates to the original one, while avoiding to solve the latter, remains a challenge.

Consequently, the works presented in this first part of the manuscript deal with model reduction techniques that exploit the multiscaleness property of biochemical reaction networks.

In Chapter 5, we investigate how the multiscaleness property can be exploited in the context of *rule-based models* of genetic circuits. We argue that, while typical primitives for modelling such circuits are gene interaction networks, they fail to capture low-level mechanistic details of genetic interactions, which are necessary when designing *in vivo* experiments.

Rule-based modeling languages such as Kappa allow, by their nature, to unambiguously specify mechanistic details such as DNA binding sites, dimerisation of transcription factors, or co-operative interactions. Nonetheless, such a detailed description comes with complexity and computationally costly executions. In order to tackle this issue, we propose a reduction method for rule-based programs, based on equilibrium approximations. Our algorithm is an adaptation of an existing algorithm, which was designed for reducing reaction-based programs; our version of the algorithm scans the rule-based Kappa model (in which each rule operates exclusively on site-graphs whose interfaces are fully specified, *i.e.,* they are species), in search of interaction patterns which are amenable to equilibrium approximations, such as the Michaelis-Menten scheme. Additional checks are then performed, as to verify if the reduction is meaningful in the

context of the full model. The reduced model is efficiently obtained by static inspection over the rule-set. Our tool is tested on a detailed rule-based model of a λ-phage switch, which lists 92 rules and 13 agents. The reduced model has 11 rules and 5 agents, and provides a dramatic (several orders of magnitude) reduction in simulation time.

Our time-scale separation technique is justified by asymptotic convergence results; however, for any concrete parameter values, one cannot obtain any information on the accuracy of the reduced model's trajectories, without comparing them to those obtained by executing the original model.

This is why, in Chapter 6, we design and test an approximation method for ODE models of biochemical reaction systems, in which the guarantees are our major requirement. Borrowing from tropical analysis techniques, we argue that the deterministic model of a multiscale BRN can be simplified by exploiting the dominance relations among terms of each of its equations, in order to determine the dominant subsystems that govern the evolution of each species' concentration. As the dominant terms can change during a dynamic execution of the system, depending on which species dominate the others, several possible modes exist. Thus, simpler models consisting of only the dominant subsystems can be assembled into hybrid, piecewise smooth models, which approximate the behavior of the initial system. By combining the detection of dominated terms with symbolic bounds propagation, we show how to approximate the original model by an assembly of simpler models, consisting in ordinary differential equations that provide time-dependent lower and upper bounds for the concentrations of the initial model's species.

The utility of our method is twofold. On the one hand, by the nature of its design, it provides sound interval bounds for each species' concentration, and can hence serve to evaluate the accuracy of tropicalization reduction heuristics for ODE models of biochemical reduction systems.

On the other hand, our method also provides a reduction heuristics that performs without any prior knowledge of the initial system's behavior (*i.e.*, no simulation of the initial system is needed in order to reduce it).

Chapter 5
Prototyping genetic circuits using rule-based models

This chapter is based on the work published in the proceedings of the 4th Hybrid Systems Biology (HSB) workshop [13].

5.1 Motivation

As mentioned in the introductory section of this thesis, one of the goals of synthetic and systems biology consists in the design and control of genetic circuits in an analogous way to how electronic circuits are manipulated in human made computer systems. In this sense, one notes previous successes in engineering simple genetic circuits that are encoded in DNA, and which perform their function in the cellular environment [71], [87]. However, there remains a need for rigorous quantitative characterization of such small circuits and their mutual compatibility [113], and an important ingredient towards such a characterization is having an appropriate modeling language: one that is able to capture model requirements, to prototype circuits, and to predict their quantitative behavior before committing to time-intensive *in vivo* experiments.

As biomolecular systems are usually highly dimensional, stochastic, non-linear dynamical systems, their quantitative modeling is particularly challenging. Consequently, a common modeling practice consists in applying ad-hoc simplifications that neglect the mechanistic details of the underlying biological processes, but which allow to predict the system's behaviour as a function of time.

For example, the fact that *"protein A activates protein P"* is often modeled immediately in terms of a reaction $A \rightarrow A + P$ with the Hill kinetic coefficient (e.g. $\frac{k[A]^n}{1+k[A]^n}$) - however, the actual biological protein activation mechanism includes the formation of a macromolecular complex and its binding to a molecular target.

While models that employ such simplifications are easier to execute, this type of simplification makes models hard to edit or refine. For example, the fact that several mechanistic steps are merged into a single kinetic rate means that a new experimental insight about an interaction mechanism cannot be easily integrated into the model. Moreover, such abstract models do not provide guidelines that are precise enough for circuit synthesis, and in some cases, only the more detailed models are able to explain certain behaviours (e.g., in [57], it is shown that only

when incorporating the mRNA, the model explains certain experimentally observed facts).

As detailed in Section 4.2, rule-based languages, such as Kappa [149] or BioNetGen [18], are designed to naturally capture the protein-centric and concurrent nature of biochemical signalling: in a rule-based model, the internal protein structure is maintained in form of a *site-graph*, and interactions can occur by simply testing *patterns*, *i.e.*, local contexts of molecular species. We have also shown in Section 4.2 that the executions of rule-based models are traces of a continuous-time Markov chain (CTMC), defined according to the principles of chemical kinetics.

In general, rule-based models can be considered as an improvement of classical biochemical reaction models for two major reasons. First, the explicit graphical representation of molecular complexes makes models easy to read, write, edit or compose (by simply merging two collections of rules).

For example, the dimerization reaction between two CI molecules of the λ-phage [151] model is classically written as $\mathrm{CI} + \mathrm{CI} \rightarrow \mathrm{CI}_2$, where the convention is that CI represents a free monomer, and CI_2 represents a free dimer. On the other hand, the same reaction written in Kappa amounts to:

```
'CI2' CI(ci[.],or[.]), CI(ci[.],or[.]) <-> CI(ci[1],or[.]), CI(ci[1],or[.]) @ 'k2+', 'k2-'
```

where *ci* and *or* denote the binding sites of the protein CI, and $\mathrm{CI}(\mathrm{ci}[1], \mathrm{or})$ indicates that the identifier of the rule-based bond on the site *ci*, which accounts for the physical interaction between the two CI monomers, is 1. The left-handside of the rule states that dimerization can occur only between two CI monomers that are free (unbound) on both the *cri* and the *or* sites.

Secondly, a rule set can be both *executed*, and subjected to formal *static analysis*: for example, it provides efficient simulations [50], [96], but also automated answers about the reachability of a particular molecular complex [51], or about causal relations between rule executions [49].

In this sense, in Section 5.2, we illustrate the advantages of using rule-based modeling languages to prototype genetic circuits, by building a detailed Kappa model of the λ-phage genetic switch. When compared to traditional ODE modeling, and even to reaction-based modeling, using the chemically intuitive syntax/notations of Kappa results in a model that captures the protein-centric mechanistic details of the underlying biochemical processes in a way that can be easily understood and used by the biologist modeler, and which moreover is concise, transparent, and easily extendable.

However, a downside to incorporating too many mechanistic details in the model is that they lead to computationally costly execution. For this reason, in Section 5.3, we propose and implement an efficient method for automatically detecting and applying equilibrium approximations to the Kappa model. We note that in the specific Kappa models we build in this chapter, the kinetic rate constants of the chemical interactions are highly dependent on the *full* interface of the reactant patterns. The consequence of this parameter sensitivity is that reactions operating on similar left-handside patterns cannot be subsumed into a single rule. Hence, we assume that the left-handside of each rule is composed of patterns whose interface

is fully specified, *i.e.*, each pattern is a *species*, and as such our algorithm provides a sound reduction of models in which each rule represents a reaction.

These approximations allow one to obtain a smaller model, in which some agents are eliminated, and the kinetic rates are appropriately adjusted. In this way, the experimentalist can choose to obtain predictions more efficiently but less accurately, however without losing track of the underlying low-level mechanisms.

In related works [135] and [112], the authors propose an algorithm for reducing a reaction-based model, by searching for interaction schemes amenable to equilibrium approximations. In this paper, we adapt this algorithm to rule-based models of genetic circuits. We implement the algorithm in OCaml, and test it on the detailed rule-based model of the λ-phage switch ([151], [152]) we construct in Section 5.2.

The results of this approximation are presented in Section 5.4.1: while the complete chemical genetic circuit model contains 92 rules, 13 agents, and 61 species, the reduced model contains only 11 rules and 5 agents, and is able to approximate the behavior of the original system. We conclude this chapter by discussing the limitations of our reduction method, as well as future work.

Related work. The principle of obtaining conclusions about system's dynamics by analysing their model description, originates from, and is exhaustively studied in the field of formal program verification and model checking [45], [25], while it is recently gaining recognition in the context of programs used for modeling biochemical networks. An example is the related work of detecting fragments for reducing the deterministic or stochastic rule-based models [62], [69], [63], detecting the information flow for ODE models of biochemical signaling [91], [20], or the reaction network theory [46].

5.2 A Kappa model of the λ-phage decision circuit

We start by building a Kappa model of the λ-phage decision circuit, using the reaction-based model presented in ([135], [112]). The λ-phage circuit is one of the most studied genetic circuits in synthetic and systems biology. Over the years, much has been learned about this simple genetic circuit, but even more than half a century after its discovery, new mechanisms are being discovered about its functioning.

The phage λ ([151],[152]) is a virus that infects *E.coli* cells, and replicates using one of two strategies: *lysis* or *lysogeny*. In the *lysis* strategy, phage λ uses the machinery of the *E.coli* cell to replicate itself and then lyses the cell wall, killing the cell and allowing the newly formed viruses to escape and infect other cells. On the other hand, in the *lysogeny* scenario, the virus inserts its DNA into the host cell's DNA and replicates through normal cell division, remaining in a latent state in the host cell (it can always revert to the lysis strategy). This switch mechanism is illustrated in Figure 5.1.

The decision between *lysis* and *lysogeny* is known to be influenced by environmental parameters, as well as the multiplicity of infection and variations in the average phage input [8].

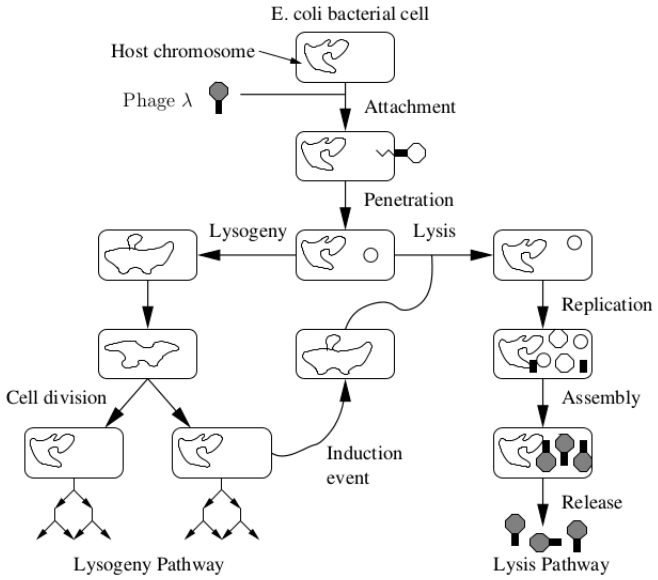
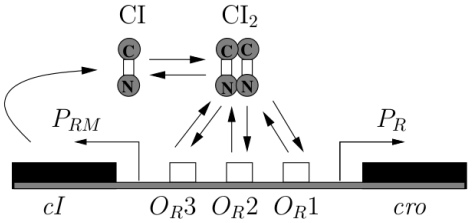


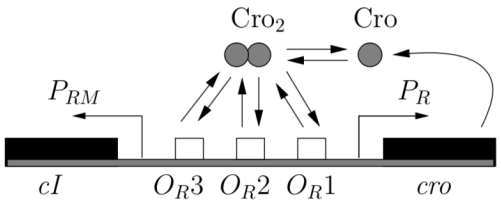
Figure 5.1: Phage λ developmental model (Figure taken from [135])

The key element controlling this decision process is the O_R operator (shown in Figure (5.2)), which is composed of three operator sites (O_{R1}, O_{R2}, O_{R3}) to which transcription factors can bind, in order to activate or repress the two promoters (P_{RM} and P_R) overlapping the operator sites. When RNA polymerase ($RNAP$) binds to P_{RM}, it initiates transcription to the left, to produce mRNA transcripts from the cI gene; RNAP bound to the P_R promoter, on the other hand, initiates transcription to the right, producing transcripts from the *cro* gene. The two promoters form a genetic switch, since transcripts can only be produced in one direction at a time. This production mechanism is illustrated in Figure 5.2.

The cI gene codes for the CI protein, also known as the λ *repressor*: in its dimer form, it is attracted the the O_R operator sites in the phage's DNA, repressing the P_R promoter from



(a) CI monomers are produced from the cI gene. Two CI monomers can bind together, and form a dimer that can in turn bind to one of the O_R operator sites.



(b) Cro monomers are produced from the *cro* gene. Two Cro mononers can form a dimer, which can bind to one of the O_R operator sites.

Figure 5.2: λ-phage switch: production of CI and Cro proteins. (Figure taken from [135])

which Cro production is initiated, and further activating CI production. Similarly, the *cro* gene codes for the Cro protein, which also dimerizes in order to bind to the O_R operator sites and prevent production from P_{RM} and its own production.

While CI_2 and Cro_2 can bind to any of the three operator sites at any time, they have a different affinity to each site. The CI_2 has its strongest affinity to the O_{R1} operator site, next to the O_{R2} site, and finally to the O_{R3} site (in other words, CI_2 first turns off P_R , then activates P_{RM} , and finally, represses its own production), while Cro_2 has the reverse affinity (it first turns off CI production, then turns off its own production).

The effects of CI and Cro dimers binding to the operator sites are illustrated in Figure 5.3.

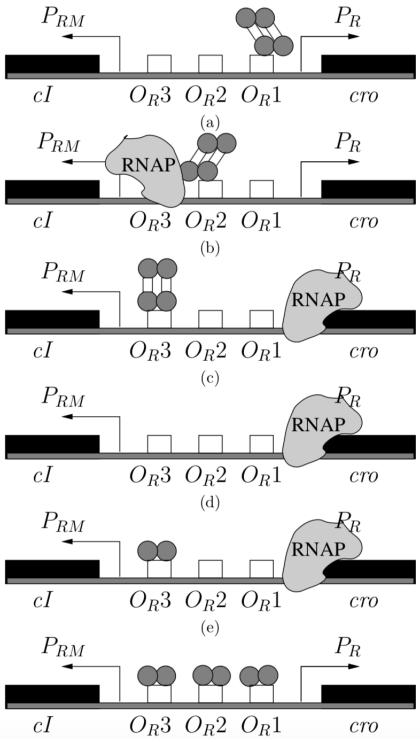


Figure 5.3: The effects of CI and Cro dimers binding to the operator sites. *(Top row)* A CI dimer bound to the site O_{R1} turns off the P_R promoter. *(Second row)* A CI dimer bound to the site O_{R2} activates the P_{RM} promoter. *(Third row)* A CI dimer bound to the site O_{R3} turns off the P_{RM} promoter completely, while P_R is activated. *(Fourth row)* When all operator sites are free, promoter P_R is active. *(Fifth row)* A Cro dimer bound to O_{R3} turns off P_{RM} completely. *(Bottom row)* An additional binding of Cro dimers to O_{R1} and O_{R2} turns P_R off completely. (Figure taken from [135])

Remark 5.2.1. The specificities of the interactions described above, as illustrated in Figure 5.3, imply that the decision to execute a specific reaction in the model is taken by considering the *full* interface of the operator pattern O_R: consider for example how in the fourth panel, P_R is activated when *all sites* of O_R are free. Similarly, in the last panel, P_R is completely repressed when *all sites* of O_R are bound to a Cro dimer.

This particularity of the λ-phage mechanism translates in the model by a high sensitivity of a reaction's kinetic constant on its reactants' *full* interface, *e.g.*, each possible configuration of the operator has a different binding affinity to the same given protein. The dependency of specific molecular interactions on the full reactant pattern interface thus prevents the aggregation of different reactions describing similar mechanisms operating on the same reactant species (*e.g.*, the same protein binding some site of the same operator) into a single rule.

This restriction is perpetuated throughout the λ-phage model, which means that the major advantage of rule-based modeling w.r.t. reaction based models (*"a rule subsumes many possible reactions"*) could remain unexploited when constructing models of genetic circuits. This feature motivates the "each-rule-is-a-reaction" constraint we will assume for our reduction algorithm: we assume that a rule-based model is such that all left-handside and right-handside represent fully specified site-graphs (*i.e.*, species), instead of "don't care, don't write patterns".

However, we argue that even under the above-mentioned restriction, rule-based models are better suited for prototyping genetic circuits, when compared to reaction-based models. Indeed, the former benefit from an intuitive chemical syntax that the latter lack, *i.e.*, they enable the explicit representation of protein-protein interactions, which in itself is a strong enough reason to prefer rule-based models to reaction-based models, when prototyping genetic circuits. All in all, we stress that in this chapter, we choose to employ rule-based models due to the *chemical expressivity of their syntax*, instead of their ability of avoiding combinatorially large model representations.

The feedback through the binding of the products as transcription factors coupled with the affinities described makes the O_R operator behave as a genetic bistable switch. In one state, the production of the Cro protein inhibits the production of CI. In this state, the cell follows the *lysis* pathway, since genes downstream of Cro produce the proteins necessary to construct new viruses and lyse the cell. In the other state, the production of CI prevents production of Cro, and the cell follows the *lysogeny* pathway, since proteins necessary to produce new viruses are not produced. Instead, the cell proceeds with the production of proteins that are used to insert the DNA of the phage into the host cell.

We note that in the lysogeny state, the cell develops an immunity to further infection: the *cro* genes found on the DNA inserted by further infections of the virus are also shut off by CI_2 molecules that are produced by the first virus to commit to lysogeny. Once a cell commits to lysogeny, it becomes very stable and does not easily change over to the lysis pathway. An induction event is necessary to cause the transition from lysogeny to lysis. For example, lysogens (i.e., cells with phage DNA integrated within their own DNA) that are exposed to UV light end up following the lysis pathway.

5.2.1 Example: Kappa model of CI and CII production

To exemplify the advantages of rule-based models, consider the following sub-network of the λ-phage circuit, which describes a simplified mechanism for the production of CI from P_{RE}, and for the production of CII from P_R.

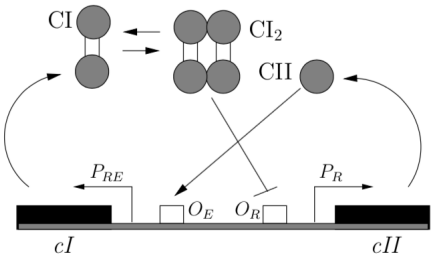


Figure 5.4: CI and CII production from P_R and P_{RE}.(Figure taken from [135])

CII production is initiated from the P_R promoter. Initially, CI production from promoter P_{RE} proceeds at a low basal rate. As CII builds up, it binds to the O_E operator site, activating the production of CI molecules from P_{RE}. The CI molecules dimerize and bind to the O_R operator sites, further repressing production of CII. Over time CII degrades, reducing the production rate of CI. Finally, after CI degrades, the system returns to the initial state

For readability purposes, assume for the time being that we want to create a model for the network of Figure 5.4, in which we abstract away the details regarding CI and CII binding to the operators O_E and O_R, and focus solely on the regulatory effects of the interaction between the proteins CI and CII and the promoters P_R and P_{RE}. More specifically, we want to create a model in which *protein CII activates CI production through the promoter P_{RE}, and protein CI represses CII production through the promoter P_R.*

We create a Kappa model for this regulatory circuit, by assuming that proteins CI and CII can bind directly to their respective promoters P_R and P_{RE}. Among the different ways of modeling genetic regulatory networks, we use a systematic procedure that produces a fairly reasonable model, and which can represent a guideline for prototyping genetic circuits in Kappa:

- create agents for RNAP (the RNA-polymerase), as well as for every protein and promoter/operator of the network under consideration;
- model the open complex formation for a promoter by a pair of rules: a reversible rule, in which the RNAP binds to the promoter, and an irreversible rule in which the resulting RNAP-promoter complex acts as a modifier for the production of the protein the promoter encodes for;
- model the repression of a promoter P by a repressor R through a reversible rule, in which R binds to P, preventing the subsequent binding of P's activators;
- model the activation of a promoter P by an activator A by a pair of rules: a reversible rule, in which A and $RNAP$ both bind to P, followed by an irreversible reaction in which

the $A - RNAP - P$ complex acts as a modifier for the production of P's protein;

- add rule for modeling the remaining chemical behavior: protein degradation, dimerisation, *etc...*

Using these guidelines, the resulting Kappa model for the mechanism of Figure 5.4 writes as:

(i) degradation of CI and CII proteins, at the same rate kd:

```
'CIdegr'    CI(ci[.],or[.]) -> . @ 'kd'
'CIIdegr'   CII(pre[.])     -> . @ 'kd'
```

(ii) basal-rate production of CI and CII:

```
'PRE.RNAP' PRE(cii[.],rnap[.]), RNAP(pr[.],pre[.])   <-> PRE(cii[.],rnap[1]), RNAP(pr[.],pre[1])           @ 'kpre+', 'kpre-'
'CIIprod'  PRE(cii[.],rnap[1]), RNAP(pr[.],pre[1]),.  -> PRE(cii[.],rnap[1]), RNAP(pr[.],pre[1]), CII(pre) @ 'ko'

'PR.RNAP'   PR(ci[.],rnap[.]), RNAP(pr[.],pre[.])   <-> PRE(ci[.],rnap[1]), RNAP(pr[1],pre[.])            @ 'kpr+', 'kpr-'
'CIIprodb'  PR(ci[.],rnap[1]), RNAP(pr[1],pre[.]),.  -> PRE(ci[.],rnap[1]), RNAP(pr[1],pre[.]), CI(ci,or) @ 'kb'
```

(iii) CI dimerization:

```
'CI2' CI(ci[.],or[.]), CI(ci[.],or[.]) <-> CI(ci[1],or[.]), CI(ci[1],or[.]) @ 'k2+', 'k2-'
```

(iv) CI dimers repress the activity of promoter P_R, by binding to it:

```
'PR.CI2' PR(ci[.],rnap[.]), CI(ci[1],or[.]), CI(ci[1],or[.]) <->
         PR(ci[2],rnap[3]), CI(ci[1],or[2]), CI(ci[1],or[3]) @ 'kr+', 'kr-'
```

(v) CII monomers activate the promoter P_{RE}, by binding to it:

```
'PRE.CII.RNAP' PRE(cii[.],rnap[.]), CII(pre[.]), RNAP(pr[.],pre[.])
                 <-> PRE(cii[1],rnap[2]), CII(pre[1]), RNAP(pr[.],pre[2])          @ 'ka+','ka-'

'CIproda'      PRE(cii[1],rnap[2]), CII(pre[1]), RNAP(pr[.],pre[2]), .
                 -> PRE(cii[1],rnap[2]), CII(pre[1]), RNAP(pr[.],pre[2]), CI(ci[.],or[.]) @ 'ka'
```

One notes that, when compared to the equivalent reaction-based model, the Kappa model represents a significant improvement: no supplementary names need to be introduced for the complex species, which are instead denoted in an intuitive chemical style, that is indicative of the underlying biological mechanisms. The result is a model that is more concise, more transparent, easily readable, and easily extendible. The CI-CII Kappa model represents just an example of the ease with which the Kappa language can be used to model biological interactions not only on a protein-protein level, but also at the genetic level - a feature that results in detailed

stochastic models that naturally account for transcriptional and translational resource usage [184]. What's more, the existing visual representations of Kappa models mirror the intuitively modular nature of the modeling processes involved.

We also implement the full λ-phage circuit model, in which we include interactions with the O_R and O_E operators, as well as the remaining reactions of the λ-phage circuit, the detailed description of the which can be found in [135].

The full Kappa model contains 96 rules, 16 proteins and 61 species. This model will serve as a case study for our reduction algorithm. Using available tools [149], we generate the model's contact map, which we present in Figure 5.5.

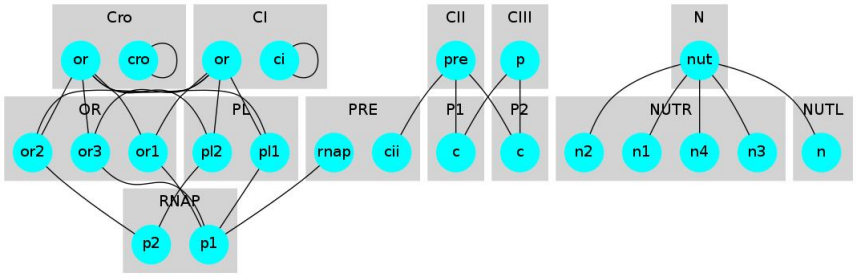


Figure 5.5: The contact map of the full λ-phage model. The model consists of 96 rules, 16 proteins and 61 species.

5.3 Model Approximation using Michaelis-Menten-like reaction schemes

In this section, we present the approximation algorithm, which comprises three reduction schemes: competitive enzymatic reduction, operator-site reduction, and fast dimerization reduction.

As all three schemes are based on the Michaelis-Menten (MM) enzymatic model, we start by providing some mathematical background for justifying the validity of the MM reduction scheme in the case of stochastic models.

We continue by detailing the three reductions, and conclude by presenting the top-level algorithm that employs them.

5.3.1 Validity of the Michaelis-Menten enzymatic reduction in stochastic models

As we will see in the next sections, our reduction algorithm will scan a Kappa model in search for reaction patterns similar to the original Michaelis-Menten enzymatic mechanism ($E + S \leftrightarrow E : S \rightarrow P$), and then reduce them using MM-like approximations. We have previously seen that this approximation is generally considered to be sufficiently good under

different assumptions, such as, for example, that the rate of dissociation of the complex to the substrate is much faster than its dissociation to the product (i.e. $k_1 \gg k_2$), (the *equilibrium approximation*). Even if the equilibrium condition is not satisfied, it can be compensated in a situation where the total amount of substrates is significantly larger than the enzyme quantity: $[S]_0 + K_m \gg E_T$ (quasi-steady state assumption, or *QSSA*).

The informal terminology of being "significantly faster", motivated the rigorous study of the limitations of the approximations based on separating time scales. The enzymatic (Michaelis-Menten) approximation has been first introduced and subsequently studied in the context of deterministic models (e.g. [134]), and it had not been given much consideration in the context of stochastic models, until relatively recently. However, this has changed, due to the computational demands of the stochastic simulation algorithm (SSA), which not only requires a potentially large number of runs in order reasonably [1] estimate the system behavior, but more importantly, also requires every single reaction event to be simulated one at a time. In this sense, the QSSA approximation not only reduces the dimensionality of the system, but it can also substantially reduce simulation time, by removing fast reactions such as complex formation and complex dissociation, which are commonly found in enzymatic models. Indeed, the propensities of such very fast reactions can be significantly larger than those of other reactions, and thus tend to dominate (and slow down) stochastic simulations. As a result, in order to facilitate more efficient temporal behavior analysis, extended MM approximations, and more generally time-scale separation techniques, have recently started to be investigated in the stochastic context ([157], [90], [47], [93], [164], [81]).

Moreover, mathematical justifications for the application of the QSSA within the stochastic chemical kinetic framework have been investigated to establish a theoretical basis to illustrate how the QSSA can be applied to stochastic simulation algorithms.

Notably, the following result from [54] (also shown as a special case of the multi scale stochastic analysis from [103]), shows that, under an appropriate scaling of species' abundance and reaction rates, the original model and the approximate model converge to the same process.

Theorem 5.3.1 (*(Darden [54], Kang [103])*)
Consider the usual MM reaction network:

$$\mathrm{E} + \mathrm{S} \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} \mathrm{E:S} \xrightarrow{k_2} \mathrm{E} + \mathrm{P}, \tag{5.1}$$

and let $X_S(t)$, $X_E(t)$, $X_{E:S}(t)$ and $X_P(t)$ denote the respective species' copy numbers, due to the random-time change model (3.30). Write $E_T = X_{E:S}(t) + X_E(t)$ to denote the total enzyme quantity, and let $V_E(t) = \int_0^t N^{-1} X_E(s) ds$.

Assuming that the amount of substrate is much larger than that of enzyme, the system volume can be expressed as $N = \mathcal{O}(X_S)$.

Finally, let $\gamma_{-1}, \gamma_1, \gamma_2$ be parameters of order 1.

Then, under the scaling $k_{-1} \to \gamma_{-1}$, $k_1 \to N\gamma_1$, $k_2 \to N\gamma_2$, $N \to \infty$, and $\frac{X_S(0)}{N} \to x_S(0)$, one has that:

[1] *i.e.*, at a reasonable degree of statistical confidence

$$\left(\frac{X_S(t)}{N}, V_E(t)\right) \longrightarrow (x_S(t), v_E(t))\,,$$

with:

$$\begin{cases} \frac{d}{dt}v_E(t) = \frac{E_T}{1+\hat{K}x_E(t)} \\ \\ \frac{d}{dt}x_S(t) = -\frac{E_T\gamma_2\hat{K}x_S(t)}{1+\hat{K}x_S(t)} \end{cases}$$

and $\hat{K} = \frac{\gamma_{-1}}{\gamma_1+\gamma_2}$.

The assumptions listed in the theorem capture the following: (i) X_S and X_P are scaled to concentrations, while X_E and $X_{E:S}$ remain in copy numbers; (ii) the stochastic reaction rate k_{-1} is an order of magnitude smaller than the rates k_1 and k_2.

A complete proof is provided in [103]. Herein, we outline the general idea.

Let $N > 0$ be a natural number, and let $Z_S(t) = X_S(t)/N$, $Z_E(t) = X_E(t)$, $Z_{S:E}(t) = X_{S:E}(t)$, $Z_P(t) = X_P(t)/N$. Writing out the scaled random time-change model for the substrate gives:

$$\begin{aligned} Z_S(t) = Z_S(0) &- N^{-1}\xi_1(N\int_0^t \gamma_{-1}Z_S(s)Z_E(s)ds) \\ &+ N^{-1}\xi_2(N\int_0^t \gamma_1 Z_{S:E}(s)ds), \end{aligned}$$

Similarly, the scaled random time-change model for the complex $E:S$ writes as:

$$\begin{aligned} Z_{E:S}(t) = Z_{E:S}(0) &+ \xi_1(N\int_0^t \gamma_{-1}Z_S(s)Z_E(s)ds) \\ &- \xi_2(N\int_0^t \gamma_1 Z_{S:E}(s)ds) \\ &- \xi_3(N\int_0^t \gamma_2 Z_{S:E}(s)ds). \end{aligned}$$

After dividing the latter equation with N, and applying the law of large numbers, we obtain the balance equations analogous to assuming that the complex is at equilibrium. This equation implies the expression for $\frac{d}{dt}v_E(t)$.

The equation for $\frac{d}{dt}x_S(t)$ follows from the model of $Z_S(t)$: we first use the conservation law $Z_{S:E}(s) = N^{-1}E_T - Z_E(t)$ and then substitute the obtained value of $\frac{d}{dt}v_E(s)$.

In order to confirm that the reduction is appropriate, our goal is now to show that the scaled versions of the original enzymatic model (5.1) and the reduced model (4.5) are equivalent in the limit when $N \to \infty$.

Let $Z_P(t) := N^{-1}X_P(t)$ be the scaled random time change for the product in the original model, and let $\hat{Z}_P(t) := N^{-1}\hat{X}_P(t)$, in the reduced model.

Notice that, from the balance equations, $\frac{x_P}{dt} = -\frac{x_S}{dt}$.

According to the reduced Michaelis-Meten system, the random time change for the product is given by

$$\begin{aligned}\hat{Z}_P(t) &= \hat{Z}_P(0) + N^{-1}\xi(\int_0^t \frac{k_2 E_T K}{1 + KN\hat{Z}_S(s)} N\hat{Z}_S(s)ds) \\ &= \hat{Z}_P(0) + N^{-1}\xi(\int_0^t N\frac{\gamma_2 E_T \hat{K}}{1 + \hat{K}\hat{Z}_S(s)} \hat{Z}_S(s)ds).\end{aligned}$$

Passing to the limit, we obtain the desired relation: $\frac{\hat{z_P}(t)}{dt} = \frac{z_P(t)}{dt}$.

We note that Theorem (5.3.1) should not be interpreted as providing the means of computing the approximation error, or even as an algorithm which suggests which difference in time-scales is good enough for an approximation to perform well.

Rather, this result shows that the enzymatic approximation is justified in the limit when the assumptions about the reaction rates and species' abundance are met. In other words, when $N \to \infty$, the scaled versions of the original and reduced models – e.g. $Z_P(t) = N^{-1}X_P(t)$ and $\hat{Z}_P = N^{-1}\hat{X}_P$ – both converge to at the same, well-behaved process. This provides confidence that the actual process $\hat{X}_P$ is a good approximation of the process X_P.

The observation that the enzymatic approximation is justified *in the limit* is in line with recent works that study the robustness of classical enzyme kinetics in the context of cellular biochemistry, *i.e.* in the *in vivo* context of small intracellular environments, where the large volume, large species abundance, and negligible fluctuations assumptions do not hold. For example, in [85], the author shows that for a microscopic stochastic model of enzyme kinetics in a small subcellular compartment, the intrinsic noise induces a breakdown of the Michaelis-Menten equation, even if steady-state metabolic conditions are enforced, and that the deterministic rate equations can severely under-estimate steady-state intracellular substrate concentrations. Moreover, a formula is given for quantifying the deviations from the MM equations - the deviations are substantial for *small* values of K_M; in this case, the bottleneck of the catalytic process (which resides in the decay of a complex, rather than the enzyme-substrate combination) leads to correlations between successive binding events. The limitations of the stochastic quasi-steady-state approximation are also studied in [176], where a simple formula for the relative error between the predictions of the two CMEs (original and reduced) for the simplest biochemical circuit embedding the MM mechanism (*i.e.*, the MM reaction plus a substrate input reaction) is obtained. This formula predicts that the reduced approach can overestimate the variance of the substrate fluctuations by as much as 30%, when the enzyme is half saturated with substrate, *i.e.*, when $x_S = K_M$. However, for substrate concentrations much smaller or larger than K_M, this error becomes negligible. Other works from the same research group deal with corrections to the classical enzyme kinetics models: in [175], the authors compute the finite-volume/finite copy number corrections to the solutions of the rate equations of a system composed of arbitrarily many enzyme-catalyzed reactions inside a small sub-cellular compartment, while in [84] the results of [85] are further explored, in order to derive novel stochastic

mesoscopic rate equations, replacing the conventional macroscopic deterministic equations, in the case of subcellular (*i.e.*, at micron and submicron scales) enzyme reaction networks; these mesoscopic equations take into account the simultaneous influence of both intrinsic noise and substrate transport mode. Finally, in [86], the CME is used in order to obtain expressions for the instantaneous and time averaged rate of product formation in an enzymatic MM scheme in which the conventional substrate abundance assumption is not imposed on the system. Notably, the authors show that *(i)* the relationship between the average rate of product formation and the substrate concentration for a MM reaction with one enzyme molecule is (approximatively) given by a logarithmically corrected MM form, and that *(ii)* the relationship between the initial average product formation rate and the initial substrate concentration for an MM reaction with no reversible reaction and with any number of enzyme and substrate molecules is a sum of MM equations.

Several other analyses of the Michaelis-Menten reaction mechanism in the single-molecule context have been carried out. For example, for the case of only one enzyme molecule reacting with one substrate molecule, differences of $20 - 30\%$ between the average substrate concentrations as given by the reduced ODE model and the stochastic CME model have been found [7]. In the case of a Michaelis-Menten reaction mechanism with substrate inflow catalysed by one enzyme molecule, it was shown [172] that the relationship between the mean steady-state rate of product formation is given not by the MM equation, but by a more complex expression, which nonetheless reduces to the usual MM equation in the limit $K_m\Omega \gg 1$, with Ω denoting the volume.

We argue for the relative accuracy of our method, as the conditions under which the MM approximation is shown to be inaccurate in the above-mentioned studies do not hold for our λ-phage stochastic model: the substrate abundant, $K_M \gg 1$, and the enzymes do not operate near saturation.

Example 5.3.1

To illustrate the meaning of Theorem 5.3.1, we compare the stochastic trajectories of the product species P in the original and the reduced model, for different values of N.

We start by plotting the mean protein level for the original and for the reduced model, for a value of $N = 1$ (i.e., without scaling).

Then, we scale up the parameters k_1 and k_2, as well as the initial concentration of substrate T, in order to mimic the effect of choosing a larger N in 5.3.1.

Figure 5.6 shows that, as expected, the distance between the original and reduced model decreases in the scaled system (large N). This observation is consistent with those of [164], which state that the Michaelis–Menten approximation is applicable in discrete stochastic models, and that its validity conditions are the same as in the deterministic regime, as given in [168]. Otherwise said, that a stochastic simulation of the reduced mechanism $S \to P$ with effective propensity function $\frac{V_{max}S}{K_m+S}$ will closely approximate the solution of the full stochastic model of the Michaelis-Menten reaction set whenever $S_0 + K_m \gg E_T$. Indeed, scaling the MM system as in 5.3.1 widens the gap between these two quantities, as it implies increasing the values of both S_0 and K_m (as $K_m = \frac{k_1+k_2}{k_{-1}}$), while keeping enzyme levels E_T constant.

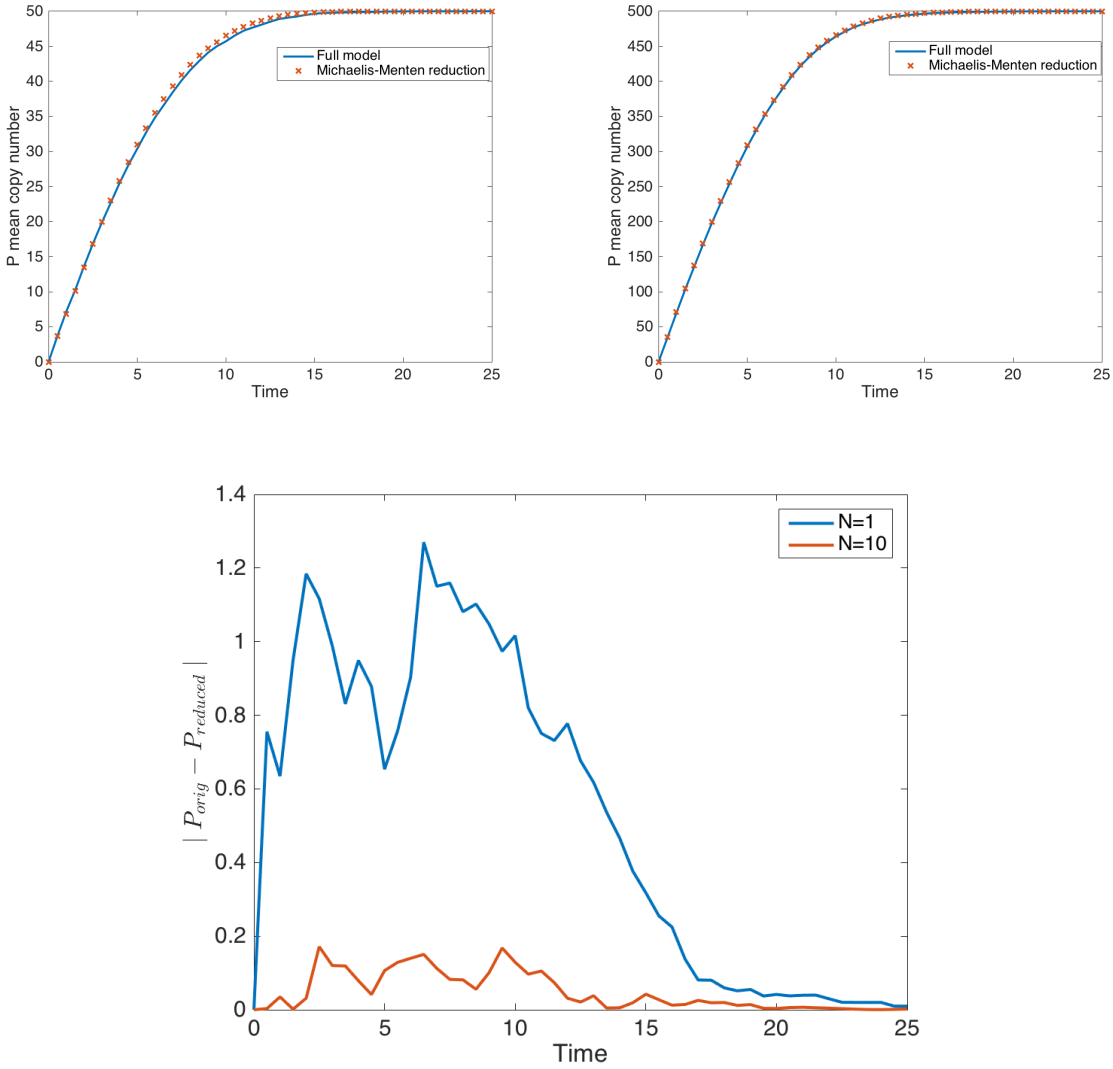


Figure 5.6: (Top) We plot the mean protein expression for the MM enzymatic scheme, for one hundred sampled traces, before and after the enzymatic catalysis reduction. (Top Left) No scaling applied (N=1), initial species abundance given by $x_S(0) = 50$, $x_E(0) = 1$, and parameters $k_{-1} = 1$, $k_1 = 10$, $k_2 = 10$ b) Parameters k2, k3, and the initial number of substrates S are scaled up by a factor of 10 (N = 10). (Bottom) We consider the reduction error to be given by the difference in mean protein copy numbers between the original and reduced model. As such, the reduction error is significantly smaller for the scaled models, cf. what is expected from Theoerem 5.3.1

5.3.2 Generalized competitive enzymatic reduction

Generalized competitive enzymatic reduction

The Michaelis-Menten enzymatic approximation can be generalized to a situation in which several substrates compete for binding to the same enzyme E. We will call such patterns of rules *competitive enzymatic rules*.

Assume the multiple alternative substrate scheme, in which the same enzyme E can reversibly bind one of n possible substrates S_i, and in doing so, lead to production of P_i:

$$E + S_i \underset{k_i^+}{\overset{k_i^-}{\rightleftharpoons}} E : S_i \xrightarrow{k_i} E + P_i, \qquad i = 1 \dots n \tag{5.2}$$

If the alternative substrate system i is taken as the leading substrate, the system contains $n-1$ competitors. Under the assumption of a generalized quasi-steady state condition stating that each complex $E : S_i$ reaches equilibrium fast enough, (*i.e.*, $\frac{dE:S_i}{dt} \approx 0$), and taking into account the conservation laws w.r.t. both the enzyme and the substrate amounts:

$$x_E(t) + \sum_{i=1}^{n} x_{E:S_i}(t) = x_E(0)$$
$$x_{S_i}(t) + x_{E:S_i}(t) + x_{P_i(t)} = x_{S_i}(0)$$

each such enzymatic scheme can be approximated by the following irreversible rule:

$$S_i \xrightarrow{\frac{k_i K_i E_T}{Z}} P_i, \tag{5.3}$$

meaning that P_i is produced at rate $\frac{k_i K_i x_{S_i} E_T}{Z}$, where

$$Z = 1 + \sum_{i \in \{1,\dots,n\}} K_i x_{S_i}$$
$$E_T = x_E + \sum_{i \in \{1,\dots,p\}} x_{E:S_i}$$

If the complex $E : S_i$ dissociates into E and S_i much faster it is converted into the product P_i, *i.e.* if $k_i^- \gg k_i$, the *rapid equilibrium approximation* can be applied, meaning that we can set $K_i \equiv \frac{k_i^-}{k_i^+}$, instead of the usual MM constant $\frac{k_i^- + k_i}{k_i^-}$. The rapid equilibrium approximation provides a more aggressive reduction than the quasi-steady state approximation w.r.t. reducing the complexity of the kinetic law, and, as such, whenever a model contains patterns that match the conditions for both methods, the rapid equilibrium approximation will have precedence in order to reduce the complexity of the rule rate laws.

We adapt the algorithms of [112] to perform the reduction of Kappa models containing patterns of multiple alternative substrate systems.

In this sense, we adapt the following nomenclature:

Definition 5.3.1 (**Kappa reactant, modifier, product**)
Given a rule (E_l, E_r, k), a Kappa species s is called

- a *reactant*, if it is a species occuring in E_l (cf. Definition 4.2.4), but it is not a species occuring in E_r ,
- a *modifier*, if it occurs in both E_l and E_r, and if the number of its occurrences in E_l equals the number of its occurrences in E_r ,
- a *product*, if it does not occur in E_l, and it occurs in E_r.

For the top-level competitive enzymatic reduction, Algorithm 2 scans the entire species set, in search for potential enzymes. For each such detected enzyme E, the system is transformed according to Eq. (5.3), and E is eliminated from the model.

Algorithm 2: Competitive enzymatic reduction

Input : A Kappa model M over a set of species $\mathcal{S}$ and observables $\mathcal{O}$
Output: The reduced Kappa model M' over a set of species $\mathcal{S}'$ and observables $\mathcal{O}$

```
M' ⟵ M;
for s ∈ S do
    C ⟵ RapidEqCondition(M, s);
    if C ≠ ∅ then
        M' ⟵ RapidEqTransform(M', s, C)
    end
end
return M'
```

Checking if E is an enzyme is done via procedure RapidEqCondition, which ensures that[2]:

(i) E is not an observable in the Kappa model M;

(ii) E is a reactant (cf. Definition 5.3.1) in at least one rule $r : (E_l, E_r, k)$ of M;

(iii) each rule $r : (E_l, E_r, k)$ in which E is a reactant is a reversible rule, in which E_l contains only two species, E and S, and in which E_r is obtained by applying a *binding* transformation on E_l, between species E and S;

(iv) for each such rule r, the right-handside species, which we denote $E : S_i$ for readability, is not an observable, has an initial abundance 0, occurs as a reactant or product in exactly one rule, and never occurs as a modifier;

(v) there exists an irreversible rule $r_2 : (E_{l_2}, E_{r_2}, k_2)$, with $E_{l_2} = E : S_i$, and such that E_{r_2} contains only the enzyme species E and a product species P: such a reaction converts $E : S_i$ into a product and releases the enzyme;

[2]These tests are equivalent to those shown in [135] and [112].

(vi) either the *quasi-steady state* or the *rapid equilibrium* assumptions are verified, *i.e.*, either $S_i(0) + \frac{k_i^- + k_i}{k_i^+} \gg E_T$, or $k_i^- \gg k_i$.

Procedure RapidEqCondition(M,E)

Input : A Kappa model M over a set of species $\mathcal{S}$ and observables $\mathcal{O}$, and a species E from its $\mathcal{S}$

Output: Checks if E is an enzyme, and if so, returns the set of configurations corresponding to all substrates E binds to

$C \longleftarrow \emptyset$;
if $E \in \mathcal{O}$ *or* E *isn't a reactant in any rule of* M **then**
  return $\emptyset$;
end
for *every rule* r_1 *in which* E *is a reactant* **do**
  if $r_1 \neq E, S_i \leftrightarrow E : S_i @ k_i^+, k_i^-$ **then**
    return $\emptyset$;
  end
  if $x_{E:S_i}(0) \neq 0$ *or* $E : S_i \in \mathcal{O}$ **then**
    return $\emptyset$;
  end
  if $E : S_i$ *is a reactant in more than one rule, or a modifier in any rule, or a product in any other rule than* r **then**
    return $\emptyset$;
  end
  if $\nexists r_2 = E : S_i \rightarrow E, P_i @ k_i$ **then**
    return $\emptyset$;
  else
    if $\frac{k_i}{k_i^-} > threshold$ **then**
      if $\frac{E_T}{x_{S_i}(0) + (k_i^- + k_i)/k_i^+} > threshold$ **then**
        return $\emptyset$;
      else
        $C \longleftarrow C \cup \{(S_i, E : S_i, \frac{k_i^+}{k_i^- + k_i}, k_i, r, r_2)\}$;
      end
      $C \longleftarrow C \cup \{(S_i, E : S_i, \frac{k_i^+}{k_i^-}, k_i, r, r_2)\}$;
    end
  end
end
return C;

If species E satisfies these conditions, a set of configurations is formed; each configuration includes a substrate S_i to which the enzyme binds, the complex $E : S_i$ it forms by binding to S_i, the equilibrium constant k_{new} of the binding rule and the rate k_i of the production rule containing $E : S_i$ as a reactant, as well as the names of the two rules.

This configuration set is then used by procedure RapidEqTransform to reduce the model:

(i) for an enzyme E, it loops through the set of configurations, to form Z - the expression used in the denominator of each new rate law;

(ii) for each configuration $(S_i, E : S_i, k_{new}, k_i, r_1, r_2)$, it makes the substrate S_i a reactant for r_2, and changes the rate constant of r_2 accordingly;

(iii) finally E, $E : S_i$ and r_1 are removed from the model.

Procedure RapidEqTransform(M,S,C)

Input : A Kappa model M over a set of species $\mathcal{S}$ and observables $\mathcal{O}$, an enzyme E and its corresponding configuration set given by the RapidEqCondition procedure

Output: The model M, from which the enzyme E has been abstracted

/ compute the new kinetic law expression */*

$Z \longleftarrow 1 + \sum_{(S_i, E:S_i, k_{new}, r, r_2) \in C} k_{new} * x_{S_i}$;

for $(S_i, E : S_i, k_{new}, k_i, r, r_2) \in C$ **do**

  add S_i as a reactant in r_2;

  $rate_{constant}(r_2) = \frac{k_2 * x_E(0) * k_{new}}{Z}$;

  remove $E : S_i$ from M;

  remove rule r from M;

end

remove E from M;

return M;

5.3.3 Operator site reduction

Models of genetic circuits generally include multiple operator sites which can be occupied by transcription factors. Moreover, it is generally the case that the rates at which transcription factors reversibly bind operator sites are rapid with respect to the rate of transcription initiation, *i.e.*, the rate of open complex formation. The number of operator sites is also typically considerably smaller than that of RNA polymerase (RNAP) and of transcription factor molecules. Consequently, a method similar to the enzymatic catalysis reduction scheme can be used to systematically merge rules and remove operator sites and their complexes from rule-based models of genetic circuits. Intuitively, in this context, the operator site takes the role of the enzyme, and the transcription factor(s) act as the substrate.

Assume an operator S, that can bind transcription factors and RNAP, in $N+1$ possible configurations, let S_o be the operator in free form (unbound), and let $C_i \equiv (S_i,\ K_i, X_i)$, with $1 \leq i \leq N$ denote each possible configuration, where S_i is the i^{th} bound complex of S, K_i is its equilibrium constant (*i.e.*, the ration between the forward and backward rate constants), and X_i is the product of the states of the substrates for each component of the complex in this configuration.

Then, assuming rapid equilibrium, the probability of S being in each configuration is given by:

$$\Pr(C_i) = \begin{cases} \frac{1}{Z},\ \text{if } i = 0 \\ \frac{K_i X_i}{Z},\ \text{otherwise} \end{cases}, \tag{5.4}$$

with $Z = 1 + \sum_{i=1}^{N} K_i X_i$.

Assuming that $S_T = x_{S_0}(0)$, the fraction of operators in the i^{th} configuration is given by:

$$x_{S_i} = \Pr(C_i) \cdot S_T. \tag{5.5}$$

Then, Eq. (5.5) can be used for *operator site reduction* of a Kappa model, in a similar fashion to the competitive enzymatic reduction. The operator site reduction for a Kappa model M consists in Algorithm 3 checking each species S, using procedure OpSiteCondition. If S satisfies the operator conditions, a set of configurations C corresponding to the transcription factors that compete to bind to s is formed, and then used in the reduction performed by procedure OpSiteTransform.

Assuming that an operator is a species S that is small in copy number, and which can neither be produced, nor degraded, procedure OpSiteCondition checks that S is an operator, using the following conditions:

Algorithm 3: Operator-site reduction

Input : A Kappa model M over a set of species $\mathcal{S}$ and observables $\mathcal{O}$
Output: The reduced Kappa model M' over a set of species $\mathcal{S}'$ and observables $\mathcal{O}$

```
M' ⟵ M;
for s ∈ S do
    C ⟵ OpSiteCondition(M, s);
    if C ≠ ∅ then
        M' ⟵ OpSiteTransform(M', s, C)
    end
end
returnM'
```

(i) S's initial copy number is not higher than a threshold;

(ii) S is not an observable, and is a reactant in at least one reversible rule, $r : (E_l, E_r, k)$, which in turn is a reversible complex formation rule: E_r must be obtained by applying the binding operation between the species of E_l ;

(iii) the operator complex E_r mentioned above is not an observable, it is uniquely produced by r and it is never a reactant in any rule of M;

(iv) each rule r' in which E_r appears as a modifier is irreversible, has no reactants, no other modifiers, and only one product;

(v) for each such rule r, a configuration (E_r, x_{T_i}, K_i, r) is created, with E_r the product species $S : T_1 : \ldots : T_j$, $X_i = \frac{k_+}{k_-} \cdot \prod_{1 \leq k \leq j} x_{T_k}$ the product of copy numbers for reactants of r, except S, $K_i = k_+/k_-$, the ratio between the forward and backward rates of r, and r the rule name;

(vi) finally, if all the above conditions are met, the set of all such configurations for S is returned.

Procedure OpSiteTransform is then used to apply the reduction:

(i) for an operator S, it loops through the set of corresponding configurations to form Z, the expression used in the denominator of each new rate;

(ii) it then considers every configuration ($S : T_1 : \ldots : T_j$,K,r) of the set, and for every rule r' in which $S : T_1 : \ldots : T_j$ appears as a modifier, it changes its rate constant accordingly, and adds $T_1, \ldots, T_j$ as modifiers;

(iii) finally, S, sc, and r are removed.

Procedure OpSiteCondition(M,s)

Input : A Kappa model M over a set of species $\mathcal{S}$ and observables $\mathcal{O}$, and a species S from its $\mathcal{S}$

Output: Checks if S is an operator site, and if so, returns the set of configurations corresponding to transcription factors bound to S

$C \longleftarrow \emptyset$;
if $x_S(0) > threshold$ **then**
  return $\emptyset$;
end
if $S \in \mathcal{O}$ *or* S *is being produced by any rule of* M **then**
  return $\emptyset$;
end
for *every rule* r *in which* S *is a reactant* **do**
  if $r \neq S, T_1, \ldots, T_j \leftrightarrow S : T_1 : \ldots : T_j @ k_+, k_-$ **then**
    return $\emptyset$;
  else
    if $S : T_1 : \ldots : T_j \in \mathcal{O}$, *is produced by any other rule of* M *except* r, *or appears as a reactant in any rule of* M **then**
      return $\emptyset$;
    end
    for *every rule* r' *in which* $S : T_1 : \ldots : T_i$ *appears as a modifier* **do**
      if $r' \neq S : T_1 : \ldots : T_j \rightarrow S : T_1 : \ldots : T_j, P @ k_2$ **then**
        return $\emptyset$;
      end
      $K_i \longleftarrow \frac{k_+}{k_-}$;
      $X_i \longleftarrow \frac{k_+}{k_-} \prod_{1 \leq k \leq j} x_{T_k}$;
      $C \longleftarrow C \cup \{(S : T_1 : \ldots : T_j, X_i, K_i, r)\}$;
    end
  end
end
return C;

Procedure OpSiteTransform(M,S,C)

Input : A Kappa model M over a set of species $\mathcal{S}$ and observables $\mathcal{O}$, an operator site S and its corresponding configuration set given by the OpSiteCondition procedure

Output: The model M, from which the operator site S has been abstracted

/ compute the new kinetic law expression */*

$Z \longleftarrow 1 + \sum_{(S:T_1:\ldots:T_j,X_i,K_i,r)\in C} X_i$;

for $(S : T_1 : \ldots : T_j, X_i, K_i, r) \in C$ **do**
  for *every rule* $r_2 = S : T_1 : \ldots : T_j \to S : T_1 : \ldots : T_j, P@k_2$ **do**
    add $T_1, \ldots, T_j$ as modifiers in rule r_2;
    $rate_{constant}(r_2) \longleftarrow \frac{k_2 \cdot K_i \cdot x_S(0)}{Z}$
  end
  remove $S : T_1 : \ldots : T_j$ from M;
  remove r from M;
end
remove S from M;
return M;

Example 5.3.2
We illustrate the operator-site transformation on a small subnetwork of the λ-phage model. The four rules presented below model the binding of the agent RNAP to the operator site of the agent PRE and subsequent production of protein CI. Agent PRE binds either only RNAP (at rate k_{1+} and k_{1-}), or simultaneously with CII (at rate k_{2+} and k_{2-}). The protein can be produced whenever PRE and PRE are bound, but the rates will be different depending on whether only RNAP is bound to the operator (rate k_a), or, in addition, CII is bound to the operator (rate k_b):

$$\mathrm{PRE(cii[.], rnap[.]), RNAP(p1[.], p2[.])} \leftrightarrow \mathrm{PRE(cii[.], rnap[1]), RNAP(p1[1], p2[.])}@k_{1+}, k_{1-}$$

$$\mathrm{PRE(cii[.], rnap[.]), CII(pre[.]), RNAP(p1[.], p2[.])} \leftrightarrow \mathrm{PRE(cii[1], rnap[2]), CII(pre[1]), RNAP(p1[2], p2[.])}@k_{2+}, k_{2-}$$

$$\mathrm{PRE(cii[.], rnap[1]), RNAP(p1[1], p2[.])} \to \mathrm{PRE(cii[.], rnap[1]), RNAP(p1[1], p2[.]), 10CI(ci[.], or[.])}@k_a$$

$$\mathrm{PRE(cii[1], rnap[2]), CII(pre[1]), RNAP(p1[2], p2[.])} \to \mathrm{PRE(cii[1], rnap[2]), CII(pre[1]), RNAP(p1[2], p2[.]), 10CI(ci[.], or[.])}@k_b$$

After the operator site reduction, the operator PRE is eliminated from each of the two competing enzymatic catalysis patterns. Consequently, the production of CI is modeled only as a function of RNAP and CII:

$$\text{RNAP(p1[.], p2[.])} \rightarrow \text{RNAP(p1[.], p2[.])}, \text{10CI(pr[.], ci[.])@ } k_{new1}$$

$$\text{RNAP(p1[.], p2[.])}, \text{CII(pre[.])} \rightarrow \text{RNAP(p1[.], p2[.])}, \text{CII(pre[.])}, \text{10CI(pr[.], ci[.])@ } k_{new2},$$

where the rate constants are appropriately modified:

$$\begin{aligned} k_{new1} &= \frac{k_a \cdot \frac{k_1^+}{k_1^-} \cdot PRE_0}{Z} \\ k_{new2} &= \frac{k_b \cdot \frac{k_2^+}{k_2^-} \cdot PRE_0}{Z} \\ Z &= 1 + \frac{k_1^+}{k_1^-} \cdot x_{RNAP} + \frac{k_2^+}{k_2^-} \cdot x_{RNAP} \cdot x_{CII} \end{aligned}$$

5.3.4 Fast dimerization reduction

Finally, a similar reduction reasoning can be applied to fast dimerization rules:

$$M + M \underset{k}{\overset{k^-}{\rightleftharpoons}} M_2$$

Under the assumption that both rates k and k^- are fast compared to other rules involving M or M_2, it is common to also assume that the rule is equilibrated, that is:

$$k x_M^2 - k^- x_{M_2} = 0,$$

where x_M and x_{M2} denote the copy number (at time t, but for notation ease, we omit the time in the notation), of monomers and dimers respectively. Such an assumption allows for a reduction in which dimerization rules are removed, and the number of dimers is expressed in terms of the total number of monomer molecules M.

The respective monomer and dimer copy numbers can be expressed as fractions of the total quantity:

$$\begin{aligned} x_M &= \frac{1}{4K}\left(\sqrt{8KM_T(t)+1}-1\right), \\ x_{M_2} &= \frac{M_T(t)}{2} - \frac{1}{2}x_M, \end{aligned} \tag{5.6}$$

where $K = \frac{k}{k^-}$ and $M_T(t) = x_M + 2x_{M_2}$.

Our algorithm searches for such dimerization rules. Suppose that a pair of reversible rules $M, M \leftrightarrow M_2$ is detected, in which M_2 is obtained by binding the two M reactants.

Before proceeding to the reduction, we check whether an M dimer is produced elsewhere, or if the monomer is a modifier elsewhere. These checks are necessary because they prevent from deviating from the assumed equilibrium. If all checks passed, the dimerization rule can be eliminated. Instead, a new species denoting the total quantity of momonmers M_T is introduced. It used to replace both the monomer M or dimer M_2 species in rules that previously involved them (the rates are adapted according to (5.6)).

Consequently, the dimerization reduction of a Kappa model M is done by looping through M's rule set and reducing the dimerization rules as described above.

Algorithm 4 checks every rule $r \in M$ using procedure DimerCondition. If the dimerization conditions are satisfied, a record Q of the monomer species, dimer species, and equilibrium constant is created, and then used to perform the reduction using procedure DimerTransform.

Algorithm 4: Fast dimerization reduction

Input : A Kappa model M over a set of species $\mathcal{S}$ and observables $\mathcal{O}$, containing a set of rules $\mathcal{R}$

Output: The reduced Kappa model M' over a set of species $\mathcal{S}'$ and observables $\mathcal{O}$

```
1 M' ⟵ M;
2 for r ∈ R do
3 |   Q ⟵ DimerCondition(M, r);
4 |   if Q ≠ ∅ then
5 |   |   M' ⟵ DimerTransform(M', r, Q)
6 |   end
7 end
8 return M'
```

Cf. procedure DimerCondition, a rule $r : (E_l, E_r, k)$ is a dimerization rule that can be reduced, if it satisfies:

(i) r is a reversible rule;

(ii) $E_l = M, M$, *i.e.* the rule's left-handside is comprised of two occurences of the same species M, and E_r is obtained by adding a bond (or more) between the two Ms;

(iii) the monomer form (M) does not appear as a modifier in any rule;

(iv) the dimer form (E_r) does not appear as a product in any other rule other than the dimerization rule r.

When a dimerization rule r is found, the model is abstracted as follows (cf. procedure DimerTransform):

(i) a new species, M_t, accounting for the total amount of species M is introduced, with initial abundance $x_{A_t}(0) = x_A(0) + 2x_{A_2}(0)$;

Procedure DimerCondition(M,r)

Input : A Kappa rule r
Output: Checks if r is a dimerization rule

```
1  if r = A, A ↔ A2@k+, k− then
2  |  if A is never used as a modifier and A2 is only produced by rule r then
3  |  |  Sm ⟵ A;
4  |  |  Sd ⟵ A2;
5  |  |  return < Sm, Sd, k+/k− >;
6  |  else
7  |  |  return ∅;
8  |  end
9  else
10 |  return ∅
11 end
```

(ii) in all rules containing the monomer as a reactant, the monomer is replaced by M_t, and the rates are updated accordingly;

(iii) for all rules containing the dimer as a modifier, the rates are adapted accordingly;

(iv) in all rules containing the dimer as a reactant, the dimer is replaced with by M_t, and the rates are adapted accordingly ;

(v) in all rules containing the monomer as a product, the monomer is replaced by M_t;

(vi) the dimerization rule, the momoner and the dimer patterns are removed from the model.

Procedure DimerTransform($M', r, < S_m, S_d, K_e >$)

Input : A dimerization rule r in a Kappa model M, alongside its record $< S_m, S_d, k_+/k_- >$

Output: The model M, from which the dimerization rule r has been reduced

add a new species, S_t to M;
set $x_{S_t}(0) = 2 * x_{S_d}(0) + x_{S_m}(0)$;
for *every rule* r' *in which* S_m *appears as a reactant* **do**
    replace S_m with S_t in r';
    replace x_{S_m} with $\frac{1}{4\frac{k_+}{k_-}}(\sqrt{8\frac{k_+}{k_-}x_{S_t}+1}-1)$ in $rate(r)$;
end
for *every rule* r' *in which* S_d *appears as a reactant* **do**
    replace S_d with $2S_t$ in r';
    replace x_{S_d} with $\frac{x_{S_t}}{2} - \frac{1}{8\frac{k_+}{k_-}}(\sqrt{8\frac{k_+}{k_-}x_{S_t}+1}-1))$ in $rate(r')$
end
for *every rule* r' *in which* S_d *appears as a modifier* **do**
    replace x_{S_d} with $\frac{x_{S_t}}{2} - \frac{1}{8\frac{k_+}{k_-}}(\sqrt{8\frac{k_+}{k_-}x_{S_t}+1}-1))$ in $rate(r')$
end
for *every rule* r *in which* S_m *appears as a product* **do**
    replace S_m with S_t in r
end
remove S_m, S_d, and r from M;
return M;

5.3.5 Top-level reduction algorithm

Our top-level reduction algorithm is equivalent to that shown in [135] and [112], except for the adaptations resulting from the fact that the data structure used to represent species in rule-based models are site-graphs, which are different from vectors of species' multiplicities used in reaction-based models. Also, unlike in the original algorithm, there is no need to check the form of the reaction rate function, as in Kappa rule-based models one implicitly assumes chemical kinetics to follow the mass-action rule.

As mentioned previously, for reasons explained in Note 5.2.1, when designing the reduction algorithm we assume that the rule-based model is such that all left-handside and right-handside of a rule represent mixtures, that is, each rule is equivalent to one reaction. Hence, in our static inspection of rules, we test species, *i.e.,* fully defined connected mixtures). The extension to the case where this assumption does not hold is subject to future work, and is discussed in more detail in Section 5.4.2.

Our reduction of a Kappa system $\mathcal{R} = (\boldsymbol{x}_0, \mathcal{O}, \{r_1, \ldots, r_n\})$ is performed by static analysis over the rule-set $\mathcal{R}$, in search for one of the interaction patterns which are consequences of the

theory previously presented.

A run of a complete reduction step starts by applying a *modifier elimination* procedure, aimed at reducing complexity without losing accuracy. The modifier elimination abstraction can be applied when a species only appears as a modifier throughout a model; such a species will never change its copy number throughout the dynamics, and therefore, its quantity will be constant. In this case, the species can be eliminated from the reactions and each corresponding rate law will be multiplied by the initial copy number of this species.

We follow by sequentially applying the *competitive enzymatic*, *operator site* and *fast dimerization* reductions. The last operation consists in a *similar reaction composition* procedure, aimed at combining the structurally similar reactions that are often generated by the operator model abstraction. In similar reaction composition, reactions that have the same reactants, modifiers and products are combined into a single reaction, by summing their rate laws. We note that both modifier elimination and similar reaction composition are *exact* reductions: applying them does not change the semantics of the rule-based system.

The abstraction methods are applied in a loop, until they generate no more changes in the model, as presented in the top-level algorithm shown in Alg. 5.

Algorithm 5: Top-level approximation algorithm.

Input : A Kappa system $\mathcal{M} = (\boldsymbol{x}_0, \mathcal{O}, \{r_1, \ldots, r_n\})$ over a set of species $\mathcal{S}$ and observables $\mathcal{O}$.

Output: A Kappa model $\mathcal{M}'$ over a set of species $\mathcal{S}'$ and observables $\mathcal{O}$.

repeat
  $\mathcal{M}' \longleftarrow \mathcal{M}$
  $\mathcal{M} \longleftarrow$ Modifier elimination($\mathcal{M}$)
  $\mathcal{M} \longleftarrow$ Competitive enzymatic reduction($\mathcal{M}$)
  $\mathcal{M} \longleftarrow$ Operator-site reduction($\mathcal{M}$)
  $\mathcal{M} \longleftarrow$ Fast dimerization reduction($\mathcal{M}$)
  $\mathcal{M} \longleftarrow$ Similar reaction composition($\mathcal{M}$)
until $M' = M$;

5.4 Results and Discussion

5.4.1 Results: reduction of the λ-phage decision circuit

We test our reduction algorithm on the Kappa model of the λ-phage, as presented in Section 5.2.

Simulations were carried out on the complete chemical reaction genetic circuit model which contains 92 rules, 13 proteins and 61 species (the contact map of the original system is shown in Figure 5.5). After applying the reduction, the Kappa model is reduced to 11 rules and 5 proteins.

In Figure 5.7, we plot the mean for the `CI` copy number obtained from 100 runs of the original and of the reduced model, and the graphs show agreement.

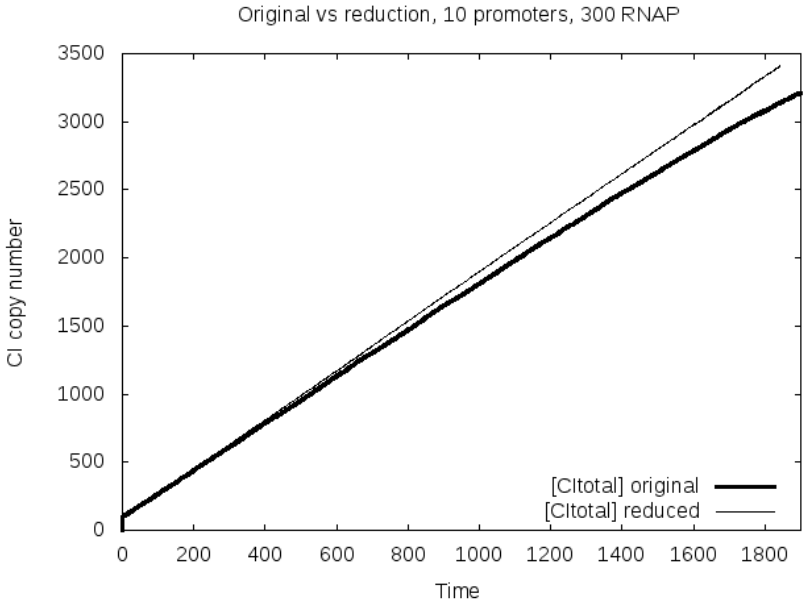


Figure 5.7: Average trace of 100 simulations of the original full λ-phage model (solid) and the reduced model (thin) after the reduction, for initially 10 λ phage cells (multiplicities of infection – MOI's). The simulation time for one simulation trace of the original model is ≈ 40 minutes of CPU time, and of the reduced model is 5 seconds of CPU time. The initial number of proteins `CI`, `Cro`, `CII` and `CIII` and `N` is set to 100.

In Figure 5.8, we compare the probability of lysogeny before and after the reduction of the model (lysogeny profile is detected if there are 328 molecules of `CI` before there are 133 molecules of `Cro`). The graphs show overall agreement in predicting the lysogeny profile. More precisely, for a value of MOI (multiplicity of infection) of at most 2, the probability of lysogeny is almost negligible. For an MOI of 3, both graphs show that lysogeny and lysis are equally

probable (the reduced model reports slightly larger probability), and for an MOI greater than 4, both graphs show that lysogeny is highly probable. We note the considerable speedup in simulation: while one simulation of the original model takes about 40 mins, one simulation of the abstracted model takes about 5 seconds. A prototype of the tool is available for download [14].

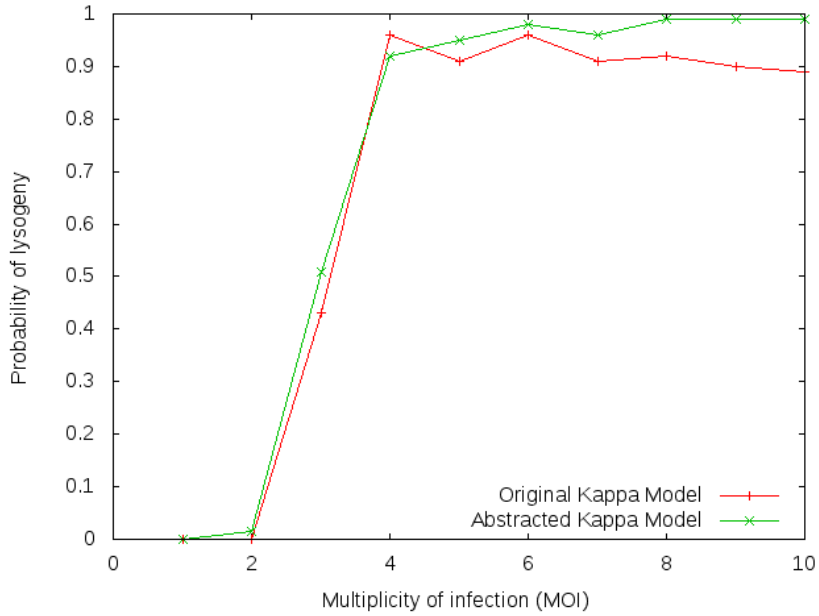


Figure 5.8: Comparison of the probability of lysogeny before and after the reduction of the full λ-phage model (lysogeny profile is detected if there are 328 molecules of `CI` before there are 133 molecules of `Cro`). The profile was obtained by running 1000 simulations of the model for one cell cycle (2100 time units), for MOIs ranging from 1 to 10.

5.4.2 Conclusion and future work

In this chapter, we argue for the use of rule-based modeling language in prototying genetic circuits, and show how the multi-scaleness of biological systems can be exploited for model reduction in the stochastic case. As such, we construct a detailed rule-based Kappa model of the λ-phage decision circuit, and subsequently propose an model approximation method based on variations of the Michaelis-Menten enzymatic scheme. Our method can be seen as a first step towards a systematic time-scale separation of stochastic rule-based models. As such, it can be extended in several directions.

From species to "don't care, don't write" patterns

When constructing our reduction method, besides the "every rule is a reaction" assumption, we also assume the following:

(i) operator sites have no internal states;

(ii) the definition of *dimer* assumes binding between two *identical* monomers. This means that given a species M with a site that has two possible internal states $M(x\tilde{}u\tilde{}p)$ (u for *unphosphorylated*, and p for *phosphorylated*), we consider that the reaction :

$$M(x\tilde{}u), M(x\tilde{}p) \leftrightarrow M(x\tilde{}u[1]), M(x\tilde{}p[1]),$$

in which an unphosphorylated M reversibly binds a phosphorylated M is *not* a dimerization reaction;

(iii) similarly, asymmetric dimerization (*i.e.*, through binding on different sites of a monomer species) is not taken into consideration by our reduction method, as it can lead to polymer species; consequently, out method only searches for *strict dimers*, obtained via symmetric dimerization (*i.e.*, binding on the same site);

We note that assumptions *(iii)* and *(iv)* amount to prohibiting *hetero-dimers*. Moreover, assumption *(iii)* ensures that no infinite species - such as polymers - occur in the model.

We argue that on the one hand that these restrictions are respected by models of genetic regulatory networks, and that on the other hand, they allow for sound model reduction, in line with the original method presented in [112]. What's more, we have seen that the specificities of the interaction mechanisms of the λ-phage decision circuit result in the creation of a rule-based model in which all left-hand-side and all right-hand-side terms represent mixtures, *i.e.*, each rule is equivalent to one reaction. Consequently, our reduction algorithm is guaranteed to work when testing for *species*, *i.e.*, fully defined connected mixtures.

However, our algorithm can be extended in order to guarantee a sound reduction of rules containing "don't care, don't write" patterns. Such an extension can be carried out by simply modifying the way in which reduction conditions are tested.

More specifically, consider testing the condition on line 2 of procedure DimerCondition:

"the dimer A_2 is produced only by rule r." (5.7)

In the original algorithm of [112], A_2 denotes the dimer *species*, so checking condition (5.7) amounts to searching for occurences of species A_2 in the right-handside (*rhs*) (*i.e.*, among the products) of other reactions.

However, in rule-based models, a rule generally subsumes many possible reactions, and a pattern subsumes many possible species. If A_2 is a "don't care, don't write" pattern (*i.e.*, it contain agents with partially specified interfaces), condition (5.7) is verified only if *none of the species S that match the pattern A_2 can be produced by any rule other than r*, *i.e.*, if none of the species that match A_2 also match a pattern that appears in the *rhs* of any rule of the Kappa model $\mathcal{M}$, except r:

$$\forall r, r' \in \mathcal{M}, \begin{cases} r = (E_l, E_r, k) \\ r' = (E'_l, E'_r, k') \end{cases} \Rightarrow \nexists \text{ species } A_2 \text{ s.t. } \begin{cases} A_2 \models E_r \\ A_2 \models E'_r \end{cases}$$

This latter condition can be checked using the notion of *pattern compatibility*. We now think about a pattern Z_1 in terms of its extension $Z_1^\diamond$, *i.e.*, the set of species that match Z_1, in order to account for the ways in which any such species can match Z_1.

Then, two patterns Z_1 and Z_2 are said to be *compatible* if the intersection of their extensions is non-empty:

$$Z_1 \text{ and } Z_2 \text{ are compatible } \Leftrightarrow Z_1^\diamond \cap Z_2^\diamond \neq \emptyset$$

The fast dimerization reduction can then proceed if and only if pattern A_2 is not compatible with any pattern that appears in the *rhs* of any other rule r':

$$\forall r' : E_{l'} \rightarrow E_{r'} \in \mathcal{M}, r' \neq r \Rightarrow \nexists P_i \in E_{r'} \text{ s.t. } A_2 \text{ and } P_i \text{ are compatible.}$$

For example, consider a species $A(x, y)$ with two sites, x and y, and two dimerization rules:

$$r_1 : \quad A(x[.]), A(x[.]) \leftrightarrow A(x[1]), A(x[1]) \quad @ \quad k_1$$

$$r_2 : \quad A(y[.]), A(y[.]) \leftrightarrow A(y[1]), A(y[1]) \quad @ \quad k_2$$

Denote $A(x[1]), A(x[1])$ with P_1, and $A(y[1]), A(y[1])$ with P_2.

As the dimer species $A(x[1], y[2]), A(x[1], y[2])$ belongs to both $P_1^\diamond$ and to $P_2^\diamond$, P_1 and P_2 are compatible, *i.e.*, condition (5.7) is not satisfied, and thus the fast dimerization reduction cannot proceed.

Under the assumption of "no infinite species", checking for pattern compatibility can be achieved, for example, by using the tool KaDe[28], which compiles Kappa models into reaction networks, in order to generate ODE models. As such, one of the features of KaDe deals with enumerating chemical species of the Kappa model, and one can consequently exploit this feature to check if the same species can be modeled by two different patterns, *i.e.*, if two patterns are *compatible*.

These extensions, as well as similar ones regarding operator site and fast enzymatic reductions are currently under implementation.

Future directions

Other future directions worth exploring include investigating how the algorithm presented herein can exploit the specificities of rule-based models to result in more efficient pattern recognition, as well as testing the applicability of the reduction algorithm on other case studies.

For example, one might consider how the set of approximation patterns can be extended as to obtain good reductions for complex models of signaling pathways. More precisely, while our tool is applicable to any rule-based model, the chosen set of approximation patterns are tailored for genetic regulatory networks, and may thus not provide significant reductions when applied to signaling pathways. To illustrate this, we analyze an EGF/insulin crosstalk model, and we observe that, unlike the λ-phage example, the number of dimerisation events does not represent a significant portion of the system's total events (see Figure 5.9).

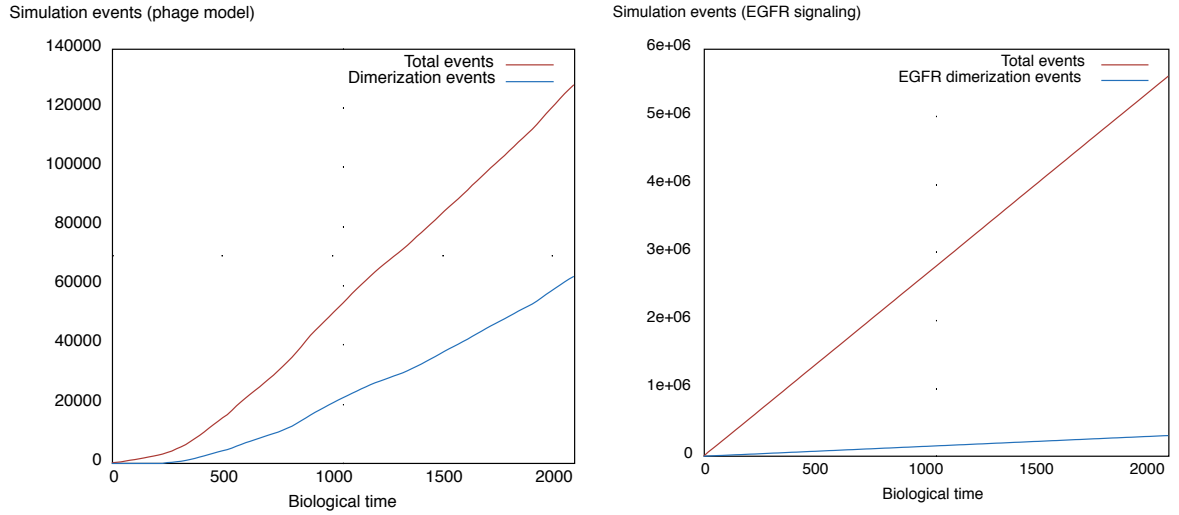


Figure 5.9: a) The ratio of dimerisation events vs. total events in lambda phage model. The number of dimerisation events takes roughly half of the total events over the whole cell cycle. b) The ratio of dimerisation events vs. total events in EGFR/insulin model. The number of dimerisation events takes only a small fraction of the total events over the whole cell cycle.

To this end, more patterns could be included in our approximation, for the purpose of reducing models of signaling pathways (*e.g.*, approximating multiple phosphorylation events).

Finally, we note that the major issue of model reduction techniques is related to the quantification of the approximation error, without simulating the original system. In this sense, in the next chapter we propose an approximation method of biochemical reaction networks that exploits the multiscaleness of biochemical systems (a property also exploited by the Michaelis-Menten reduction scheme), in which the reduction guarantees are the major requirements.

Chapter 6
Tropical Abstraction of Biochemical Reaction networks with guarantees

The work presented in this chapter has been presented at and accepted for publication in the proceedings of the Static Analysis in Systems Biology (SASB) 2018 workshop (to appear) [12].

6.1 Introduction

The work presented in this chapter tackles the design of an approximation method for ODE models of biochemical networks, in which reduction guarantees are the major requirement. Our method combines abstraction and numerical approximation, and aims at providing a better understanding of tropical reduction methods. We abstract the solution of the original system of ODEs by a "box" that over-approximates the state of the original system, providing lower and upper bounds for the value of each variable of the system in its current state. The simpler equations (which we call *tropicalized*) that define the hyperfaces of the box are obtained by combining the dominance concept, borrowed from tropical analysis, with symbolic bounds propagation. Mass invariants of the initial system of ODEs are used to refine the computed bounds, thus improving the accuracy of the method. The resulting (simplified) system provides *a posteriori* time-dependent lower and upper bounds for the concentrations of the initial model's species, and thus bounds on numerical errors stemming from tropicalization. This means that no information on the original system's trajectory is needed - the most important advantage of our approach. By contrast, the main difficulty of applying the classical QSS and QE reductions to biochemical models is that QE reactions and QSS species need to be specified *a priori*, which implies that some knowledge about the initial system's behavior is necessary. This, in turn, means that significantly high-dimensional, non-linear systems cannot benefit from these reductions, as their analysis can be prohibitive in practice. An approach similar with respect to providing *a posteriori* time-dependent lower and upper bounds has been proposed in [37], where the differential semantics of rule-based models with non-contracting dynamics and unbounded sets of variables are treated. Rather than using dominance relations between ODE terms, a finite set of patterns is used in order to bound the number of occurrences of each pattern. Further related works, similar in the sense that they provide automatizible reduction methods

with strong reduction guarantees are described in [62, 63]. However, both of these works are designed specifically for rule-based models, where they exploit the site-graph encoding of species' structure, rather than the dominance regions.

Depending on the chosen granularity of mass-invariant-derived bounds, the method presented in this paper can be used either to reduce models of biochemical networks, or to quantify the approximation error of tropicalization-based reduction methods that do not involve guarantees. The guarantees of our method are obtained by formalizing the soundness relation between the original system of equations and the abstract system of ordinary differential equations operating on the coordinates of the hyper-faces of the box. The solution of a sound abstraction of an original system of differential equations, starting from a box that contains the initial state of the original system, defines a sound abstraction of the solution(s) of the original system. We apply our method to several case studies.

The rest of this chapter is organized as follows. In Section 6.2 we define the setting and concepts used in our approach, as well as introduce motivating examples. We then formally present and justify the method for deriving the system of reduced ODEs over the lower and upper bounds of species' concentrations in Section 6.3. In Section 6.4, we show that our method can be used to quantify the approximation error of tropicalization-based reduction heuristics, while in Section 6.5 we show that our approach outperforms (in terms of accuracy) several existing interval numerical methods for the initial value problem (IVP). We discuss and conclude in Section 6.6.

6.2 Definitions and Motivating Examples

6.2.1 General Setting and Definitions

Herein, we focus on the deterministic model semantics of biochemical reaction networks, as presented in Definition 3.1.1 of Chapter 1. That is, the mass-action dynamics of a reaction system of the form:

$$r_j : \sum_i \alpha_{ji} x_i \xrightarrow{k_j} \sum_i \beta_{ji} x_i, \tag{6.1}$$

over a set of species $\mathcal{S} = \{x_1, \ldots, x_s\}$ is described by a system of ODEs:

$$\frac{d\boldsymbol{x}(t)}{dt} = \nabla^T f(\boldsymbol{x}(t)), \tag{6.2}$$

with

$$f_j(\boldsymbol{x}) = k_j \prod_{i=1}^{n} x_i^{\alpha_{ji}} \tag{6.3}$$

Then, the mass-action kinetics equation of the i-th species x_i reads as a sum of monomials:

$$\frac{dx_i}{dt} = \sum_{j=1}^{m} (\beta_{ji} - \alpha_{ji}) f_j(\boldsymbol{x}), \tag{6.4}$$

which can be split into *production* and *consumption* terms, according to the sign that precedes their occurence in the equation:

$$\frac{dx_i}{dt} = P_i^+(\boldsymbol{x}) - P_i^-(\boldsymbol{x}), \tag{6.5}$$

with $P_i^{+/-}(\boldsymbol{x})$ Laurent polynomials with positive coefficients:

$$\begin{aligned} P_i^+(\boldsymbol{x}) &= \sum_{\substack{1\le j\le m \\ \beta_{ji}-\alpha_{ji}>0}} (\beta_{ji}-\alpha_{ji}) f_j(\boldsymbol{x}) \\ P_i^-(\boldsymbol{x}) &= \sum_{\substack{1\le j\le m \\ \beta_{ji}-\alpha_{ji}<0}} (\alpha_{ji}-\beta_{ji}) f_j(\boldsymbol{x}) \end{aligned} \tag{6.6}$$

For convenience purposes, we will denote $P_i^{+/-}(\boldsymbol{x}) = \sum_j M_{i,j}^{+/-}(\boldsymbol{x})$, where $M_{i,j}^{+/-}(\boldsymbol{x})$ represent the production, respectively the consumption, monomials.

The reduction heuristics that use ideas from tropical analysis exploit the concept of dominance, which we borrow for our method. Let $M_1(\boldsymbol{x}) = c_1\boldsymbol{x}^{\alpha_1}$ and $M_2(\boldsymbol{x}) = c_2\boldsymbol{x}^{\alpha_2}$ be two (positive) monomials. We define ϵ-dominance as the following partial order relation on the set of multivariate monomials defined on subsets of $\mathbb{R}_+^n$:

Definition 6.2.1
(ϵ-dominance) *For an $\epsilon \in [0,1]$, we say that M_1 dominates M_2 at a time point t, denoted by $M_1 \succ_\epsilon M_2$, if $\epsilon \cdot M_1(\boldsymbol{x}(t)) \geq M_2(\boldsymbol{x}(t))$.*

In multiscale biochemical systems, the various monomials that compose the polynomials $P_i^{+/-}$ have different magnitude orders, such that at any given time there is only one or a few dominating monomials.

Definition 6.2.2
(Dominant monomial of a polynomial) *For a given $\epsilon \in [0,1]$, the dominant monomial of a polynomial $P_i(\boldsymbol{x}) = \sum_{j=1}^{n} M_{i,j}(\boldsymbol{x})$ is defined as $Dom(P_i) = \{M_{i,j} \mid \forall 1 \leq k \leq n, j \neq k, M_{i,j} \succ_\epsilon M_{i,k}\}$.*

By using the max-plus algebra idea that the sum of positive, well separated terms, can be replaced by the maximum term, each of the two polynomials of (6.5) can be replaced by their dominant monomials. The result is a reduced model, consisting of a piecewise smooth function. As the dominant monomials of the $P_i^{+/-}$ can change from one concentration domain to another, the reduced model is a piecewise-smooth *hybrid* model.

Definition 6.2.3
(Two-term tropicalization of the smooth ODE system) *We call two-term tropicalization of the smooth ODE system (4) the following piecewise-smooth system:*

$$\frac{dx_i}{dt} = Dom(P_i^+(\boldsymbol{x})) - Dom(P_i^-(\boldsymbol{x})) \tag{6.7}$$

We note that a *one-term tropicalization* of the smooth ODE system, $Dom(\frac{dx_i}{dt})$, is also possible, but choosing only one dominant momomial instead of the production-consumption pair of dominant monomials leads to a less precise model reduction (as more information is discarded in the one-term tropicalization). Thus, in this paper, we choose to deal with the two-term method.

6.2.2 Motivating example: Michaelis-Menten

Previously, we mentioned that the classical QSS[23] and QE[106, 127] approximations represent popular methods for the simplification of biochemical networks that operate on different time and concentration scales - that the Michaelis-Menten enzymatic reaction scheme being the most well-known example of such a simplification.

One of the main difficulties of applying the QSS and QE approximations to biochemical models is that both the QE reactions and the QSS species need to be specified a priori. Thus, simulation of the original model is usually [1] needed in order to detect dominated species, which are either QSS species, or participate in QE reactions [138]. For high-dimensional non-linear systems, this requirement can represent an obstacle towards model reduction.

The issue regarding simulation of the initial system also arises when trying to quantify the efficiency of model reduction methods: ideally, the approximation errors resulting from the reduction should be computed *without* executing the original system.

Thus, herein we propose an approximation method for biochemical networks, in which no prior knowledge about the original system's behavior is required. Our method combines the dominance concept with mass invariants of the original ODE system in order to compute inequality constraints on the species' concentrations. These constraints are then combined with the original system of equations, in order to obtain a reduced system of ODEs that provides time-dependent lower and upper bounds on the species' concentrations. Depending on the coarseness of detail we choose to incorporate in the mass invariant-generated inequalities, our approach can serve either as a reduction method, or to quantify the approximation errors of tropicalization reduction heuristics.

To achieve this, we abstract the original system by a box, the hyper-faces of which provide lower and upper bounds for the concentrations of the species. The two equations of the hyper-faces of a species represent simplified versions of the original differential equation of the species, in which only the dominant positive and negative monomials are considered. We refer to these equations as being *tropicalized*. Then, instead of interpreting the differential equations over the state of the original system, we will lift this interpretation conservatively over each hyper-face of the box. To do this, we will bound, for every hyper-face, the derivative of the corresponding coordinate in the solution of the original differential equation over the whole hyper-face. Our method should allow for formal evaluation of tropicalization approaches, and as such the bounds are derived using the dominance relations between monomials of the original

[1] In [139], a formal method for the identification of QSS species and QE conditions is proposed, which follows from the calculation of the tropicalized system, and which does not require simulation of trajectories. Instead, QE reactions and QSS species are detected by checking a sliding mode condition on the tropical manifold. The sliding mode condition is given in Theorem 2.5.1 in the original text [139].

ODE. Mass invariants of the original system will then be used to refine the bounds, and thus increase the accuracy of our method. By construction, the maximal solutions of the original, respectively tropicalized (*i.e.*, abstracted) equations are related by the following soundness criterion: when both defined at time t, the state of the original system is within the hyper-box of the abstract system.

Example 6.2.1
Let us consider the equations 4.3 of the Michaelis-Menten mechanism, under the assumption that $k_2 \gg k_{-1}$, i.e. $\epsilon \cdot k_{-2} \geq k_{-1} \geq 0$, for an $\epsilon \in [0,1]$. From (6.2.1), it follows that one can write (by extension): $k_2 \succ_\epsilon k_{-1}$. Then, we can deduce the following lower and upper bounds (that we call tropicalized) on the concentration of x_2:

$$\begin{cases} k_{-1}[E:S] - k_1[E][S] & \leq & \frac{d[S]}{dt} & \leq k_{-1}[E:S] - k_1[E][S] \\ k_2[E:S] - k_1[E][S] & \leq & \frac{d[E]}{dt} & \leq (1+\epsilon)k_2[E:S] - k_1[E][S] \\ k_1[E][S] - (1+\epsilon)k_2[E:S] & \leq & \frac{d[E:S]}{dt} & \leq k_1[E][S] - k_2[E:S] \\ k_2[E:S] & \leq & \frac{d[P]}{dt} & \leq k_2[E:S] \end{cases} \tag{6.8}$$

For convenience purposes, we will use the notation x_1, x_2, x_3, x_4 for the species' concentrations, $[S], [E], [E:S], [P]$. We propose to approximate the state of the system by a box of $\mathbb{R}^4$. A box of $\mathbb{R}^4$ is a set of the form $\{(x_1,x_2,x_3,x_4) \mid \underline{x_i} \leq x_i \leq \overline{x_i}, \forall 1 \leq i \leq 4\}$, where $(\underline{x_i}, \overline{x_i})$ are pairs of numbers satisfying $\underline{x_i} \leq \overline{x_i}$, $\forall 1 \leq i \leq 4$. Intuitively, the real number $\underline{x_i}$ provides a lower bound to the value of the variable x_i, and denotes the hyper-face $\{(x_1,x_2,x_3,x_4) \in \mathbb{R}^4 \mid x_i = \underline{x_i}, \underline{x_j} \leq x_j \leq \overline{x_j}, \forall 1 \leq j \leq 4, i \neq j\}$. We will denote this hyper-face as $\mathcal{F}_{\underline{x_i}}(\underline{x_1}, \overline{x_1}, \underline{x_2}, \overline{x_2}, \underline{x_3}, \overline{x_3}, \underline{x_4}, \overline{x_4})$. The other hyper-faces are defined in the same way, and the same reasoning applies to $\overline{x_i}$, which provides an upper bound to the same variable. For ease of notation, we shall use $(\underline{\boldsymbol{x}}, \overline{\boldsymbol{x}})$ to denote the vector $(\underline{x_1}, \overline{x_1}, \underline{x_2}, \overline{x_2}, \underline{x_3}, \overline{x_3}, \underline{x_4}, \overline{x_4})$.

Next, let us consider the following functions:

$$\begin{cases} F^{\#}_{\underline{x_1}}(\underline{\boldsymbol{x}}, \overline{\boldsymbol{x}}) = & k_{-1}\underline{x_3} - k_1\overline{x_2}\underline{x_1} \\ F^{\#}_{\overline{x_1}}(\underline{\boldsymbol{x}}, \overline{\boldsymbol{x}}) = & k_{-1}\overline{x_3} - k_1\underline{x_2}\overline{x_1} \\ F^{\#}_{\underline{x_2}}(\underline{\boldsymbol{x}}, \overline{\boldsymbol{x}}) = & k_2\underline{x_3} - k_1\underline{x_2}\overline{x_1} \\ F^{\#}_{\overline{x_2}}(\underline{\boldsymbol{x}}, \overline{\boldsymbol{x}}) = & (1+\epsilon)k_2\overline{x_3} - k_1\overline{x_2}\underline{x_1} \\ F^{\#}_{\underline{x_3}}(\underline{\boldsymbol{x}}, \overline{\boldsymbol{x}}) = & k_1\underline{x_1}\underline{x_2} - (1+\epsilon)k_2\underline{x_3} \\ F^{\#}_{\overline{x_3}}(\underline{\boldsymbol{x}}, \overline{\boldsymbol{x}}) = & k_1\overline{x_1 x_2} - k_2\overline{x_3} \\ F^{\#}_{\underline{x_4}}(\underline{\boldsymbol{x}}, \overline{\boldsymbol{x}}) = & k_2\underline{x_3} \\ F^{\#}_{\overline{x_4}}(\underline{\boldsymbol{x}}, \overline{\boldsymbol{x}}) = & k_2\overline{x_3} \end{cases} \tag{6.9}$$

The abstraction of the concrete system of equations is then defined as:

$$\begin{cases} \frac{d\underline{x_i}}{dt} = F^{\#}_{\underline{x_i}}(\underline{\boldsymbol{x}}, \overline{\boldsymbol{x}}) \\ \frac{d\overline{x_i}}{dt} = F^{\#}_{\overline{x_i}}(\underline{\boldsymbol{x}}, \overline{\boldsymbol{x}}) \end{cases}, \forall 1 \leq i \leq 4.$$

If we fix the same initial conditions for both the concrete and the abstracted system, $\underline{x_i}(0) = \overline{x_i}(0) = x_i(0), \forall 1 \leq i \leq 4$, we can relate the solution of the abstract system to that of the original

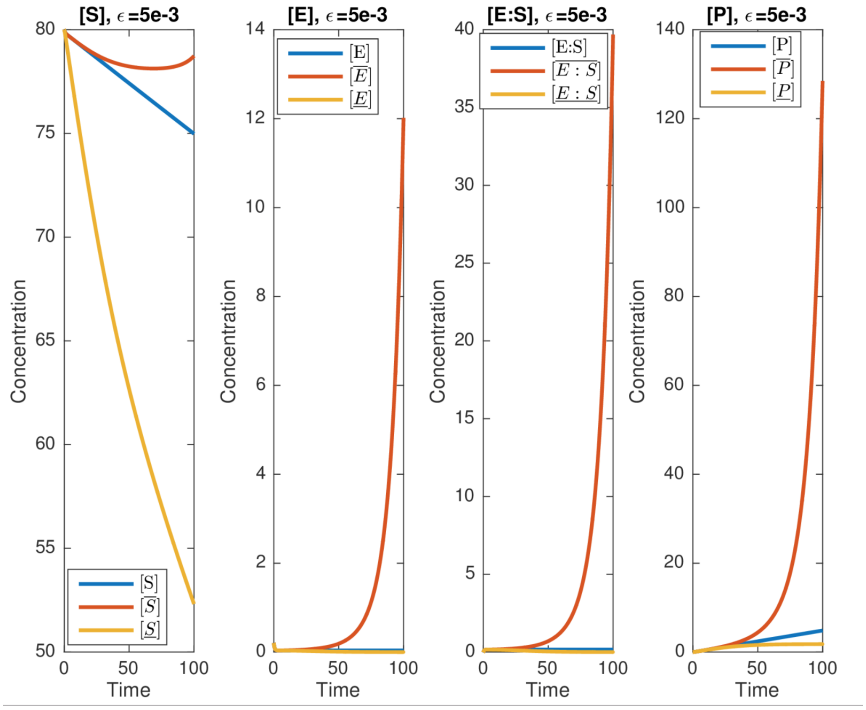


Figure 6.1: Bounds on the species' concentration with respect to simulation time, with $\epsilon = 5 \cdot 10^{-3}$, rate constants $k_1 = 0.017, k_{-1} = 0.0017, k_2 = 0.3$, and initial concentrations $[S] = 80, [E] = 0.2, [E:S] = 0, [P] = 0$ that satisfy the QSS assumption. For each of the 4 species, $[\underline{\cdot}]$ and $[\overline{\cdot}]$ denote the lower, respectively upper bounds on its concentration. The depicted results were obtained without using the mass invariants of the original system to constrain the bounds, and are consequently sub-optimal.

one. For every $1 \leq i \leq 4$, *the real number* $F^{\#}_{\underline{x_i}}(\underline{\boldsymbol{x}}, \overline{\boldsymbol{x}})$ *provides a lower bound to the value of the function* $\frac{dx_i}{dt}$ *over the hyper-face* $\mathcal{F}_{\underline{x_i}}$, *whereas the real number* $F^{\#}_{\overline{x_i}}(\underline{\boldsymbol{x}}, \overline{\boldsymbol{x}})$ *provides an upper bound to the value of the function* $\frac{dx_i}{dt}$ *over the hyper-face* $\mathcal{F}_{\overline{x_i}}$. *That is to say, we have* $\underline{\frac{dx_i}{dt}} \leq \frac{dx_i}{dt}$, *for every pair* $(x_1, x_2, x_3, x_4) \in \mathcal{F}_{\underline{x_i}}$, *and* $\frac{dx_i}{dt} \leq \overline{\frac{dx_i}{dt}}$, *for every pair* $(x_1, x_2, x_3, x_4) \in \mathcal{F}_{\overline{x_i}}$. *Then, using the results of [107], we can conclude that, for every time point* t, *and* $\forall 1 \leq i \leq 4$, *the bounds:*

$$\underline{x_i}(t) \leq x_i(t) \leq \overline{x_i}(t) \tag{6.10}$$

are satisfied. Thus, the solution of the abstract system of equations provides lower and upper bounds for the value of the variables of the original system of equations.

Remark 6.2.1. In the above example, in order to obtain safe lower/upper bounds on x_i's concentration, we make the variables range over the hyper-faces. One notices that the variable x_i is treated specifically in the derivatives of the variables $\underline{x_i}, \overline{x_i}$ - any of its occurences is replaced by the variable corresponding to the hyper-face we want to bound. By contrast, the other variables, x_j, are replaced according to the sign of their occurence:

- in $\frac{d\underline{x_i}}{dt}$, x_j is replaced with $\begin{cases} \overline{x_j}, & \text{if } x_j \text{ occurs negatively,} \\ \underline{x_j}, & \text{if } x_j \text{ occurs positively.} \end{cases}$
- in $\frac{d\overline{x_i}}{dt}$, x_j is replaced with $\begin{cases} \overline{x_j}, & \text{if } x_j \text{ occurs positively,} \\ \underline{x_j}, & \text{if } x_j \text{ occurs negatively.} \end{cases}$

This comes from the fact that the derivative on x_i is evaluated on the corresponding hyper-face, which allows for greatly reducing the loss of precision. For a formal proof of the soundness of this approach, the reader is referred to [107]. Intuitively, it is justified by the intermediate value theorem: given a family of functions $\{f_i\}$ over the real field, if one function f_i does not take the highest value at time t, whereas it is the case at time $t'' > t$, then necessarily, there exists a time t' such that $t < t' \leq t''$ in which f_i takes the highest value while crossing another function of the family.

In Fig.6.1, we show the time-evolution of the bounds on the concentration of the 4 species in the Michaelis-Menten system, for an arbitrarily chosen set of reaction rate constants and initial concentrations that satisfy the QSS condition (*i.e.*, $k_2 \succ_\epsilon k_{-1}$, and $[S] \succ_\epsilon [E] + [E:S]$ at time $t = 0$). Nonetheless, our model reduction is sound no matter the value of initial concentrations and reaction rate constants. The equations have been integrated using the solver $ode15s$ of Matlab[123]. Strictly speaking, numerical errors stemming from numerical integration may accumulate throughout the simulation, but herein we choose to ignore them.

In Fig.6.1, we notice that the bounds diverge at a fast rate from the original trajectory, despite the restriction of the derivative's evaluation on the hyper-face of the box (as explained in Remark 6.2.1). A way to improve bound accuracy is to take into account the original system's mass invariants, when computing the bounds.

In general, a biochemical system can have several conservation laws/mass invariants, which are linear functions $b_1(\boldsymbol{x}), \ldots, b_m(\boldsymbol{x})$ of the concentrations, and are constant in time. These equality constraints can be used to refine the bounds on the initial system's species' concentrations, by (safely) restricting the evaluation of the derivative of each coordinate to the intersection of the corresponding hyperface with the subspace delimited by the conservation laws containing the variable itself. Because a variable can appear in more than one mass invariant, we choose to keep the most optimistic bound that can be computed by intersecting the hyper-face with the mass invariant subspace: the greatest lower bound, respectively the smallest upper bound.

Example 6.2.2
In the Michaelis-Menten system, the total number of enzymes is constant, and so is the overall number of substrates and product. The two conservation laws can be written as:

$$\begin{cases} x_2(t) + x_3(t) = e_0 \\ x_1(t) + x_3(t) + x_4(t) = s_0 \end{cases}$$

with $e_0 = x_2(0) + x_3(0)$, *and* $s_0 = x_1(0) + x_3(0) + x_4(0)$.

Assuming once more that $k_2 \gg k_{-1}$, *by substituting* x_3 *by* $e_0 - x_2$ *or* $s_0 - x_1 - x_4$ *into 6.8, three equivalent tropicalized upper bounds on the concentration of* x_2 *are obtained:*

$$\begin{cases} \frac{dx_2}{dt} \leq (1+\epsilon)k_2x_3 - k_1x_2x_1 \\ \frac{dx_2}{dt} \leq (1+\epsilon)k_2(e_0 - x_2) - k_1x_2x_1 \\ \frac{dx_2}{dt} \leq (1+\epsilon)k_2(s_0 - x_1 - x_4) - k_1x_2x_1 \leq (1+\epsilon)k_2(s_0 - x_1) - k_1x_2x_1 \end{cases} \tag{6.11}$$

Lifting the interpretation of the differential equations over the hyper-face corresponding to $\overline{x_2}$ *results in three different expressions for the upper bound on* $\frac{dx_2}{dt}$, *of possibly different accuracies:*

$$\begin{cases} \frac{d\overline{x_{2,1}}}{dt} = (1+\epsilon)k_2\overline{x_3} - k_1\overline{x_2}\underline{x_1} \\ \frac{d\overline{x_{2,2}}}{dt} = (1+\epsilon)k_2(e_0 - \overline{x_2}) - k_1\overline{x_2}\underline{x_1} \\ \frac{d\overline{x_{2,3}}}{dt} = (1+\epsilon)k_2(s_0 - \underline{x_1}) - k_1\overline{x_2}\underline{x_1} \end{cases} \tag{6.12}$$

The most accurate sound upper bound on $\frac{dx_2}{dt}$ *then writes as:*

$$\min(\frac{d\overline{x_{2,1}}}{dt}, \frac{d\overline{x_{2,2}}}{dt}, \frac{d\overline{x_{2,3}}}{dt}) = (1+\epsilon) \cdot k_2 \cdot \min(\overline{x_3}, e_0 - \overline{x_2}, s_0 - \underline{x_1}) - k_1\overline{x_2}\underline{x_1} \tag{6.13}$$

Remark 6.2.2. The choice to introduce min and max operations in the expressions of the computed bounds is accounted for by our initial motivation: because existing tropicalization reduction heuristics are not justified by rigourous estimates, we aim to provide a method for *quantifying errors* stemming from such tropicalization reduction approaches, at the same time creating a tropicalization approach with *guarantees.*We nonetheless stress that our goal is not to correct the faults of existing tropicalization-inspired reduction methods, but rather quantify them by proposing a more rigorous tropicalization approach, in which the dominated monomials are bounded, rather than discarded from the ODEs. Consequently, we aim at computing error bounds that are as precise as possible, hence the choice of using min and max operations for bound refinement, albeit with the disadvantage of using functions that are not $\mathbb{C}^1$, thus introducing non-smooth vector fields. The trade-off between smoothness and precision can be tuned according to the desired goal: less precise bounds can be obtained by choosing to use smooth functions. Moreover, smoothness of vector fields is generally not guaranteed during the numerical simulation of biochemical models: as the model variables represent biochemical species' concentrations, a good practice is to call the numerical solvers used to approximate the system's behavior using with the 'Non-Negative' option, which amounts to introducing a max operation into the equations (*i.e.*, $\max(0, x_i)$), in order to prevent negative values of variables.

The same reasoning can be applied to all variables appearing in the expression of $\frac{dx_2}{dt}$*, in order to obtain the most accurate upper bound:*

$$\frac{d\overline{x_2}}{dt} = (1+\epsilon)\cdot k_2 \cdot \min(\overline{x_3}, e_0 - \overline{x_2}, s_0 - \underline{x_1}) - k_1 \cdot \max(\underline{x_1}, 0) \cdot \max(\overline{x_2}, e_0 - \overline{x_3}) \tag{6.14}$$

Remark 6.2.3. In (6.11), when computing the third bound, instead of substituting x_3 by its conservation law expression, $s_0 - x_1 - x_4$, we choose to bound its value by an expression not containing x_4. We do so in order to avoid introducing supplementary variables w.r.t. those present in the tropicalized original bound (*i.e.*, x_1, x_2 and x_3). This method, in which mass invariant partial refinement is introduced after the tropicalized bounds have been computed, can be considered as a *per se* model reduction method, as no supplementary information/complexity is introduced by incorporating the conservation laws. By contrast, the approach in which all the information contained by the conservation laws is exploited in order to derive the most accurate bounds can constitute a method of error-estimation for tropicalization based reduction heuristics. We present the two different methods formally in Section 3.

The issue of specifying QSS species and QE reactions a priori, when performing model reductions, is circumvented by our method. Instead, the notion of *region* is used in order to eliminate monomials from the species' ODEs. Our method uses static inspection of each ODE, in order to partition the state space into different regions according to which production, respectively consumption terms dominate the others. Using this partitioning, simplified expressions

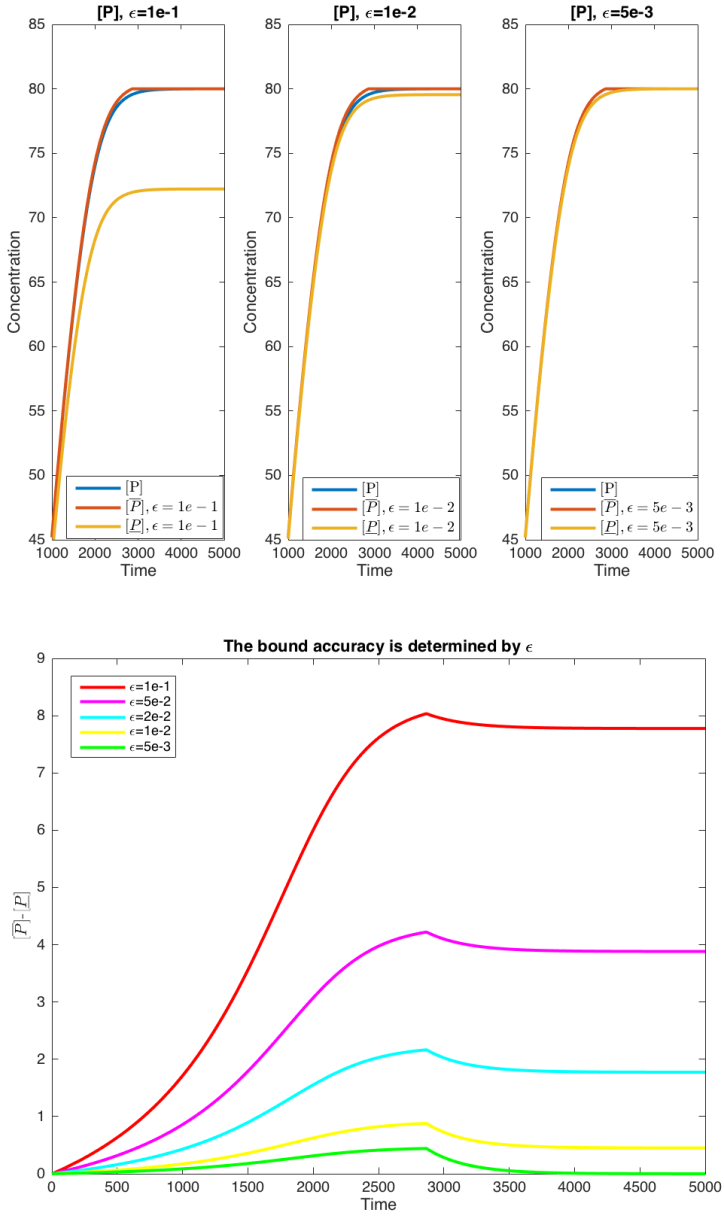


Figure 6.2: *Top*: bounds on the concentration of P, obtained by simulating the ODE system in Example6.2.4, for different values of ϵ. Rate constants and initial concentration as in Fig.6.1: $k_1 = 0.017, k_{-1} = 0.0017, k_2 = 0.3, [S] = 80, [E] = 0.2, [E : S] = 0, [P] = 0$. *Bottom*: For different values of ϵ, the accuracy of the resulting bounds is computed as the difference between the upper and the lower bound.

bounding the species concentrations are derived for each region of the state space, allowing symbolic simplification and limiting numerical approximations.

Example 6.2.3
In the case of the Michaelis-Menten mechanism, there are three possible dominance regions for $\frac{dx_2}{dt}$ *leading to three possible pairs of lower and upper bounds:*

1. *Region 1, if* k_{-1} *dominates* k_2*:*
   $\epsilon k_{-1} \geq k_2 \Rightarrow k_{-1}x_3 - k_1x_2x_1 \leq \frac{dx_2}{dt} \leq (1+\epsilon)k_{-1}x_3 - k_1x_2x_1$
2. *Region 2, if* k_2 *dominates* k_{-1}*:*
   $\epsilon k_2 \geq k_{-1} \Rightarrow k_2x_3 - k_1x_2x_1 \leq \frac{dx_2}{dt} \leq (1+\epsilon)k_2x_3 - k_1x_2x_1$
3. *Region 3, if there is no dominant rate (i.e.,* k_{-1} *and* k_2 *are of comparable magnitude):*
   $\begin{cases} \epsilon k_{-1} \leq k_2 \\ \epsilon k_2 \leq k_{-1} \end{cases} \Rightarrow (k_{-1}+k_2)x_3 - k_1x_2x_1 \leq \frac{dx_2}{dt} \leq (k_{-1}+k_2)x_3 - k_1x_2x_1$

The complete system of equations obtained using mass invariants refinement of bounds, for all the possible dominance regions, can be found in Example 6.2.4. The improvement of bound accuracy via mass invariants can be observed in Fig. 6.2. As expected, one can also observe in Fig.6.2 that results become more precise as the value of ϵ increases, *i.e.* as k_{-1} and k_2 become more separated.

Example 6.2.4
For convenience purposes, denote the species concentrations, $[S], [E], [E:S], [P]$*, using* x_1, x_2, x_3, x_4*. Then, the derivatives of the lower and upper bounds of the original system's species' concentrations write as:*

$$\frac{d\underline{x_1}}{dt} = k_{-1} \cdot \max(\underline{x_3}, e_0 - \overline{x_2}) - k_1 \cdot \min(\underline{x_1}, s_0 - \underline{x_3}) \cdot \min(\overline{x_2}, e_0 - \underline{x_3})$$

$$\frac{d\overline{x_1}}{dt} = k_{-1} \cdot \min(\overline{x_3}, e_0 - \underline{x_2}, s_0 - \overline{x_1}) - k_1 \cdot \min(\overline{x_1}, s_0 - \underline{x_3}) \cdot \max(\underline{x_2}, e_0 - \overline{x_3})$$

$$\frac{d\underline{x_2}}{dt} = \underline{c_+} \cdot \max(\underline{x_3}, e_0 - \underline{x_2}) - k_1 \cdot \min(\overline{x_1}, s_0 - \underline{x_3}) \cdot \min(\underline{x_2}, e_0 - \underline{x_3}),$$

$$\text{with } \underline{c_+} = \begin{cases} k_{-1}, & \text{if } \epsilon k_{-1} \geq k_2 \\ k_2, & \text{if } \epsilon k_2 \geq k_{-1} \\ (k_{-1}+k_2), & \text{otherwise} \end{cases}$$

$$\frac{d\overline{x_2}}{dt} = \overline{c_+} \cdot \min(\overline{x_3}, e_0 - \overline{x_2}, s_0 - \underline{x_1}) - k_1 \cdot \max(\underline{x_1}, 0) \cdot \max(\overline{x_2}, e_0 - \overline{x_3}),$$

$$\text{with } \overline{c_+} = \begin{cases} (1+\epsilon)k_{-1}, & \text{if } \epsilon k_{-1} \geq k_2 \\ (1+\epsilon)k_2, & \text{if } \epsilon k_2 \geq k_{-1} \\ (k_{-1}+k_2), & \text{otherwise} \end{cases}$$

$$\frac{d\underline{x_3}}{dt} = k_1 \cdot \max(\underline{x_1}, 0) \cdot \max(\underline{x_2}, e_0 - \underline{x_3}) - \underline{c_-} \cdot \min(\underline{x_3}, e_0 - \underline{x_2}, s_0 - \underline{x_1}),$$

$$\text{with } \underline{c_-} = \begin{cases} (1+\epsilon)k_{-1}, & \text{if } \epsilon k_{-1} \geq k_2 \\ (1+\epsilon)k_2, & \text{if } \epsilon k_2 \geq k_{-1} \\ (k_{-1}+k_2), & \text{otherwise} \end{cases}$$

$$\frac{d\overline{x_3}}{dt} = k_1 \cdot \min(\overline{x_1}, s_0 - \overline{x_3}) \cdot \min(\overline{x_2}, e_0 - \overline{x_3}) - \overline{c_-} \cdot \max(\overline{x_3}, e_0 - \overline{x_2}),$$

$$\text{with } \overline{c_-} = \begin{cases} k_{-1}, & \text{if } \epsilon k_{-1} \geq k_2 \\ k_2, & \text{if } \epsilon k_2 \geq k_{-1} \\ (k_{-1}+k_2), & \text{otherwise} \end{cases}$$

$$\frac{d\underline{x_4}}{dt} = k_2 \cdot \max(\underline{x_3}, 0)$$

$$\frac{d\overline{x_4}}{dt} = k_2 \cdot \min(\overline{x_3}, s_0 - \overline{x_4})$$

The Michaelis-Menten system represents a particular, simple case study: the choice of reaction rate constants fixes the dominance region in which the system evolves. In general, the state of a biochemical network can traverse multiple such regions, as the dominant monomials can change from one concentration domain to another. Thus, we next introduce a case study in which the dominant monomials are concentration-dependent, which in turn means that the dominance region is no longer fixed. Our method is designed with this more general situation in mind: having computed the most accurate bounds for each region of the state space partitioning, and having no information regarding the region in which the original system evolves at a given time t, our approach chooses the least accurate local bound, in order to ensure global soundness.

6.2.3 Motivating example: A DNA model

We construct a simple extension of the Michaelis-Menten system, in which the product formation reaction is catalazyed by a dimer of an enzyme M. The reaction system is given by:

$$\begin{cases} M + M \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} M_2 \\ M_2 + DNA \underset{k_{-2}}{\overset{k_2}{\rightleftharpoons}} M_2.DNA \\ M_2.DNA \xrightarrow{k_3} DNA + P \end{cases}$$

and its ODE system[2] writes as:

$$\begin{cases} \frac{dx_1}{dt} = -2k_1x_1^2 + 2k_{-1}x_2 \\ \frac{dx_2}{dt} = -k_{-1}x_2 - k_2x_2x_3 + k_{-2}x_4 + k_1x_1^2 \\ \frac{dx_3}{dt} = -k_2x_2x_3 + k_{-2}x_4 + k_3x_4 \\ \frac{dx_4}{dt} = -k_{-2}x_4 - k_3x_4 + k_2x_2x_3 \\ \frac{dx_5}{dt} = k_3x_4 \end{cases} \tag{6.15}$$

The mass invariants are given by:

$$\begin{cases} x_1 + 2x_2 + 2x_4 + 2x_5 = M_0 \\ x_3 + x_4 = DNA_0 \end{cases} \tag{6.16}$$

Dominance regions become concentration dependent: for example, the dominant positive monomial in $\frac{dx_2}{dt}$ is determined by the dominance relations between both the concentrations of x_1 and x_4, and between reaction rate constants k_1 and k_{-2}.

This DNA example will serve as a case study for the remainder of this chapter.

6.3 Model reduction using conservative numerical approximations

The guarantees of our method are a consequence of a carefully designed symbolic propagation of inequality constraints on the species' concentrations. Thus, symbolic transformations have to be applied on numerical expressions, of which we introduce a syntax and semantics. We also introduce an alternative definition of a biochemical model to that presented in Sect. 2, which is then used to define and justify our approximation method.

[2] once again, we denote the species $M, M_2, DNA, M_2.DNA, P$ by x_1, x_2, x_3, x_4, x_5

Definition 6.3.1
(Syntax of expressions) *Let $\mathcal{S}$ be a set of variables. We define an $\mathcal{S}$-expression inductively, as follows*[3]*:*

1. *each positive real number $k \in \mathbb{R}_+$ is an $\mathcal{S}$-expression;*
2. *each variable $x \in \mathcal{S}$ is an $\mathcal{S}$-expression;*
3. *if e is an $\mathcal{S}$-expression, then $(\dot{-}e)$ is an $\mathcal{S}$-expression;*
4. *if e_1 and e_2 are $\mathcal{S}$-expressions, then $(e_1 \dot{+} e_2)$, $(e_1 \dot{:} e_2)$, $\dot{\min}(e_1, e_2)$, $\dot{\max}(e_1, e_2)$ are all $\mathcal{S}$-expressions;*

The set of $\mathcal{S}$-expressions is denoted as $Expr_{\mathcal{S}}$. Given an $\mathcal{S}$-expression e, we define its support, denoted $supp(e)$, as the set of variables it contains.

Definition 6.3.2
(Semantics of expressions) *Let $\mathcal{S}$ be a set of variables and e be an $\mathcal{S}$-expression. The semantics of the expression e is the function $[\![e]\!]_{\mathcal{S}} : \mathbb{R}^{\mathcal{S}} \to \mathbb{R}$, defined inductively as follows:*

1. $\forall c \in \mathbb{R}, [\![c]\!]_{\mathcal{S}(\rho)} = c$
2. $\forall x \in \mathcal{S}, [\![x]\!]_{\mathcal{S}(\rho)} = \rho(x)$
3. $\forall e \in Expr_{\mathcal{S}}, [\![\dot{-}e]\!]_{\mathcal{S}(\rho)} = -[\![e]\!]_{\mathcal{S}(\rho)}$
4. $\forall e_1, e_2 \in Expr_{\mathcal{S}}, [\![e_1 \dot{+} e_2]\!]_{\mathcal{S}(\rho)} = [\![e_1]\!]_{\mathcal{S}(\rho)} + [\![e_2]\!]_{\mathcal{S}(\rho)}$
5. $\forall e_1, e_2 \in Expr_{\mathcal{S}}, [\![e_1 \dot{:} e_2]\!]_{\mathcal{S}(\rho)} = [\![e_1]\!]_{\mathcal{S}(\rho)} [\![e_2]\!]_{\mathcal{S}(\rho)}$
6. $\forall e_1, e_2 \in Expr_{\mathcal{S}}, [\![\dot{\min}(e_1, e_2)]\!]_{\mathcal{S}(\rho)} = \min([\![e_1]\!]_{\mathcal{S}(\rho)} [\![e_2]\!]_{\mathcal{S}(\rho)})$
7. $\forall e_1, e_2 \in Expr_{\mathcal{S}}, [\![\dot{\max}(e_1, e_2)]\!]_{\mathcal{S}(\rho)} = max([\![e_1]\!]_{\mathcal{S}(\rho)} [\![e_2]\!]_{\mathcal{S}(\rho)})$

for every environment $\rho \in \mathbb{R}^{\mathcal{S}}$.

We use Defs. 6.3.1 and 6.3.2 to define the notion of system of symbolic differential equations and symbolic equality constraints derived from conservation laws.

Definition 6.3.3
(Symbolic ODE system)

A system of symbolic ordinary differential equations and equality constraints modeling a biochemical network is a tuple $(\mathcal{S}, \mathbb{I}, \mathbb{F}, (\mathbb{E}_b))$, where:

- *$\mathcal{S} = \{x_1, \ldots, x_s\}$ is a set of variables, denoting species' concentrations,*
- *$\mathbb{I} : \mathcal{S} \to \mathbb{R}^+$ is a non-negative function, mapping each species to its initial concentration,*

[3] the syntactic operators are written using a superscript dot, in order to distinguish them from their associated mathematical functions

- $\mathbb{F} : \mathcal{S} \to Expr_{\mathcal{S}}$ *is a function describing the evolution of species' concentrations, as described in Eq.(6.5):*

$$\forall x_i \in \mathcal{S}, \mathbb{F}(x_i) = P_i^+(\boldsymbol{x}) - P_i^-(\boldsymbol{x}),$$

with $P_i^{+/-} \in Expr_{\mathcal{S}}$*, Laurent polynomials with positive coefficients,*

- $\{\mathbb{E}_b\}$[4] *is a family of functions from the set* $\mathcal{S}$ *into the set* $Expr_{\mathcal{S}}$*, denoting equality constraints derived from conservation laws, such that* $\forall f : \mathcal{S} \to \mathbb{R}^{\mathbb{R}^+}$ *satisfying*

$$\begin{cases} f(x_i)(0) = \mathbb{I}(0), & \forall x_i \in \mathcal{S} \\ \frac{df(x_i)}{dt}(t) = [\![\mathbb{F}(x_i)]\!]_{\mathcal{S}[x_i \mapsto f(x_i)(t)]}, & \forall x_i \in \mathcal{S} \text{ and } t \in \mathbb{R}^+ \end{cases}$$

the constraint

$$f(x_i)(t) = [\![\mathbb{E}_b(x_i)]\!]_{\mathcal{S}[x_i \mapsto f(x_i)(t)]}$$

is satisfied for every function $\mathbb{E}_b$ *of the family* $\{\mathbb{E}_b\}$*,* $\forall x_i \in \mathcal{S}$*, and for every time* $t \in \mathbb{R}^+$*.*

Example 6.3.1
(A DNA example) *In our running example,* $\mathcal{S} = \{x_1, x_2, x_3, x_4\}$*,* $\mathbb{F}$ *is defined by the equations of (6.15), and the equality constraints derived from the conservation laws of (6.16) write:*

$$\begin{cases} \mathbb{E}_1(x_1) = M_0 - 2x_2 - 2x_4 - 2x_5; & \mathbb{E}_2(x_1) = M_0 - 2DNA_0 - 2x_2 + 2x_4 - 2x_5 \\ \mathbb{E}_1(x_2) = \frac{M_0 - x_1}{2} - x_4 - x_5; & \mathbb{E}_2(x_2) = \frac{M_0 - x_1}{2} - DNA_0 + x_3 - x_5 \\ \mathbb{E}_1(x_3) = DNA_0 - x_4; & \mathbb{E}_2(x_3) = DNA_0 - \frac{M_0 - x_1}{2} + x_2 + x_5; \\ \mathbb{E}_1(x_4) = DNA_0 - x_3; & \mathbb{E}_2(x_4) = \frac{M_0 - x_1}{2} - x_2 - x_5 \\ \mathbb{E}_1(x_5) = \frac{M_0 - x_1}{2} - x_2 - x_4; & \mathbb{E}_2(x_5) = \frac{M_0 - x_1}{2} - DNA_0 + x_3 - x_2 \end{cases} \tag{6.17}$$

We partition the state space of each ODE into regions, each one defined by the corresponding pair of dominant monomials, $(Dom(P_i^+(\mathbf{x})), Dom(P_i^-(\mathbf{x})))$. At any given time t, several monomials can be dominant, which can lead to an exponential number of possible regions. To circumvent this issue and obtain a linear number of regions, we choose to replace each region that has more than one dominant term with the unique region in which no term is dominant: if $|Dom(P_i^{\pm}(\mathbf{x}))| > 1$, we choose to keep $P_i^{\pm}(\mathbf{x})$ in the reduced ODE, instead of replacing it with $Dom(P_i^{\pm}(\mathbf{x}))$. The following definition formalizes these concepts.

Definition 6.3.4
(State partitioning of a symbolic ODE) *Let* $(\mathcal{S}, \mathbb{I}, \mathbb{F}, \{\mathbb{E}_b\})$ *be a symbolic ODE system, and* $\epsilon \in [0, 1]$ *a scale separation constant. Then, for every variable* $x_i \in \mathcal{S}$*, if* $P_i^+ = \sum_{j=1}^{p} M_j^+$ *and* $P_i^- = \sum_{j=1}^{n} M_j^-$*, its state space can be partitioned into* $(p+1) \times (n+1)$ *regions, each one determined by the corresponding pair of dominant monomials*

[4] the number b indexes the different ways of expressing a species x_i, by using the mass invariants in which it appears

$$r_i^{k,l} := \begin{cases} (M_k^+, M_l^-), & \text{if } k \leq p, l \leq n, Dom(P_i^+) = M_k^+, Dom(P_i^-) = M_l^-, \\ (P_i^+, M_l^-), & \text{if } k = p+1, l \leq n, Dom(P_i^-) = M_l^- \\ (M_k^+, P_i^-), & \text{if } k \leq p, l = n+1, Dom(P_i^+) = M_k^+ \\ (P_i^+, P_i^-), & \text{if } k = p+1, l = n+1 \end{cases} \tag{6.18}$$

Example 6.3.2
(A DNA example) *In Eq.(6.15), the state space of x_2 can be partitioned in 9 regions, as its ODE contains 2 positive terms and 2 negative terms:*

$$\begin{aligned} r_2^{1,1} &= (k_1 x_1^2, k_{-1} x_2); \\ r_2^{2,1} &= (k_{-2} x_4, k_{-1} x_2); \\ r_2^{3,1} &= (k_1 x_1^2 + k_{-2} x_4, k_{-1} x_2); \\ r_2^{1,2} &= (k_1 x_1^2, k_2 x_2 x_3); \\ r_2^{2,2} &= (k_{-2} x_4, k_2 x_2 x_3); \\ r_2^{3,2} &= (k_1 x_1^2 + k_{-2} x_4, k_2 x_2 x_3); \\ r_2^{1,3} &= (k_1 x_1^2, k_{-1} x_2 + k_2 x_2 x_3); \\ r_2^{2,3} &= (k_{-2} x_4, k_{-1} x_2 + k_2 x_2 x_3); \\ r_2^{3,3} &= (k_1 x_1^2 + k_{-2} x_4, k_{-1} x_2 + k_2 x_2 x_3) \end{aligned}$$

We next use the dominance relations that define each region, in order to obtain region-specific lower and upper bounds on the ODE being considered. The next definition formalizes this procedure:

Definition 6.3.5
(Region-specific tropicalized bounds) *Given a symbolic ODE system $(\mathcal{S}, \mathbb{I}, \mathbb{F}, (\mathbb{E}_b))$, and the set of regions $r_i^{k,l}$ for each species x_i, the dominance definition 6.2.1 can be used to define the following functions, for every region $r_i^{k,l}$:*

$$\mathbb{F}_{\downarrow}^{k,l}(x_i) := \begin{cases} M_k^+ \dot{-} (1 \dot{+} (n \dot{-} 1) \dot{:} \epsilon) \dot{:} M_l^-, & \text{if } k \leq p, l \leq n, Dom(P_i^+) = M_k^+, Dom(P_i^-) = M_l^- \\ P_i^+ \dot{-} (1 \dot{+} (n \dot{-} 1) \dot{:} \epsilon) \dot{:} M_l^-, & \text{if } k = p+1, l \leq n, Dom(P_i^-) = M_l^- \\ M_k^+ \dot{-} P_i^-, & \text{if } k \leq p, l = n+1, Dom(P_i^+) = M_k^+ \\ P_i^+ \dot{-} P_i -, & \text{if } k = p+1, l = n+1 \end{cases}$$

$$\mathbb{F}_{\uparrow}^{k,l}(x_i) := \begin{cases} (1 \dot{+} (p \dot{-} 1) \dot{:} \epsilon) \dot{:} M_k^+ \dot{-} M_l^-, & \text{if } k \leq p, l \leq n, Dom(P_i^+) = M_k^+, Dom(P_i^-) = M_l^- \\ P_i^+ \dot{-} M_l^-, & \text{if } k = p+1, l \leq n, Dom(P_i^-) = M_l^- \\ (1 \dot{+} (p \dot{-} 1) \dot{:} \epsilon) \dot{:} M_k^+ \dot{-} P_i^-, & \text{if } k \leq p, l = n+1, Dom(P_i^+) = M_k^+ \\ P_i^+ \dot{-} P_i -, & \text{if } k = p+1, l = n+1 \end{cases}$$

Functions $\mathbb{F}_{\downarrow}^{k,l}$ and $\mathbb{F}_{\uparrow}^{k,l}$ provide symbolic tropicalized lower, resp. upper bounds for $\mathbb{F}(x_i)$ on region $r_i^{k,l}$.

Example 6.3.3
(A DNA example) *In our running example, in region $r_2^{2,1} = (k_{-2}x_4, k_{-1}x_2)$, the dominant positive (production) monomial is $k_{-2}x_4$, and the dominant negative (consumption) monomial is $k_{-1}x_2$. Formally, this writes as $\epsilon \cdot k_{-2}x_4 \geq k_1 x_1^2 \geq 0$, and $\epsilon \cdot k_{-1}x_2 \geq k_2 x_2 x_3 \geq 0$.*

Thus, the $r_2^{2,1}$ specific tropicalized bounds write as:

$$\begin{cases} \mathbb{F}_{\downarrow}^{2,1}(x_2) = k_{-2}x_4 - (1+\epsilon)k_{-1}x_2 \\ \mathbb{F}_{\uparrow}^{2,1}(x_2) = (1+\epsilon)k_{-2}x_4 - k_{-1}x_2 \end{cases},$$

which by construction satisfy $\mathbb{F}_{\downarrow}^{2,1}(x_2) \leq \frac{dx_2}{dt} \leq \mathbb{F}_{\uparrow}^{2,1}(x_2)$.

The bounds of Def. 6.3.5 can further be refined by using the mass invariants given by the family of functions $\{\mathbb{E}_b\}$, as follows:

Definition 6.3.6
(Region-specific refined tropicalized bounds) *Given a symbolic ODE system $(\mathcal{S}, \mathbb{I}, \mathbb{F}, (\mathbb{E}_b))$, the set of regions $r_i^{k,l}$ and the symbolic tropicalized bounds $\mathbb{F}_{\downarrow}^{k,l}(x_i)$, $\mathbb{F}_{\uparrow}^{k,l}(x_i)$ for each species x_i, we define the following bounds:*

$$\forall r_i^{k,l}, \forall x_j \in \mathcal{V}, \begin{cases} \mathbb{L}_{i,b}^{k,l}(x_j) := \begin{cases} \mathbb{E}_b(x_j), & \text{if } \mathcal{V} = \mathcal{V}_b \\ 0, & \text{otherwise} \end{cases} \\ \mathbb{U}_{i,b}^{k,l}(x_j) := \begin{cases} \mathbb{E}_b(x_j), & \text{if } \mathcal{V} = \mathcal{V}_b \\ [\![E_b(x_j)]\!]_{\mathcal{V}_b \setminus \mathcal{V}[x_j \mapsto b_i^{k,l}(x_j)]}, & \text{otherwise} \end{cases} \end{cases}$$

with $\mathcal{V} = supp(\mathbb{F}_{\downarrow}^{k,l}(x_i)) = supp(\mathbb{F}_{\uparrow}^{k,l}(x_i))$, $\mathcal{V}_b = supp(E_b(x_j))$, for each function $\mathbb{E}_b$ of the family $(\mathbb{E}_b)$ that applies to the variable x_j, and $b_i^{k,l}(x_j) \in Expr_{\mathcal{V}}$ is either 0, or a bound generated by the dominating monomial inequality constraints.

Remark 6.3.1. The recipe of Definition 6.3.6 for defining bounds on the concentration of a species x_i is associated to the use of our method as a **model reduction** technique.

Consequently, when considering the refinement of bounds via mass invariants, we are careful as to not introduce supplementary variables w.r.t. those found in the support of the dominant monomials - this explains why we choose to introduce a refining bound $\mathbb{E}_b(x_j)$ in its entirety, only if its support is identical to that of the corresponding tropicalized bound, *i.e.*, if $\mathcal{V} = \mathcal{V}_b$.

Otherwise, we approximate the bound $\mathbb{E}_b$ by an expression that does not introduce supplementary variables: 0 for lower bounds, and an over-approximation $\mathbb{E}'_b > \mathbb{E}_b$ that satisfies $\mathcal{V}'_b = \mathcal{V}$ for upper bounds. This latter over-approximation is obtained by replacing variables $x_j \in \mathcal{V}_b \setminus \mathcal{V}$ with either 0 (as variables x_j appear in mass-invariant-derived constraints with negative polarity, replacing x_j by 0 will yield an upper bound greater than $\mathbb{E}_b$), or with one of its bounds derived from dominating monomial inequality constraints.

In Section 6.4, however, one will be interested in using our method for estimating the approximation error of tropicalization reduction techniques. Thus, we will trade model size for precision, and allow for introduction of supplementary variables. The definition of region-specific tropicalized bounds will then simply be:

$$\forall r_i^{k,l}, \forall x_j \in \mathcal{V}, \begin{cases} \mathbb{L}_{i,b}^{k,l}(x_j) := \mathbb{E}_b(x_j), \\ \mathbb{U}_{i,b}^{k,l}(x_j) := \mathbb{E}_b(x_j) \end{cases}$$

Example 6.3.4
(A DNA example)

When dealing with the tropicalized bounds of Ex.(6.3.3), one needs to refine the bounds of the variables in their support: $\mathcal{V} = \{x_2, x_4\}$. *We do so by using their respective equality constraints from (6.17):* $\mathbb{E}_1(x_2), \mathbb{E}_2(x_2), \mathbb{E}_1(x_4)$, *and* $\mathbb{E}_2(x_4)$.

What's more, the second dominance inequality of region $r_2^{2,1}$ *in Ex.(6.3.3)can be rewritten as* $x_3 \leq \epsilon \frac{k_{-1}}{k_2}$. *This allows for a new upper bound on variable* x_3*:* $b_2^{2,1}(x_3) = \epsilon \frac{k_{-1}}{k_2} \in Expr_{\mathcal{V}}$.

Using Def.6.3.6, the $r_2^{2,1}$*-specific bounds on* x_2 *and* x_4 *write as:*

$$\begin{aligned}
&\mathbb{L}_{2,1}^{2,1}(x_2) = 0; \\
&\mathbb{L}_{2,2}^{2,1}(x_2) = 0; \\
&\mathbb{L}_{2,1}^{2,1}(x_4) = DNA_0 - \epsilon \frac{k_{-1}}{k_2}; \\
&\mathbb{L}_{2,2}^{2,1}(x_4) = 0 \\
&\mathbb{U}_{2,1}^{2,1}(x_2) = \frac{M_0}{2} - x_4; \\
&\mathbb{U}_{2,2}^{2,1}(x_2) = \frac{M_0}{2} - DNA_0 + \epsilon \frac{k_{-1}}{k_2}; \\
&\mathbb{U}_{2,1}^{2,1}(x_4) = DNA_0; \\
&\mathbb{U}_{2,2}^{2,1}(x_4) = \frac{M_0}{2} - x_2
\end{aligned}$$

Using mass invariants to compute the most optimistic bound is done inductively over the $\mathcal{S}$ expressions of the candidate bounds, by applying usual formulae of interval arithmetics [5] to

[5] the more complicated case is that of multiplication, in which every combination of lower/upper bounds

propagate the $\dot{\min}$ and $\dot{max}$ operators. The resulting evaluation functions, which we call $f_{\dot{\min}}$ and $f_{\dot{\max}}$ respectively, are defined by mutual induction over the syntax of the $\mathcal{S}$-expressions denoting monomials:

1. $\forall e_1, e_2, \ldots, e_k \in Expr_{\mathcal{S}}$,
   - $f_{\dot{\min}}(e_1, e_2, \ldots, e_k) \equiv \dot{\min}(e_1, e_2, \ldots, e_k)$
   - $f_{\dot{\max}}(e_1, e_2, \ldots, e_k) \equiv \dot{\max}(e_1, e_2, \ldots, e_k)$
2. $\forall e_1, e_2 \in Expr_{\mathcal{S}}$,
   - $f_{\dot{\min}}(\dot{-}e_1, \dot{-}e_2) \equiv \dot{-}(f_{\dot{\max}}(e_1, e_2))$
   - $f_{\dot{\max}}(\dot{-}e_1, \dot{-}e_2) \equiv \dot{-}(f_{\dot{\min}}(e_1, e_2))$
3. $\forall e_1, e_2 \in Expr_{\mathcal{S}}, \forall c \in \mathbb{R}$,
   - $f_{\dot{\min}}(c \dot{\cdot} e_1, c \dot{\cdot} e_2) \equiv \begin{cases} c \dot{\cdot} f_{\dot{\min}}(e_1, e_2), & \text{if } c \geq 0 \\ c \dot{\cdot} f_{\dot{\max}}(e_1, e_2), & \text{if } c < 0 \end{cases}$
   - $f_{\dot{\max}}(c \dot{\cdot} e_1, c \dot{\cdot} e_2) \equiv \begin{cases} c \dot{\cdot} f_{\dot{\max}}(e_1, e_2), & \text{if } c \geq 0 \\ c \dot{\cdot} f_{\dot{\min}}(e_1, e_2), & \text{if } c < 0 \end{cases}$
4. $\forall e, e_1, e_2 \in Expr_{\mathcal{S}}$,
   - $f_{\dot{\min}}(e_1 \dot{\pm} e, e_2 \dot{\pm} e) \equiv f_{\dot{\min}}(e_1, e_2) \dot{\pm} e$
   - $f_{\dot{\max}}(e_1 \dot{\pm} e, e_2 \dot{\pm} e) \equiv f_{\dot{\max}}(e_1, e_2) \dot{\pm} e$
5. $\forall e, e_1, e_2 \in Expr_{\mathcal{S}}$,
   - $f_{\dot{\min}}(e \dot{-} e_1, e \dot{-} e_2) \equiv e \dot{-} f_{\dot{\max}}(e_1, e_2)$
   - $f_{\dot{\max}}(e \dot{-} e_1, e \dot{-} e_2) \equiv e \dot{-} f_{\dot{\min}}(e_1, e_2)$

may provide the lower or the upper bound of the multiplication result; in our case, this issue is nonetheless circumvented, since variables denote concentrations, which are always positive

With all this in place, we can proceed to the definition of the reduced system.

Definition 6.3.7
(Reduced system) *Let $\mathcal{A} = (\mathcal{S}, \mathbb{I}, \mathbb{F}, (\mathbb{E}_b))$ be a system of ordinary equations with equality constraints. The reduction of the system $\mathcal{A}$ is defined as the triple $(\mathcal{S}^\#, \mathbb{I}^\#, \mathbb{F}^\#)$, with:*

1. $\mathcal{S}^\# = \{\underline{x_i} \mid x_i \in \mathcal{S}\} \cup \{\overline{x_i} \mid x_i \in \mathcal{S}\}$
2. $\mathbb{I}^\# : \mathcal{S}^\# \to \mathbb{R}^+$ *is defined by* $\mathbb{I}^\#(\underline{x_i}) = \mathbb{I}^\#(\overline{x_i}) = \mathbb{I}(x_i), \forall x_i \in \mathcal{S}$ *h*
3. $\mathbb{F}^\# : \mathcal{S}^\# \to Expr_{\mathcal{S}^\#}$, *defined as:*

$$\begin{cases} \mathbb{F}^\#(\underline{x_i}) = f_{\min}([[\mathbb{F}^{1,1}_{\downarrow}(x_i)]_{\rho_1^\downarrow}]_{\rho_2^\downarrow}, \ldots, [[\mathbb{F}^{p+1,n+1}_{\downarrow}(x_i)]_{\rho_1^\downarrow}]_{\rho_2^\downarrow}) \\ \mathbb{F}^\#(\overline{x_i}) = f_{\max}([[\mathbb{F}^{1,1}_{\uparrow}(x_i)]_{\rho_1^\uparrow}]_{\rho_2^\uparrow}, \ldots, [[\mathbb{F}^{p+1,n+1}_{\uparrow}(x_i)]_{\rho_1^\uparrow}]_{\rho_2^\uparrow}) \end{cases}$$

for every variable $x_i \in \mathcal{S}'$, where:

- $\rho_1^\downarrow = \begin{cases} x_j \mapsto \dot{\max}(x_j, \dot{\max}_b(\mathbb{L}^{k,l}_{i,b}(x_j))), & \text{if } x_j \in t^{k,l,i}_{\downarrow,+} \\ x_j \mapsto \dot{\min}(x_j, \dot{\min}_b(\mathbb{U}^{k,l}_{i,b}(x_j))), & \text{if } x_j \in t^{k,l,i}_{\downarrow,-} \end{cases}$
- $\rho_2^\downarrow = \begin{cases} x_j \mapsto \underline{x_j}, & \text{if } x_i = x_j \\ x_j \mapsto \underline{x_j}, & \text{if } x_i \neq x_j, \text{ for positive polarity/sign occurences of } x_j \\ x_j \mapsto \overline{x_j}, & \text{if } x_i \neq x_j, \text{ for negative polarity/sign occurences of } x_j \end{cases}$
- $\rho_1^\uparrow = \begin{cases} x_j \mapsto \dot{\min}(x_j, \dot{\min}_b(\mathbb{U}^{k,l}_{i,b}(x_j))), & \text{if } x_j \in t^{k,l,i}_{\uparrow,+} \\ x_j \mapsto \dot{\max}(x_j, \dot{\max}_b(\mathbb{L}^{k,l}_{i,b}(x_j))), & \text{if } x_j \in t^{k,l,i}_{\uparrow,-} \end{cases}$
- $\rho_2^\uparrow = \begin{cases} x_j \mapsto \overline{x_j}, & \text{if } x_i = x_j \\ x_j \mapsto \overline{x_j}, & \text{if } x_i \neq x_j, \text{ for positive polarity/sign occurences of } x_j \\ x_j \mapsto \underline{x_j}, & \text{if } x_i \neq x_j, \text{ for negative polarity/sign occurences of } x_j \end{cases}$

Intuitively, for each region (k, l) of species x_i, the reduction method first replaces $\mathbb{F}(x_i)$ by the pair of tropicalized lower and upper bounds, $\mathbb{F}^{k,l}_{\downarrow}(x_i)$ and $\mathbb{F}^{k,l}_{\uparrow}(x_i)$, that result directly from the dominance inequalities that characterize the region. Then, $\mathbb{F}^{k,l}_{\downarrow}(x_i)$ and $\mathbb{F}^{k,l}_{\uparrow}(x_i)$ are refined, using the bounds on variables that can be deduced from the conservation laws of the original system. For example, replacing any occurence of a variable x_j in $\mathbb{F}^{k,l}_{\downarrow}(x_i)$ with one of its expressions $\mathbb{E}_b(x_j)$ (or with its appropriate bound derived from $\mathbb{E}_b(x_j)$[6]) results in another safe upper bound for $\mathbb{F}(x_i)$. By choosing the minimum such candidate bound, one obtains the most accurate, *locally* safe upper bound. The same reasoning applies to the computation of lower bounds, but the min operation is replaced with *max*.

In order to obtain safe (*i.e.*, correct) global bounds, the least precise local bounds are chosen: the miminal lower, resp. the maximal upper bounds.

[6] $\mathbb{L}^{k,l}_{i,b}(x_j)$ for the positive occurences of x_j, and $\mathbb{U}^{k,l}_{i,b}(x_j)$ for its negative occurences

Finally, the interpretation of the variables is lifted over the hyper-faces. Any occurence of x_i is replaced with its analogue corresponding to the hyperface we want to bound, while the others are replaced to their analogue given by the polarity of their occurence, as explained in Note 6.2.1.

Theorem 6.3.1
Let $\mathcal{A} = (\mathcal{S}, \mathbb{I}, \mathbb{F}, (\mathbb{E}_b))$ be a system of ordinary equations with equality constraints. Let $(\mathcal{S}^\#, \mathbb{I}^\#, \mathbb{F}^\#)$ be a reduction of the system $\mathcal{A}$.

Let f be a function from the set $\mathcal{S}$ into the set $\mathbb{R}^{\mathbb{R}^+}$ s.t. for every variable $x_i \in \mathcal{S}$, we have:

$$\begin{cases} f(x_i)(0) = \mathbb{I}(x_i) \\ \frac{df(x_i)}{dt}(t) = \mathbb{F}[x_i \mapsto f(x_i)(t)] \end{cases}$$

and $f^\#$ be a function from the set $\mathcal{S}^\#$ into the set $\mathbb{R}^{\mathbb{R}^+}$ s.t. for every abstract variable $x_i^\# \in \mathcal{S}^\#$, we have:

$$\begin{cases} f(x_i^\#)(0) = \mathbb{I}^\#(x_i^\#) \\ \frac{df^\#(x_i^\#)}{dt}(t) = \mathbb{F}^\#[x_i^\# \mapsto f^\#(x_i^\#)(t)] \end{cases}$$

Under these assumptions, we have that for every variable $x_i \in \mathcal{S}$ and every time $t \in \mathbb{R}^+$:

$$f^\#(\underline{x_i})(t)) \leq f(x_i)(t) \leq f^\#(\overline{x_i})(t),$$

i.e., the reduced system provides sound lower and upper bounds for the concentration of the original system's species.

Proof. The proof of this theorem essentially lays in the following result of [125].

Assume the initial value problem:

$$\begin{cases} \frac{d\boldsymbol{x}}{dt} = \mathrm{F}(t, \boldsymbol{x}) \\ \boldsymbol{x}(a) = \boldsymbol{x}_a \end{cases} \tag{6.19}$$

and $\underline{\boldsymbol{x}}, \overline{\boldsymbol{x}}$ the solutions of the system of differential inequalities:

$$\begin{cases} \frac{d\underline{\boldsymbol{x}}}{dt} \leq \underline{\mathrm{F}}(t, \underline{\boldsymbol{x}}, \overline{\boldsymbol{x}}) \\ \frac{d\overline{\boldsymbol{x}}}{dt} \geq \overline{\mathrm{F}}(t, \underline{\boldsymbol{x}}, \overline{\boldsymbol{x}}) \end{cases} \tag{6.20}$$

with $\underline{\mathrm{F}}$ and $\overline{\mathrm{F}}$ defined as:

$$\begin{cases} \underline{\mathrm{F}}_i(t, \underline{\boldsymbol{x}}, \overline{\boldsymbol{x}}) = \inf\limits_{\theta} \mathrm{F}_i(t, \theta), \text{ when } \theta_i = \underline{x}_i, \text{ and } \underline{x}_j \geq \theta_j \leq \overline{x}_j, \forall j \neq i \\ \overline{\mathrm{F}}_i(t, \underline{\boldsymbol{x}}, \overline{\boldsymbol{x}}) = \sup\limits_{\theta} \mathrm{F}_i(t, \theta), \text{ when } \theta_i = \overline{x}_i, \text{ and } \underline{x}_j \geq \theta_j \leq \overline{x}_j, \forall j \neq i \end{cases} \tag{6.21}$$

Then, if

$$\underline{\boldsymbol{x}}(a) \leq \boldsymbol{x}(a) \leq \overline{\boldsymbol{x}}(a), \tag{6.22}$$

the solutions of (6.20) provide lower and upper bounds on the solution of the initial value problem of (6.19) on the interval (a, b):

$$\underline{\boldsymbol{x}}(t) \leq \boldsymbol{x}(t) \leq \overline{\boldsymbol{x}}(t), \forall t \in (a, b). \tag{6.23}$$

With this result in place (its detailed proof can be found in [125]), our proof is immediate.

Indeed, it can be easily seen that the functions $\mathbb{F}^{\#}$ that define the reduced system of Definition 6.3.7 verify conditions (6.20) and (6.21) by construction (remember that $\mathbb{F}^{\#}(\underline{x_i}) \equiv f_{\min}(\ldots)$,$\mathbb{F}^{\#}(\overline{x_i}) \equiv f_{\max}(\ldots)$), which in turn means that the inequalities of (6.23) hold for all times, *i.e.,* the reduced system provides sound lower and upper bounds for the concentration of the original system's species.

■

Example 6.3.5

We apply our method on the DNA example constructed in Sect.6.2.3, for different values of the scale separating constant ϵ, and for arbitrarily chosen reaction rate constants $k_1 = k_2 = k_3 = 0.1, k_{-1} = 0.01, k_{-2} = 0.00001$ and initial concentrations $[M]_0 = 1, [DNA]_0 = 0.05$. We show in Fig.6.3 the time evolution of the bounds on the concentration of the product species P, i.e. the variable x_5. We notice once again that the results become more precise as ϵ decreases, i.e. as the monomials become more separated. As an example, in Appendix B we detail the equation for the lower bound on the concentration of species M_2 (i.e., x_2).

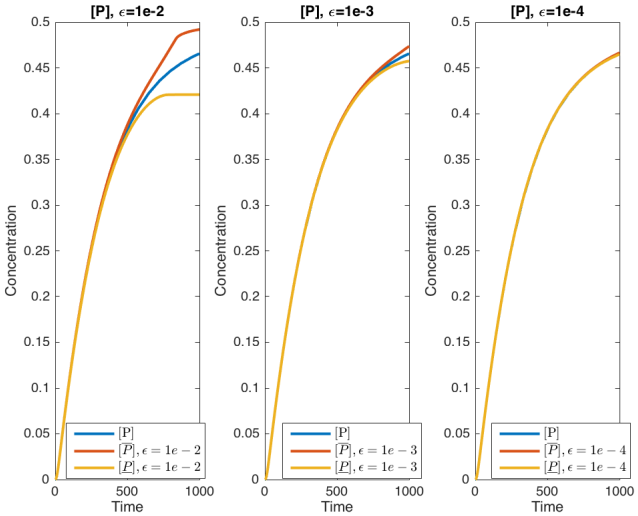


Figure 6.3: Bounds on the concentration of P, in the DNA example, obtained by simulating the ODE of tropicalized bounds for different values of the scale separation constant ϵ, rate constants $k_1 = k_2 = k_3 = 0.1, k_{-1} = 0.01, k_{-2} = 0.00001$ and initial concentrations $[M]_0 = 1, [DNA]_0 = 0.05$.

6.4 Error estimates of tropicalized systems using conservative numerical approximations

Our approach also serves as a heuristics for quantifying errors of tropicalization approaches for biochemical model reduction, provided a slight modification is applied to Def.6.3.6. Instead of computing bounds using only variables from the support of the tropicalized bounds, one can use the equality constraints $\{E_b\}$, to refine the accuracy of bounds, cf. Remark 6.3.1. The resulting model presents a trade-off: it introduces new variables w.r.t. the support of the tropicalized bounds, albeit exclusively in the form of conservation laws which are always linear functions, but gains in bound accuracy. Then, the approximation error/accuracy of a given reduction method can be assessed by checking if the reduced trajectory lies between the lower and upper bounds computed by our method.

Example 6.4.1
It is well known that the Michaelis-Menten reduction is valid only under the QSS or QE assumptions. In Fig.6.4, we simulate the reduced Michaelis-Menten system (4.5), as well as our modified reduced system, as presented above, for a set of initial conditions that no longer satisfy neither the QSS, nor the QE assumptions, i.e. the total enzyme concentration is comparable to the initial substrate concentration, and the complex dissociation reaction is significantly slower than product formation. As expected, the reduced Michaelis-Menten system no longer represents a good approximation of the initial enzymatic system (1.2.1); this is reflected by the fact that the trajectory of the reduced model does not lie between the lower and upper bounds computed by our approach.

6.4.1 Tyson's Cell Cycle Model

The tropicalization heuristics can be difficult to justify by rigourous estimates, although this is possible in some cases[138]. For example, the existence of tropical varieties - the set of points $x \in \mathbb{R}^n$ where at least two monomials of $P^{-/+}$ are equal- can lead to sliding modes, which in turn represent challenges in providing accuracy justifications for hybrid models obtained using tropical ideas. Sliding modes are well known phenomena in ODEs with discontinuous vector fields, in which the dynamics can follow discontinuity hyper-surfaces where the vector field is not defined; what's more, the conditions for the existence of sliding modes are usually intricate. As noted in [139], sliding modes can have a nefarious effect on the behavior of the tropicalized system: tropical varieties (*i.e.* tropical curves) decompose the state space into sectors corresponding to the smooth modes of the hybrid tropicalized system, which passes from one type of smooth dynamics to another intrinsically, when the trajectory attains the tropical curve. However, if certain conditions w.r.t. the sliding modes are fulfilled, the trajectory can continue along some tropical curve instead of changing sector, which further deviates the reduced system's trajectory from the original one (see Figure 1 in [139], for an example).

In [139], such phenomena become apparent when tropicalization is applied to the minimal cell cycle model proposed by Tyson[178], in order to obtain a reduced hybrid model. The Tyson model describes the interplay between cyclin and cyclin dependent kinase cdc2 during

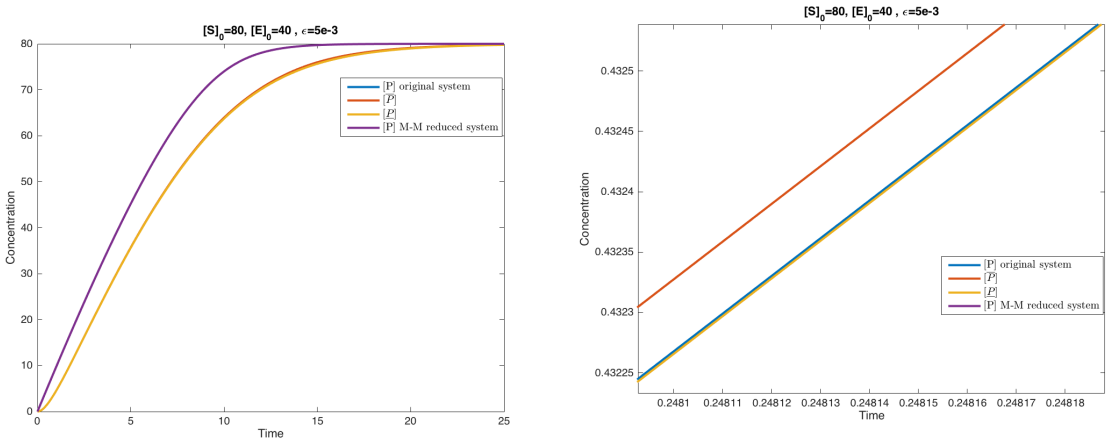


Figure 6.4: Estimating the accuracy of the Michaelis-Menten approximation: bounds on the concentration of [P], with respect to simulation time, for $\epsilon = 5 \cdot 10^{-4}$, rate constants $k_1 = 0.017, k_{-1} = 0.0017, k_2 = 0.3$, and initial concentrations that do not satisfy the QSS condition: $[S] = 80, [E] = 40, [E : S] = 0, [P] = 0$. (left) Whereas the original system's trajectory lies between the lower and upper bound given by our method, this is not the case for the classical Michaelis-Menten approximation. Thus, as expected, one can conclude that if neither the QSS nor the QE conditions are met, the Michaelis-Menten approximation is inaccurate. (right) Zoomed in version, showing the enclosed original trajectory (in blue)

the progression of the cell cycle, and demonstrates the existence of three possible regimes, that can be associated to different phases in the cell life: the biochemical system can either function as an oscillator, converge to a steady state, or behave as an excitable switch. The three possible behaviours can be associated to early embryos rapid division, arrest of unfertilised eggs and growth controlled division of somatic cells, respectively.

The dynamics of this non-linear model with rational reaction rates contains 6 species and 9 reactions, and is described by the following system of polynomial differential equations:

$$\begin{cases} \frac{dy_1}{dt} = k_6 y_4 - k_8 y_1 + k_9 y_2 \\ \frac{dy_2}{dt} = -k_3 y_2 y_5 + k_8 y_1 - k_9 y_2 \\ \frac{dy_3}{dt} = k_3 y_2 y_5 - k_4' y_3 - k_4 y_4^2 y_3 \\ \frac{dy_4}{dt} = k_4' y_3 + k_4 y_4^2 y_3 - k_6 y_4 \\ \frac{dy_5}{dt} = k_1 - k_3 y_2 y_5 \\ \frac{dy_6}{dt} = k_6 y_4 - k_7 y_6 \end{cases}, \tag{6.24}$$

and has the conservation law $y_1(t) + y_2(t) + y_3(t) + y_4(t) = 1$, where the value 1 denoting the total initial concentration of kinase cdc2 (*i.e.* $y_1(0) + y_2(0) + y_3(0) + y_4(0)$) was chosen by convenience. The values of the reaction rates constants are fixed as to have the model display the oscillatory behavior: $k_1 = 0.015, k_3 = 200, k_4 = 180, k_4' = 0.018, k_6 = 1, k_8 = 10^3, k_9 = 10^6$.

The corresponding two-term tropicalized system writes as:

$$\begin{cases} \frac{dy_1}{dt} = Dom(k_6 y_4, k_9 y_2) - k_8 y_1 \\ \frac{dy_2}{dt} = k_8 y_1 - Dom(k_3 y_2 y_5, k_9 y_2) \\ \frac{dy_3}{dt} = k_3 y_2 y_5 - Dom(k'_4 y_3, k_4 y_4^2 y_3) \\ \frac{dy_4}{dt} = Dom(k'_4 y_3, k_4 y_4^2 y_3) - k_6 y_4 \\ \frac{dy_5}{dt} = k_1 - k_3 y_2 y_5 \\ \frac{dy_6}{dt} = k_6 y_4 - k_7 y_6 \end{cases}, \tag{6.25}$$

In [138, 139], a hybrid model of the Tyson cell cycle is obtained by detecting and eliminating QSS species of the original model, pruning dominated monomials, and then ultimately tropicalizing the reduced-size model. The resulting model is a one-term tropicalization, described by the following two-variable ODE system (the remaining 4 species can be retrieved using mass invariants and quasi-steady state equations):

$$\begin{cases} \frac{dy_3}{dt} = Dom(-k'_4 y_3, -k_4 y_3 y_4^2, k_1) \\ \frac{dy_4}{dt} = Dom(k'_4 y_3, k_4 y_3 y_4^2, -k_6 y_4) \end{cases} \tag{6.26}$$

Figure 6.5 depicts the trajectories of the three ODE systems for the concentration of species y_4, as well as the tropicalization approximation error, computed as the difference between the original trajectory and each of the reduced models.

The equations for the lower and upper bounds on the concentration of y_4 are:

$$\begin{cases} \frac{dy_3}{dt} = f_{\min}(t_{\downarrow}^{1,1}, t_{\downarrow}^{2,1}, t_{\downarrow}^{3,1}) \\ \frac{dy_4}{dt} = f_{\max}(t_{\uparrow}^{1,1}, t_{\uparrow}^{2,1}, t_{\uparrow}^{3,1}) \end{cases} \tag{6.27}$$

with

- $t_{\downarrow}^{1,1} = k'_4 \cdot \max(\underline{x_3}, 1 - \overline{x_1} - \overline{x_2} - \underline{x_4}) - k_6 \cdot \min(\underline{x_4}, 1 - \underline{x_1} - \underline{x_2} - \underline{x_3})$
- $t_{\downarrow}^{2,1} = k_4 \cdot \max(\underline{x_4}, 1 - \overline{x_1} - \overline{x_2} - \overline{x_3})^2 \cdot \max(\underline{x_3}, 1 - \overline{x_1} - \overline{x_2} - \underline{x_4}) - k_6 \cdot \min(\underline{x_4}, 1 - \underline{x_1} - \underline{x_2} - \underline{x_3})$
- $t_{\downarrow}^{3,1} = k'_4 \cdot \max(\underline{x_3}, 1 - \overline{x_1} - \overline{x_2} - \underline{x_4}) + k_4 \cdot \max(\underline{x_4}, 1 - \overline{x_1} - \overline{x_2} - \overline{x_3})^2 \cdot \max(\underline{x_3}, 1 - \overline{x_1} - \overline{x_2} - \underline{x_4}) - k_6 \cdot \min(\underline{x_4}, 1 - \underline{x_1} - \underline{x_2} - \underline{x_3})$
- $t_{\uparrow}^{1,1} = (1+\epsilon) \cdot k'_4 \cdot \min(\overline{x_3}, 1 - \underline{x_1} - \underline{x_2} - \overline{x_4}) - k_6 \cdot \max(\overline{x_4}, 1 - \overline{x_1} - \overline{x_2} - \overline{x_3})$
- $t_{\uparrow}^{2,1} = (1+\epsilon) \cdot k_4 \cdot \min(\overline{x_4}, 1 - \underline{x_1} - \underline{x_2} - \underline{x_3})^2 \cdot \min(\overline{x_3}, 1 - \underline{x_1} - \underline{x_2} - \overline{x_4})/(1^2) - k_6 \cdot \max(\overline{x_4}, 1 - \overline{x_1} - \overline{x_2} - \overline{x_3})$
- $t_{\uparrow}^{3,1} = k'_4 \cdot \min(\overline{x_3}, 1 - \underline{x_1} - \underline{x_2} - \overline{x_4}) + k_4 \cdot \min(\overline{x_4}, 1 - \underline{x_1} - \underline{x_2} - \underline{x_3})^2 \cdot \min(\overline{x_3}, 1 - \underline{x_1} - \underline{x_2} - \overline{x_4}) - k_6 \cdot \max(\overline{x_4}, 1 - \overline{x_1} - \overline{x_2} - \overline{x_3})$

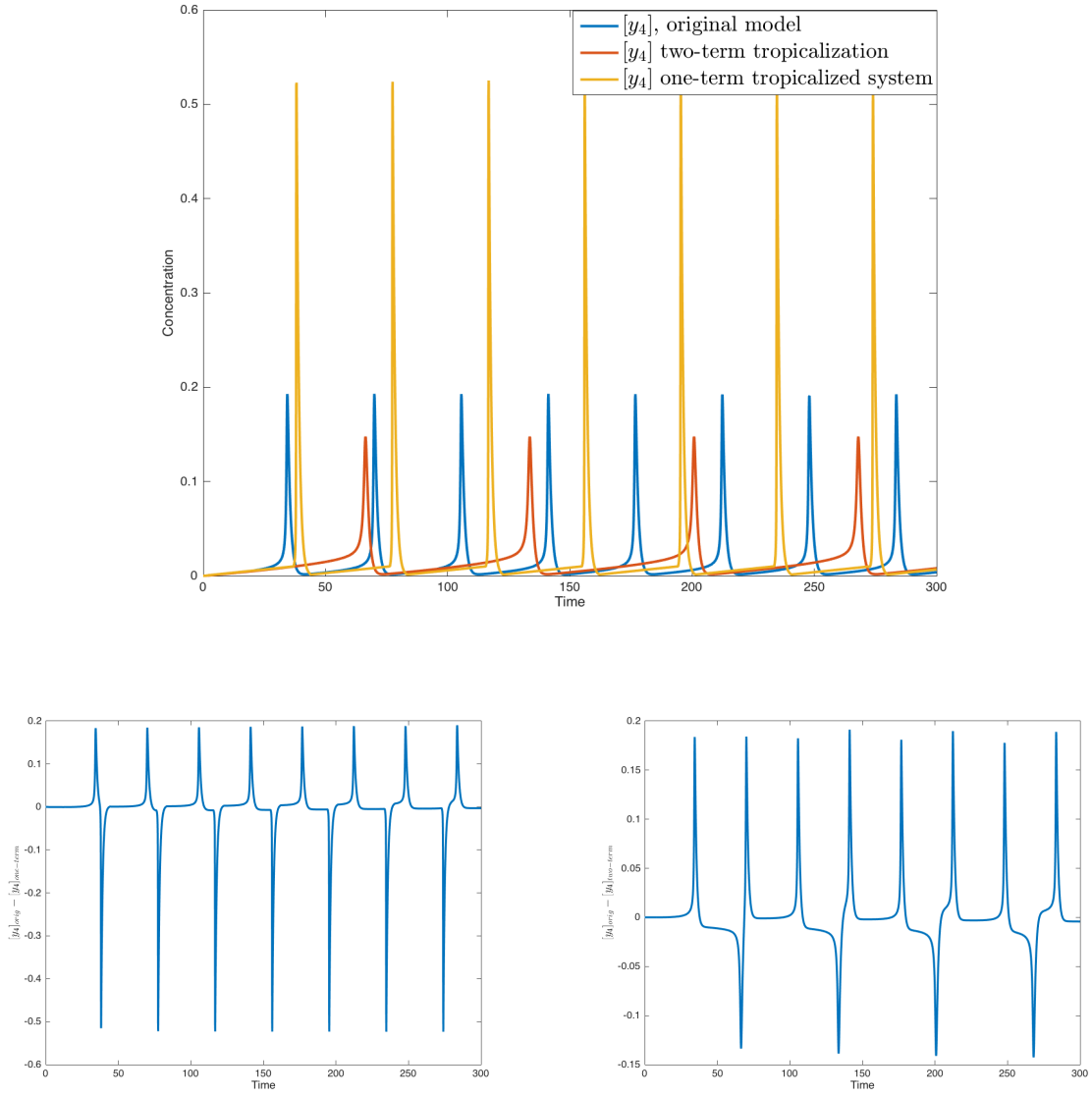


Figure 6.5: Comparison of the original Tyson model with its tropicalized versions, for species y_4. The two-term tropicalization of Definition 6.2.3 is dubbed "two-term tropicalization", while the reduced model of [138] will be referred to as the "one-term tropicalized system". **(Top)** Comparison of system trajectories. All three systems display oscillatory behavior, however of different periods and amplitudes. **(Bottom)** We plot the approximation errors of the two-term tropicalization (left) and the one-term tropicalization (right), as the difference in concentration between the original system and the reduced one. As expected, the two-term tropicalization is more a accurate approximation of the original system, however neither approximation is able to faithfully recreate the original oscillatory behavior, from a **quantitative** point of view - they only do so **qualitatively**. An explanation for this can be given by the existence of *sliding modes* in the tropical manifold [138].

Besides having the inconvenient of analyzing trajectories of the original model in order to detect QSS species, the reduced model of [138] - which we dub the "one-term" sytems in the figure captions - suffers from the sliding mode-related issues mentioned above: although both the smooth (original) and the reduced system exhibit oscillating behavior and have stable periodic trajectory (*i.e.* limit cycle), the period of the tropicalized limit cycle is different with respect to that of the smooth cycle, due to the tropicalized trajectory sliding along the tropical manifolds instead of changing sectors. Having different oscillation periods means in turn that assessing the accuracy of the tropicalized reduced model is not a trivial question, as the distance between original and tropicalized trajectories is variable from cycle to cycle (as can be seen in Figure6.6). What's more, it can also provide an indication of the poor performance of tropicalization based reduction methods when dealing with more complex systems, such oscillating systems.

Indeed, by applying our method to the original Tyson model, we are able to effectively provide guarantees on the reduced model, albeit not very strong ones: this could be interpreted as an indication of the poor accuracy of the tropicalized Tyson model. In Figure 6.6, we plot the bounds for the concentration of species y_4 obtained using our method, in order to compare the trajectory of the original model to the one of the hybrid one that can be found in [138]. The lack of oscillating behavior in the computed bounds could intuitively be explained by the difference in period of the original and reduced systems, that causes a shift at every cycle in the tropicalized trajectory w.r.t. the original behavior. Nonetheless, the obtained bounds accurately capture the amplitude of the tropicalized model. One also notes that the time points where the upper bound, respectively the tropicalized system, begin to diverge w.r.t. the original trajectory coincide.

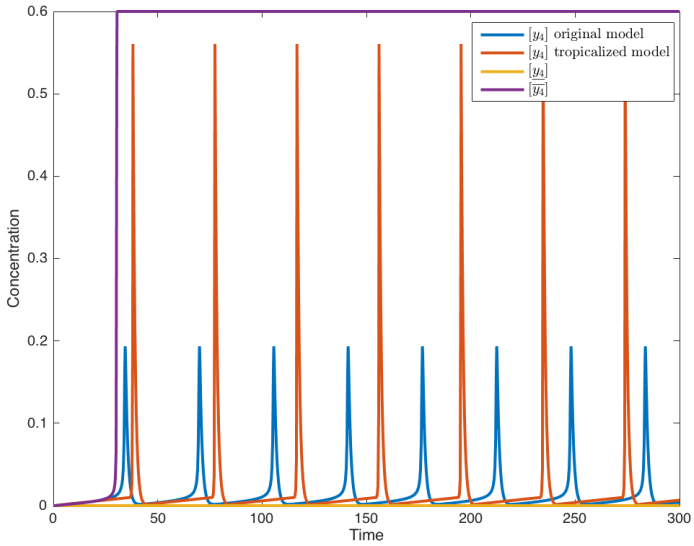


Figure 6.6: Estimating the accuracy of the tropicalized Tyson model of [138]: bounds on the concentration of $[y_4]$, with respect to simulation time, for $\epsilon = 10^{-3}$.

From a more practical point of view, we note that while simulation of the tropicalized Tyson model proposed in [139] was performed in 354.87 seconds, on a 2.2 GHz Intel Core i7 processor simulating the model obtained via our method was carried out in 9.77 seconds using the same numerical integration method (*i.e.* Matlab's *ode15s*), thus providing a significant improvement in computation time.

We note that an alternative reduced model is obtained in [138], using tropical equilibration, that circumvents the need to simulate the original system. We plan to include tropical equilibration techniques in future work.

6.5 Comparison with existing methods

We mentioned previously that numerical errors stemming from numerical integration are ignored herein. Indeed, numerical integration methods, while heavily used, only provide *approximations* of the solution of the initial value problem (IVP) of ODE systems. Even when using variable-step size methods, there are no guarantees that the approximate solution computed by the chosen method is close to the actual solution. In order to solve the drawbacks associated to traditional ODE solvers/numerical solutions of IVP, interval numerical methods for IVP are used for computing validated enclosures of the solution of an IVP for an ODE. For example, the VNODE-LP[136] C++ solver proves that a unique solution to a problem exists, and then computes rigorous bounds that are guaranteed to contain it. Such bounds can then be used to help prove theoretical results, check if a solution satisfied a condition in a safety-critical calculation, or simply to verify the results produced by a traditional ODE solver. Another example of such software is the CAPD library [1]. Both represent well-established software for computing enclosures of generic ODE systems, and are integrated in various SMT solvers (e.g., iSAT[68], dReal[70]). For a more comprehensive state of the art on such methods, the reader is refered to [137].

Interval methods for IVPs for ordinary differential equations are typically based on Taylor series expansions, which require the computation of Taylor coefficients up to some order k. Given a final time point, the aim is to compute interval vectors that are guaranteed to contain the solution to a given IVP, at all intermediary points. In order to compute such interval vectors, interval propagation methods are used to enclose roundoff and truncation errors in the computed bounds, and thus obtain rigorous bounds on the true solution of the ODE.

In our approach, instead of interpreting the differential equations over the state of the system, the interpretation is lifted conservatively over each hyper-face of the hyper-box abstracting the system state (*i.e.*, we over-approximate the derivatives only on the hyper-faces). When compared to our method, interval propagation methods over-approximate the partial derivatives of the function over the whole enclosing hyper-box, instead of doing so only on the hyper-faces. This in turn means that our approach computes tighter bounds than those computed by interval methods for IVPs.

We demonstrate our claim with the following example:

Example 6.5.1
Let us consider the following initial value problem :

$$\begin{cases} \frac{dx}{dt} = y \cdot (2 - \cos(y)) - x \cdot (2 - \sin(y)) \\ \frac{dy}{dt} = x \cdot (2 - \cos(y)) - y \cdot (2 - \sin(y)) \\ x(0) = y(0) = 1 \end{cases}$$

As presented in Section 3, our framework can be decomposed in two independents parts: the first part consists in synthesizing bounds on the derivatives of the original system, and the second part deals with the propagation of said bounds, in order to obtain the enclosing system.

As our goal is to better understand and evaluate tropicalization approaches for biochemical model reduction, so far we chose to focus on bounds obtained by using dominance relations between monomials. The second part of our method simply represents an improved alternative to existing ODE enclosure methods, as explained above, and as such can be used in such methods in order to get better enclosure results.

For example, in order to compare the performance of our method to that of VNODE-LP and CAPD, instead of using dominance relations to derive inequality constraints on species' concentrations, we now use the Taylor Series expansion with k terms (k will serve as a parameter) for the functions sin and cos, in order to derive bounds on $\frac{dx}{dt}$ *and* $\frac{dy}{dt}$*:*

$$\begin{cases} \sin(x) \approx x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \ldots = \sum_{n=0}^{k-1} (-1)^n \frac{x^{2n+1}}{(2n+1)!} \\ \cos(x) \approx 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!} - \ldots = \sum_{n=0}^{k-1} (-1)^n \frac{x^{2n}}{(2n)!} \end{cases}$$

Then, for a fixed order k and an ϵ*, instead of using dominance-related inequalities with our method, one can use the following inequalities:*

$$\sum_{n=0}^{k-1} (-1)^n \frac{x^{2n+1}}{(2n+1)!} - \epsilon \leq \sin(x) \leq \sum_{n=0}^{k-1} (-1)^n \frac{x^{2n+1}}{(2n+1)!} + \epsilon$$
$$\sum_{n=0}^{k-1} (-1)^n \frac{x^{2n}}{(2n)!} - \epsilon \leq \cos(x) \leq \sum_{n=0}^{k(-1} -1)^n \frac{x^{2n}}{(2n)!} + \epsilon$$

,

where $\epsilon = (-1)^{2k+1} \frac{x^{2k+1}}{(2k+1)!} \cos(c_x)$ *for* $c_x \in [0, x]$ *is the residual for the Taylor expansion, and can be bound by* $\epsilon \leq \frac{|x|^{2k+1}}{(2k+1)!}$*, which in turn is* $\leq (2k+1)!^{-1}$ *for* $x \in [-1, 1]$*.*

Our method then proceeds as usual to the computation of ODEs for bounds on the concentrations of x and y.

In Fig.6.7, we compare the accuracy of our method to that of VNODE-LP and CAPD, for different values of the order k. The accuracy is given by the tightness of bounds, which can be

evaluated by computing the difference between the upper and lower bounds, during a simulation. The results indicate that, when compared to existing enclosure interval methods, our approach represents a consistent improvement of several orders of magnitude, across different values of k.

6.6 Conclusion and outlook

In this section, we present an approximation method for biochemical networks, which can also serve as a technique for evaluating the accuracy of existing tropicalization reduction methods that do not involve guarantees. Our approach relies on the multiscaleness property of biochemical systems. Tropical geometry offers a natural framework for studying such networks. Tropical approaches [154, 155] can guide model reduction of ODE systems, by using time- and concentration scales separation to identify and neglect equation terms whose values are significantly smaller than those of other terms of the same equation. This leads to partitioning the state space into different regions, according to which term dominates the others. A similar approach is employed in our method, but instead of neglecting the dominated terms, we propose to conservatively bound their value using an amortizing scale separation constant and the value of the dominant terms. These bounds can be further refined by incorporating the conservation laws of the initial system. The resulting approximated model is composed of two-term ODEs (which we call tropicalized), which by construction provide time-dependent lower and upper bounds for the concentration of the initial system's species. As such, our approach can also serve to test the accuracy of other given reduction methods, while circumventing the execution of the original system: the suitability of a reduction will be confirmed if the reduced model's trajectory lies between the bounds provided by our abstraction.

We have tested our approach on the classical Michaelis-Menten system, a simple extension of it, and Tyson's cell cycle model. Our method can be easily automatized, either using static analysis, or existing symbolic math tools[7] ; as such, Definitions 3.1-3.11 are written in an operational-semantics style, as to describe the different procedures composing the algorithm that implements our method. Further work includes expanding the case studies to larger networks, possibly with no conservation laws.

[7] for example, Matlab's Symbolic Math Toolbox

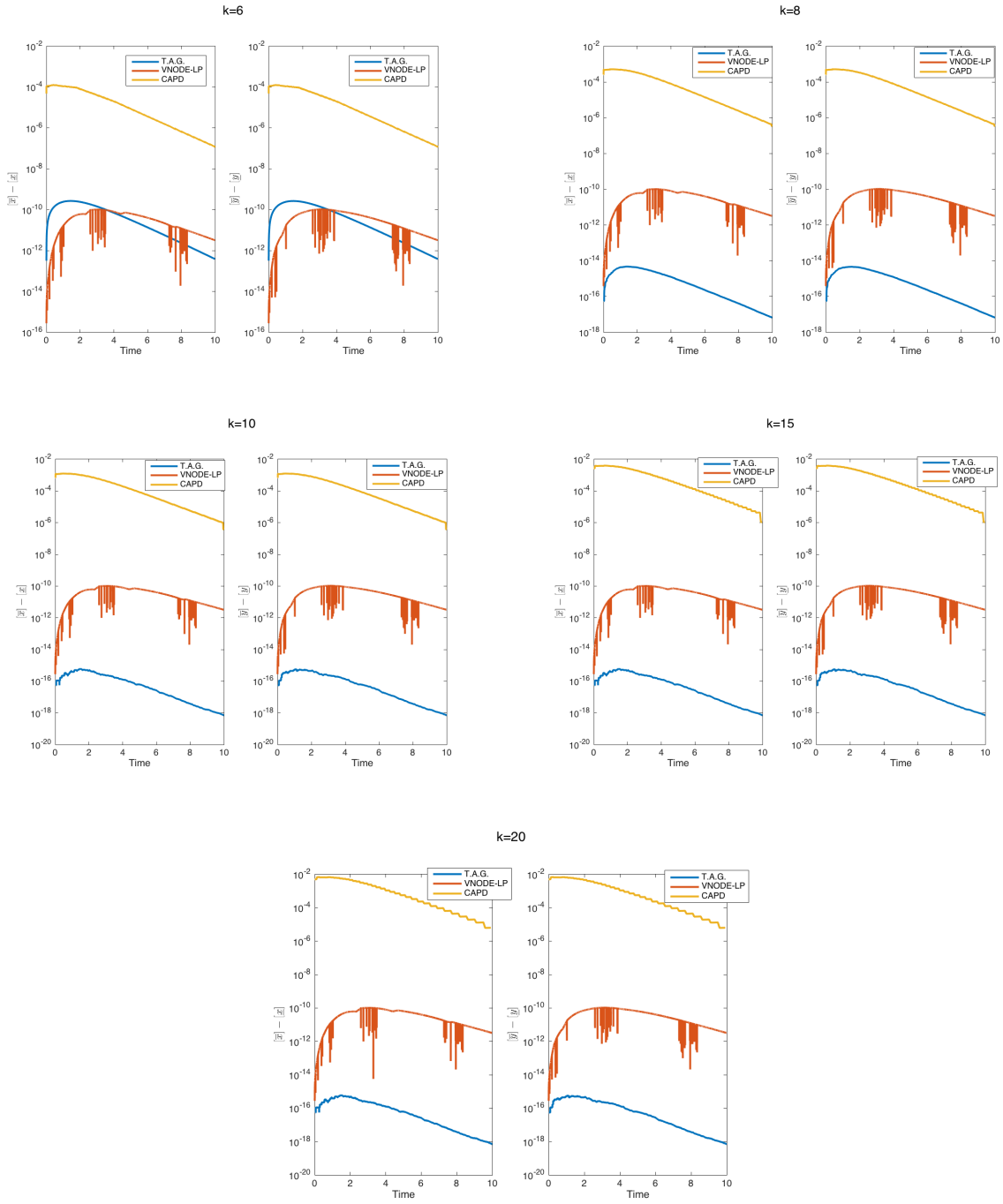


Figure 6.7: Comparison of our method - which we denote by "T.A.G."- with existing ODE enclosure methods CAPD and VNODE-LP. For different values of the Taylor expansion order k, used to derive bounds on the system in Example 5.1, our method ultimately out-performs both CAPD and VNODE-LP, in terms of accuracy; for $k = 6$, our method is initially outperformed by VNODE-LP on the studied example, but ultimately proves to be more efficient, for the time frame $t > 4$. The exponent on the y-axis is an indicator of the decimal precision of the methods (i.e. a value of 10^{-15} means that the first 15 decimals of the computed lower and upper bounds are identical). We also note that both CAPD and VNODE-LP only allow values of $k > 4$.

Part III

Modelling storage and growth

Chapter 7
Effects of cellular resource storage on growth

This chapter is based on a joint work with Guillaume Terradot, parts of which were subject to a publication in [174].

7.1 Introduction

Growing cells have to perform concurrent tasks, each carried out by different groups of proteins. Said proteins, and their tasks, can be roughly classified into:

- *enzymes*, which take up extra-cellular resources and convert them into biosynthetic precursors;
- biosynthetic proteins, the most important of which are *ribosomes*, which are responsible for protein production;
- and other *housekeeping* proteins.

In order to perform these task well, cells need to balance the uptake of extracellular resources with the intracellular demands of biosynthetic processes ([130],[167],[182]). Depending on how they coordinate uptake and consumption, cells can adjust the buildup, or *storage*, of intracellular resources. Cellular storage allows nutrients to be consumed some time after they have been internalized by a living cell [60]. By storing the gains of favourable environmental periods to survive unfavourable ones ([60],[158]), the "storage effect" has been proposed as an evolutionary tunable strategy of coping with variable environments. What's more, it is said to provide an explanation to behavioural diversity [40]: under the assumption that no single strategy can perform best in all environmental conditions, the storage effect explains the coexistence of diverse responses in fluctuating environments, countering the competitive exclusion principle which states that two species exploiting the same set of resources cannot coexist in a closed environment [88].

The careful exercise of balancing, allocating, and storing cellular resources will depend on the growth conditions (growth media, competitors, antibiotics, *etc...*). Indeed, bacterial cells are known to allocate their resources in a context-sensitive manner ([58],[97]). For example, one concrete regulation by which this adaptation happens in bacteria is the ppGpp-dependent stringent

response, which senses starvations in amino acids, *i.e.,* protein precursors, and down-regulates the synthesis of catalysts of biosynthetic processes. What's more, recent modeling work [82] seems to reinforce the idea that near-optimal control strategies w.r.t. cellular resource reallocation are triggered by sensing the concentration of immediate precursors to protein synthesis. Consequently, even if any internalized nutrient can *a priori* be considered an energy storage molecule[1], the study of cellular storage presented in this chapter focuses on resources that are immediate precursors to protein synthesis.

We base our analysis on a recent model [182] that determines growth in terms of coarse-grained cellular mechanisms, and which reflects the adaptive behaviour previously mentioned, *i.e.,* in which allocation of resources is modulated by the amount of protein precursors available in the media. The mechanisms considered in the Weisse model comprise resource uptake and conversion into cellular precursors, as well as how the latter fuels protein biosynthesis, and thus growth. Most importantly, in the Weisse model the growth rate is emergent from the model dynamics, instead of being obtained by optimization - this is different from the other models we have mentioned, such as [130] and [82], in which mechanisms for resource allocation are left implicit and the growth rate is maximized.This aspect will be key to our investigation, as in the second part of our investigation, we study the ability of the model in [182] to efficiently allocate its resources when the availability of the growth substrate changes stochastically, as well as study the impact of various storage strategies on the efficiency of such dynamic reallocations. As successful negotiation of changes in substrate availability is clearly the key to evolutionary success of bacteria, finding out what the model of [182] can teach us about the most efficient resource storage strategies seems to be a worthwhile pursuit.

Our study of storage strategies is composed of two parts. In the preliminary sections, we include a brief review of the model, and introduce several modifications that aim at clarifying some underlying assumptions w.r.t. the definition of mass and growth (Section 7.2). We continue by defining a generic scaling transformation of BRNs that acts as a proxy for the storage phenomenon (Section 7.3), which we then apply to the growth model (Sections 7.3 and 7.4). The first part of our storage strategy investigation is presented in Section 7.5, in which we use a *single-cell model* to investigate the impact of such strategies on cellular growth during shifts of the sugar yield. The results of the numerical experiments enable a number of observations, that we summarize below:

- storage capacity can be modulated over several orders of magnitude, without *significantly* affecting the growth rate,
- however, excessive storage (*i.e.*, of an order of magnitude that depasses those of the previous point) of protein precursor is detrimental to growth,
- the cost of storage, in terms of reduced growth, is condition-dependent, and proves to be higher in rich growth conditions,
- storage results in smoother physiological transitions during environmental up-shifts and increases biomass during such transitions, as resource allocation is dependent on protein

[1] by considering that the decision to use their contained energy has not yet been made, *i.e.* they can subsequently be invested in building ribosomes, or transporters, or any other functions

precursor concentration,

- the evolutionary benefits of storage increase with the frequency and magnitude of environmental fluctuations.

In Section 7.6 we perform a complementary series of experiments, in order to study how storage strategies fare in a competitive context. To do so, we test populations of low- and high-storage strategies against each other, in a variety of environments parametrized by the frequency of two superimposed probabilistic trains of high and low pulses of sugar. Our results suggest that faster growing, low-storage cells perform better in environments with high infrequent sugar pulses, whereas outside this regime they are driven to extinction by the high-storage, slow growers.

Other observations enabled by this second series of numerical experiments are as follows:

- the results of the competition depends on the way an amount of sugar is delivered,
- "fat" cells are better at allocating their resources than "thin" cells,
- the amount of transporters of a cell seems to predict the outcome of the competition,
- during periods of nutrience abundance, fat cells are not the fittest.

7.2 Review of the Weisse cell model

7.2.1 Overview

In the spirit of Molenaar's model of a self-replicating cell [130], the model introduced in [182] - and which will subsequently be referred to as the *Weisse model*, is a coarse-grained mechanistic cellular model built around the three *universal*[2] cellular trade-offs that arise due to finite levels of *(i)* cellular energy, *(ii)* ribosomes, and *(iii)* proteome/cell mass. These trade-offs prove to be expressive enough as to recover the laws of microbial growth, as well as to enable the study of evolutionary benefits of gene regulation, and to explain phenomena such as gene dosage compensation or host effects on the performance of synthetic circuits [182].

The Weisse model describes the allocation of cellular resources to different functions in different growth media, and implements the trade-offs mentioned above by considering two core biochemical processes: nutrient import and metabolism, and gene expression. Consequently, the model combines nutrient import and conversion to cellular energy with the biosynthetic processes of transcription and translation. It includes 14 variables, expressed as *concentrations* (number of molecules per a constant volume of 10^8 units[3] of proteic mass), and accounting for four classes of genes:

- ribosomes, r, which represent the only proteins capable of protein production, and are thus necessary in order to replicate the mass;

[2] In the sense that they are experienced by all living cells.

[3] One unit of proteic mass corresponds to one amino-acid polymerized within a protein.

- transporter enzymes, e_t, for importing external sugar into the cell;
- metabolic enzymes, e_m, for processing the internalized sugar into protein precursors;
- non-ribosomal housekeeping proteins, q, the function of which is not represented in the model, but which account for roughly half of the cell's proteic mass across different growth conditions [167].

Each of the four classes of genes has an associated messenger RNA (mRNA), that conveys genetic information from DNA to the ribosomes. mRNAs can appear either free ($m_x, x \in \{r, q, e_m, e_t\}$) or bound to a ribosome ($c_x, x \in \{r, q, e_m, e_t\}$).

We note that, besides the finite energy, finite ribosomes and finite proteome trade-offs, the following assumptions were used in elaborating the model:

1. Intracellular species are subject to first-order dilution, at a rate proportional to the growth rate;
2. There is no degradation of proteins, however mRNAs are subject to first-order degradation;
3. The binding-unbinding reactions for mRNAs and free ribosomes are assumed to follow mass-action kinetics;
4. Energy consumption from transcription is neglected; instead, energy consumption within the cell is assumed to stem from translation only.

The schematic of the biochemical processes contained in the model is shown in Fig. 7.1.

The model consists in a deterministic system of ODEs, which reads:

$$\frac{ds_i}{dt} = \nu_{imp}(e_t) - \nu_{cat}(e_m) - \lambda s_i, \tag{7.1}$$

$$\frac{da}{dt} = n_s \cdot \nu_{cat}(e_m) - \sum_{x \in \{e_m, e_t, r, q\}} n_x \nu_x - \lambda a, \tag{7.2}$$

$$\frac{dr}{dt} = \nu_r - \lambda r + \sum_{x \in \{e_m, e_t, r, q\}} (\nu_x - k_b r m_x + k_u c_x), \tag{7.3}$$

$$\frac{de_t}{dt} = \nu_{e_t} - \lambda e_t, \tag{7.4}$$

$$\frac{de_m}{dt} = \nu_{e_m} - \lambda e_m, \tag{7.5}$$

$$\frac{dq}{dt} = \nu_q - \lambda q, \tag{7.6}$$

$$\frac{dm_x}{dt} = \omega_x(a) - (\lambda + d_m) m_x + \nu_x - k_b r m_x + k_u c_x, \tag{7.7}$$

$$\frac{dc_x}{dt} = -\lambda c_x + k_b r m_x - k_u c_x - \nu_x, \forall x \in \{r, e_m, e_t, q\} \tag{7.8}$$

Below, we detail the different reactions of the cell model, as well as their associated rate functions.

7.2.2 Nutrient Uptake and Metabolism

The transporter enzymes e_t import extracellular nutrients s into the cell. The metabolic enzymes e_m transform the imported nutrients s_i into a metabolite a, with a stoichiometry n_s. The underlying reactions are assumed to be enzymatically catalyzed, and saturable. Consequently, they follow the Michaelis-Menten kinetics with maximal rates v_t and v_m:

$$s \xrightarrow{\nu_{imp}(e_t)} s_i, \tag{r_1}$$

$$s_i \xrightarrow{\nu_{cat}(e_m)} n_s \cdot a, \tag{r_2}$$

with maximal rates given by

$$\nu_{imp} = e_t \cdot \frac{v_t \ s}{K_t + s}, \tag{f_1}$$

$$\nu_{cat} = e_m \cdot \frac{v_m \ s_i}{K_m + s_i}. \tag{f_2}$$

The *nutrient efficiency parameter* n_s represents the quality of the medium, *i.e.*, the yield of a from s_i. A biological interpretation of n_s is that it is a measure of how many metabolic steps or *anabolic effort* are/is needed by a cell to turn the nutrients present in the environment into protein precursors: the higher n_s, the less metabolic work is needed. This translates in the model into a higher yield of a from s_i for the same metabolic enzyme number.

7.2.3 Transcription

Transcription of mRNA has been estimated to cost ≈ 20 times less ATP (the main energy currency for most cellular processes) than translation [121]. What's more, in *E.coli*, the mass fraction of RNA polymerases - which are the proteins responsible for transcription- is ten times smaller than that of ribosomes [22]. Consequently, it is assumed in the model that in comparison to translation, the cost of transcription is negligible, both in terms of proteic and energetic costs: *(i)* no proteins are needed to produce mRNA, and *(ii)* mRNA production does not consume the metabolite a. However, transcription is assumed to be an *energy-dependent*[4] process with effective rates as follows:

$$\omega_x(a) = \frac{w_x}{\frac{\theta_x}{a} + 1}, \quad \forall x \in \{t, m, r\}, \tag{f_3}$$

$$\omega_q(a) = \frac{w_q}{\frac{\theta_q}{a} + 1} \cdot \mathcal{I}(q), \quad \text{with} \quad \mathcal{I}(q) = \frac{1}{\left(\frac{q}{K_q}\right)^{\alpha_q} + 1}. \tag{f_4}$$

[4] transcription ceases when the cell runs out of energy

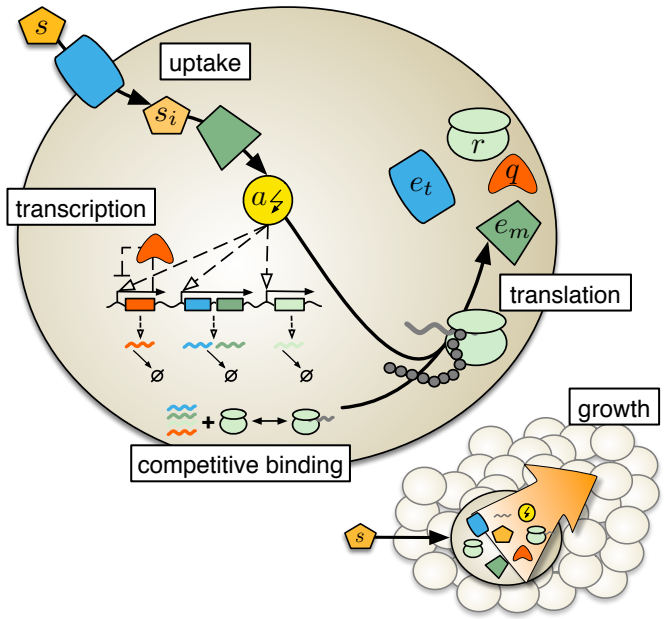


Figure 7.1: Schematic of the Weisse cellular growth model (adapted from Ref. [182]). Four types of proteins are considered: transporter enzymes (e_t), metabolic enzymes (e_m), ribosomes (r), and house-keeping proteins (q). External sugar s is imported into the cell by the transporter enzymes e_t (blue), after which the internalised sugar s_i is processed into protein precursor a by the metabolic enzymes e_m (dark green) - see reactions (r_1),(r_2) for a quantitative description of nutrient import and metabolism reactions. Messenger RNAs (mRNAs) are then produced through transcription, as described in reactions (r_3),(r_4)). Dashed arrows indicate that transcription rates depend on the concentration of metabolites a, but do not consume it. mRNAs m_x compete for the same pool of ribosomes and reversibly bind them to form mRNA-Ribosome complexes c_x (see reaction (r_5)). mRNA-Ribosome complexes then incorporate a to produce the protein x (see reaction (r_6)). Finally, the growth rate is defined as the rate at which cells reproduce their own proteic mass (see reaction (r_7)).

We note the existence of a separate effective rate for the transcription of housekeeping proteins q; this is the case because q-proteins are assumed to be auto-regulated in order to sustain stable protein levels accross different growth conditions. What's more, ribosomes and non-ribosomal genes are assumed to have different transcriptional thresholds, $\theta_r \neq \theta_{nr}$, with $\theta_x = \theta_{nr}, \forall x \in \{e_m, e_t, q\}$. In order to fit the experiments measuring the ribosomal mass fraction and the growth-rate in different growth-conditions from [167], the transcriptional threshold of the ribosomes θ_r must be such that $\theta_r \gg \theta_x, \quad \forall x \in \{e_t, e_m, r\}$. Consequently, for high values of the protein precursor a, the composition of the transcriptome shifts to one that accommodates more ribosomal mRNAs. This regulation mechanism ensures the balance between production and consumption of a. Indeed, it promotes the consuming processes (ribosome-dependent) when quantities of a a are high, and the production processes (metabolic/transporter-dependent) when quantities of a are low.

With all this in place, and assuming that mRNA degradation happens at a rate d, production and consumption of mRNA are described by the following reactions:

$$\varnothing \xrightarrow{\omega_x} m_x \tag{r_3}$$

$$m_x \xrightarrow{d} \varnothing \tag{r_4}$$

7.2.4 Competitive Binding

Including the trade-off that results from the finite pool of intracellular ribosomes is achieved by explicitly modeling the competition between mRNAs for binding free ribosomes. The different types of mRNAs m_x compete for the same pool of free ribosomes r, in order to form the mRNA-ribosome complex c_x. We assume that the different mRNAs m_x have the same binding constant for the ribosomes, k_b, and that c_x have the same dissociation constant, k_u:

$$r + m_x \underset{k_u}{\overset{k_b}{\leftrightarrows}} c_x \tag{r_5}$$

7.2.5 Translation

A simplified mechanism of translation is assumed: first, the mRNA-ribosome complex c_x reversibly binds the "energy" metabolite a (the precursor for synthesizing new proteins), after which the nascent peptide chain elongates by one amino acid, while consuming energy. The two steps are repeated n_x times, where n_x is the length in amino acids of protein x. Finally, the ribosome, the mRNA and the newly synthesised protein are released, and the translation terminates.

This mechanism can be simply denoted by an irreversible reaction in which the mRNA-ribosome complex c_x consumes a to produce the corresponding protein x, after which it dissociates into m_x and r:

$$c_x + n_x \cdot a \xrightarrow{\nu_x} r + x + m_x. \tag{r_6}$$

As previously mentioned, n_x denotes the amount of a required to produce one protein x. The effective translation rate ν_x writes as:

$$\nu_x = c_x \cdot \frac{\gamma(a)}{n_x}, \quad \text{with} \quad \gamma(a) = \frac{\gamma_{\max}}{\frac{K_\gamma}{a} + 1}, \tag{f_6}$$

where $\gamma(a)$ is the rate of elongation per translating ribosome (a incorporated per unit of time per c_x complex). We note that the trade-off resulting from the finite levels of cellular energy is implemented through the energy dependence of Eq. (r_6), and through the sum in Eq.(7.2).

For a more detailed derivation of reaction f_6, the reader is referred to the original paper [182].

7.2.6 Growth and cellular mass

The growth-rate λ is defined as the time derivative of the mass[5], relative to the current considered mass:

$$\lambda = \frac{dM}{dt} \cdot \frac{1}{M} \tag{7.9}$$

At stationary state, the original paper [182] gives the following definition for mass:

$$M = \sum_x n_x \cdot x + n_r \cdot \sum_x c_x, \tag{7.10}$$

i.e., the mass of the cell equals that of the proteic mass. As n_x denotes the number of a per protein x, this means that mass is counted in numbers of a, *i.e.,* in number of amino-acids, meaning that the mass contribution of the two nutrients, s_i and a, is neglected. Indeed, in the original model, the number of a and s_i do not exceed 150 a units, whereas the parametrized proteic mass is taken to be 10^8 a.

At exponential growth, *i.e.*, when intracellular variables are at steady state, the growth rate in the original model is proportional to the rate of protein synthesis, which agrees with other definitions of growth rate in the literature:

$$\lambda = \frac{\gamma(a) R_t}{M}, \tag{f_8}$$

with $R_t := \sum_x c_x$ denoting the number of translating ribosomes.

Equation f_8 implements the finite proteome trade-off through its enforcement of Eq. 7.10, by specifying a value for M at steady-state: $M = M_s$, where M_s is approximately 108 amino-acids for *E. coli*.It is important to emphasize that the underlying assumption for fixing the value $M = M_s$ in the original model is not that the cell mass is fixed, but rather that the model describes the composition of a volume unit of a cell and that this volume unit, independently on the growth conditions, always contains an identical mass $M = M_s = 10^8$. Otherwise said, one should not understand it as a constant mass per cell, but rather as a fixed size of the cell volume we decide to observe. This volume may be set arbitrarily to any value, under the

[5]Or equivalently of the volume, under the assumption of constant density

assumptions that: *(i)* the density (mass per volume) in a cell is constant (or invariant with the growth conditions, as it seems to be the case for protein per volume unit in several microbes [129]) and that *(ii)* the vector field describing the system's dynamic is 1-homogeneous, as we will later show.

However, in our analysis, we will employ a more general definition of the mass than the one given in Eq. 7.10, which includes the contributions of internal nutrients:

$$M = \frac{m_{s_i}}{m_a} \cdot s_i + a + \sum_x n_x \cdot x + n_r \cdot \sum_x c_x, \tag{7.11}$$

where m_{s_i} and m_a are respectively the mass (in mass units this time) of s_i and a, such that m_{s_i}/m_a is the mass of s_i counted in a equivalents. In this manuscript, the number of protein precursors a per cell at stationary state will be scaled over several order of magnitudes, such that its contribution to the mass cannot be neglected anymore. However, this will not be the case for s_i, whose numbers won't exceed 150 per cell. Neglecting s_i's contribution to the mass may therefore be an option; we nevertheless choose to include it. The total mass of the the cell model is once again set arbitrarily to $M = 10^8$, as in [182].

Substituting the new definition of mass in Eq. (7.9), one obtains:

$$M \cdot \lambda = n_s \cdot \frac{ds_i}{dt} + \frac{da}{dt} + \sum_x n_x \cdot \frac{dx}{dt} + n_r \cdot \sum_x \frac{dc_x}{dt} \tag{7.12}$$

where x denotes the amount of protein of type x in the cell, and n_s is the mass of s_i, counted in amino-acids a. Substituting their respective time derivatives with the expressions of Eqs. (7.1)-(7.8), one obtains:

$$M \cdot \lambda = n_s \cdot \nu_{imp} = n_s \cdot e_t \cdot \frac{v_t \cdot s_i}{K_t + s_i} \tag{7.13}$$

which is equivalent to the Monod equation [131] for $\lambda_{\max} = n_s \cdot e_t \cdot v_t$.

As we will see in Section 7.4, the growth rate λ is crucial for connecting cellular processes and growth. More specifically, all intracellular species x_i get diluted due to growth, at a rate λ:

$$x_i \xrightarrow{\lambda} \varnothing. \tag{r_7}$$

which denotes a redistribution of cellular content between mother and daughter cells, and ensures that the cell model, a 1-homogeneous vector field, reaches the steady state parametrized mass. Indeed, when the mass defined in Eq. (7.11) has reached the parametrized value (in this case 10^8), the dilution pseudo-reactions will remove from the model *exactly* the same amount of mass that is being produced.

7.2.7 Model parameters

In our experiments, original parameter values from [182] are used, unless otherwise stated. These values are given in Table 7.1. Parameters that have $\star$ left of their name have been obtained in [182] by fitting the model to data from [167]. We denote by aa the protein precursors, Amino acids, or a in the growth model.

Parameter name	Description	Default value	Unit
s	Amount of external nutrient	10^4	[molecules]
d_m	mRNA-degradation rate	0.1	[min^{-1}]
n_s	Nutrient efficiency	0.5	none
n_r	Ribosome length	7459	[aa/protein]
n_x, $x \in \{t, m, q\}$	Length of non-ribosomal proteins	300	[aa/protein]
γ_{max}	max. Translation elongation rate	1260	[molecules/min]
K_γ	Translation elongation threshold	7	[molecules]
v_t	Max. nutrient import rate	726	[molecules/min]
K_t	Nutrient import threshold	1000	[molecules]
v_m	Max. enzymatic rate	5800	[molecules/min]
K_m	Enzymatic threshold	1000	[molecules]
w_r $\star$	Max. ribosome transcription rate	930	[molecules/min]
$w_e = w_t = w_m$ $\star$	Max. enzyme transcription rate	4.14	[molecules/min]
w_q $\star$	Max. q-transcription rate	948.93	[molecules/min]
θ_r	Ribosome transcription threshold	426.87	[molecules]
θ_{nr} $\star$	Non-ribosomal transcription threshold	4.38	[molecules]
K_q $\star$	q-autoinhibition threshold	152219	[molecules]
h_q	q-autoinhibition Hill coefficient	4	none
k_b	mRNA-ribosome binding rate	1	[molecules^{-1}/min]
k_u	mRNA-ribosome unbinding rate	1	[min^{-1}]
M	Total steady-state cell mass	10^8	[aa]
k_{cm} $\star$	Chloramphenicol-binding rate	0.00599	[μM^{-1}/min]

Table 7.1: Parameter values for the Weisse model

7.3 A scaling procedure for modelling nutrient storage in the cell

7.3.1 Motivation

A key element of the Weisse model is the way the cell allocates its resources: what fraction of the mass is allocated for ribosomes *vs* for transporters and for metabolic enzymes. The main feature of the regulation phenomenon - an example of the previously-mentioned fact that bacterial resource allocation strategies occur as a result of sensing protein precursors' concentration- that is needed to understand our investigation is that *the amount of protein precursors a is positively correlated to the mass fraction of ribosomes, and negatively correlated to that of transporter and metabolic enzymes.* This prompts us to define the storage capacity in the Weisse model as a scaling factor of the protein precursor amount a at stationary state[6]. To do so, we define a generic scaling transformation of Biochemical Reaction Networks (BRNs) that allows us to tune the concentrations of certain chemical species, while preserving the BRN behavior at steady state. In our specific case, this means that we can guarantee by construction that various storage strategies preserve approximatively goodness-of-fit to the original growth data, and therefore correctly match growth conditions to sectorial resource allocations.

In this section, we proceed to define this scaling/reparametrisation procedure, and show how to apply it to reactions with different types of kinetic rate functions: mass-action, Michaelis-Menten or Hill. We also elaborate on how the scaling is (partially) applied to the modified Weisse model, in order to modulate the concentration of protein precursors.

7.3.2 Definition

Assume a biochemical reaction system $A = (\mathcal{S}, \mathcal{R}, \boldsymbol{\alpha}, \boldsymbol{\beta})$, with a set of rate functions $f = \{f_1, \ldots, f_m\}$ that describe the deterministic chemical kinetics of its reactions. Each such function f_j is parametrized by κ_j, the set of reaction rate constants associated with reaction r_j. That is, $\forall \mathrm{r}_j \in \mathcal{R}$, we write $f_j(x; \kappa_j)$ to denote the kinetic law of reaction r_j in state $x \in \mathbb{R}^n_{\geq 0}$ (*e.g.*, mass action, Michaelis-Menten, Hill, etc...)

We will write the deterministic dynamics of species S_i in BRN A in state x as:

$$\left(\frac{d_A S_i}{dt} \mid x\right) = \sum_{j=1}^{m} \nabla_{ij} f_j(x; \kappa_j) \tag{7.14}$$

Definition 7.3.1 (**Scalability**)
Let (A, f) be a BRN with reaction rate functions f, and $S_i \in \mathcal{S}$ a species we want to scale.

Define $d_{\alpha,i} : \mathbb{R}^n_{\geq 0} \to \mathbb{R}^n_{\geq 0}$, the state expansion of species S_i, as:

$$d_{\alpha,i}(x_1, \ldots, x_i, \ldots, x_n) = (x_1, \ldots, \alpha x_i, \ldots, x_n).$$

*Then (A, f) is said to be **scalable** along species S_i, if the reaction kinetics allows rescaling, that is to say if for any rescaling factor α in $\mathbb{R}_{>0}$, and for any reaction r_j in $\mathcal{R}$, there exists κ'_j*

[6] We further elaborate on this choice in Section 7.3.4

such that:

$$f_j(x;\kappa_j) = f_j(d_{\alpha,i}(x);\kappa'_j).$$

We will denote a scalable couple (A, f)*, where* $A = (\mathcal{S}, \mathcal{R}, \boldsymbol{\alpha}, \boldsymbol{\beta})$*, by using the tuple* $\mathcal{A} \equiv (\mathcal{S}, \mathcal{R}, f, \kappa)$.

The scalability condition ensures that the initial reaction fluxes can be retrieved through the scaling of the rates of reactions containing the species of interest (S_i). Evidently, if S_i is not a reactant species of r_j, there is no need for scaling, and one can simply take $\kappa'_j = \kappa_j$.

We note that the scaling is parametrized by *reactions*, rather than *reaction rates*. In models where a reaction rate *name* κ appears in several reactions, what is scaled is κ's *value* in a certain reaction r_j, rather than a scaling of its value across every reaction it appears in. In terms of model variables, it can be interpreted as every κ_j being defined *locally* in reaction r_j, rather than *globally* (i.e., across the whole model).

Definition 7.3.2 (**Scaling of** $\mathcal{A}$ **along species** S_i)
Let $\mathcal{A} = (\mathcal{S}, \mathcal{R}, f, \kappa)$ *be a BRN that is scalable along species* $S_i \in \mathcal{S}$*. We define the scaling of* $\mathcal{A}$ *along species* S_i *by a factor of* α *as follows:*

$$(\mathcal{A}, f) = (\mathcal{S}, \mathcal{R}, f, \kappa) \xrightarrow{\alpha, S_i} (\mathcal{S}, \mathcal{R}, f, \kappa') = \mathcal{B},$$

where each κ'_j *satisfies the condition of Def. 8.2.1.*

A multi-species scaling of a CRN can be achieved by sequentially applying the transformation described above, along different species.

Theorem 7.3.1
Let $\mathcal{A} = (\mathcal{S}, \mathcal{R}, f, \kappa)$ *be a CRN scalable along* $S_i \in \mathcal{S}$*, and let* $\mathcal{B}$ *be its scaling according to the transformation of Def. 7.3.2:* $\mathcal{A} \xrightarrow{\alpha, S_i} \mathcal{B}$*. Then:*

$$\forall x = (x_1, \ldots, x_n) \in \mathcal{R}^n_{\geq 0}, \forall S_i \in \mathcal{S} : \left(\frac{d_{\mathcal{A}} S_i}{dt} \mid x\right) = \left(\frac{d_{\mathcal{B}} S_i}{dt} \mid d_{\alpha,i}(x)\right) \tag{7.15}$$

Proof of Theorem 7.3.1 is obtained directly via the definition of the network dynamics in Def.1, and the scalability condition.

Corollary 7.3.1
If $\mathcal{A}$ *is scalable along* S_i*,* $\mathcal{A} \xrightarrow{\alpha, S_i} \mathcal{B}$*, and* x *is a steady state of* $\mathcal{A}$*, then* $d_{\alpha,i}(x)$ *is a steady state of* $\mathcal{B}$*.*

We note that neither Theorem 7.3.1, nor its corollary should be interpreted as meaning trajectory homothety, but rather as steady state equivalence, as can be observed in Fig. 7.2.

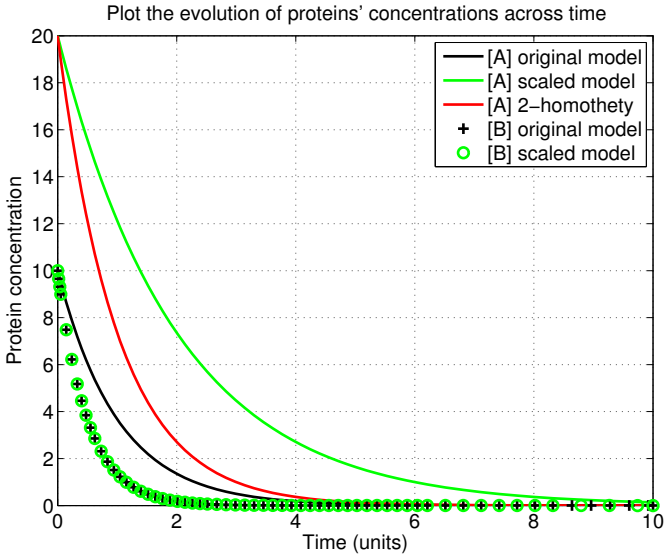


Figure 7.2: Simulation of a simple BRN ($A \xrightarrow{1*[A]} \varnothing; B \xrightarrow{2*[B]} \varnothing$) and its scaling along species A, by a factor of 2. As stated by our theorem, the dynamics of species B remains unchanged. The trajectory of the scaled species is not homothetic to the original trajectory (homothety plotted in red, for comparison), but rather the two models (original and scaled) exhibit steady state equivalence.

7.3.3 Examples

We now show how to apply the scaling transformation to common kinetic laws:

- if reaction r_j has mass action kinetics:

$$f_j(x;\kappa_j) = \kappa_j \cdot \prod_k x_k^{\alpha_{kj}},$$

  the reaction rate scales to $\kappa'_j = \frac{\kappa_j}{\alpha^{\alpha_{ij}}}$, s.t. the reaction dynamics remains unchanged after the scaling: $f_j(x;\kappa_j) = f_j(d_{\alpha,i}(x);\kappa'_j)$

- if reaction r_j has Michaelis-Menten kinetics: $f_j(x_i; v_{max}, K_M) = \frac{v_{max}\cdot x_i}{K_M + x_i}$, the reaction rates scale to $v'_{max} = v_{max}$ and $K'_M = \alpha K_M$, s.t. the reaction dynamics remains unchanged after the scaling: $f_j(x_i; v_{max}, K_M) = f_j(\alpha x_i; v'_{max}, K'_M)$

- if reaction r_j has Hill kinetics: $f_j(x_i; v_{max}, n, K_h) = \frac{v_{max}\cdot x_i^n}{K_h^n + x_i^n}$, the reaction rates scale to $n' = n$, $v'_{max} = v_{max}$ and $K'_h = \alpha K_h$, s.t. $f_j(x_i; v_{max}, n, K_h) = f_j(\alpha x_i; v'_{max}, n', K'_h)$. A concrete example of the results of scaling such a rate function is given in Fig. 7.3.

Example 7.3.1 (**Mass-action kinetics**)
Consistent with the notations used in Theorem 7.3.1, consider a CRN

$$\mathcal{A} = (\{A^*, B\}, \{\mathrm{r}_1, \mathrm{r}_2\}, \{f_1, f_2\}, \{k_1, k_2\} >$$

with reactions:

$$r_1 : \varnothing \xrightarrow{f_1(k_1)} A^*$$
$$r_2 : A^* \xrightarrow{f_2(A^*;k_2)} B$$

and mass-action kinetics:

$$f_1(k_1) = k_1$$
$$f_2(A*;k_2) = k_2 \cdot [A^*]$$

meaning that:

$$\begin{cases} \left(\frac{d_A A^*}{dt} \mid (A^*, B)\right) & = f_1 - f_2 = k_1 - k_2 \cdot [A^*] \\ \left(\frac{d_A B}{dt} \mid (A^*, B)\right) & = f_2 = k_2 \cdot [A^*] \end{cases}$$

Construct $\mathcal{A} \xrightarrow{\alpha, A^*} \mathcal{B} =< \{A^*, B\}, \{\mathrm{r}_1, \mathrm{r}_2\}, \{f_1, f_2\}, \{k_1', k_2'\} >$ *with:*

$$r_1 : \varnothing \xrightarrow{f_1(k_1')} A^*$$
$$r_2' : A^* \xrightarrow{f_2(\alpha A^*;k_2')} B$$

and mass-action kinetics.

Then we can scale reaction rates as $k_1' = k_1$, $k_2' = \frac{k_2}{\alpha}$, *which satisfies Eq.* (7.15) *of Theorem 7.3.1.*

Example 7.3.2 (**Michaelis-Menten kinetics**)
Consider the same two reactions as above, but where reaction r_2 *follows Michaelis-Menten kinetics:*

$$f_2(A^*; v, K_M) = \frac{v \cdot [A^*]}{K_M + [A^*]}$$

Again, one can easily verify that scaling the reaction rates using $K_M' = \alpha K_M$ *and* $v' = v$, *satisfies Eq.* (7.15) *of Theorem 7.3.1.*

Example 7.3.3 (**Hill kinetics**)
To illustrate how our scaling procedure applies to Hill kinetics, we apply it to a model of the toggle-switch, based on the model first published in [71].

Proteins A *and* B *are produced at rates* f_A *and* f_B*:*

$$B \xrightarrow{f_A} A + B \qquad (r_1)$$
$$A \xrightarrow{f_B} B + A \qquad (r_2)$$

The rates of proteins A and B production write:

$$f_A = \alpha_A \cdot \mathcal{H}_B = \alpha_A \cdot \frac{1}{\left(\frac{K_B}{[B]}\right)^{\beta_B} + 1}$$
$$f_B = \alpha_B \cdot \mathcal{H}_A = \alpha_B \cdot \frac{1}{\left(\frac{K_A}{[A]}\right)^{\beta_A} + 1}$$

where α_A and α_B are the maximal production rates of proteins A and B. The production rate of each protein is modulated by $\mathcal{H}_A$ and $\mathcal{H}_B$. They are Hill functions that model the repressions exerted by:

1. *protein A on protein B expression*
2. *protein B on protein A expression*

Proteins A and B are degraded at rates d_A and d_B:

$$A \xrightarrow{d_A} \varnothing \tag{r_3}$$
$$B \xrightarrow{d_B} \varnothing \tag{r_4}$$

We set the initial conditions to $[A](t=0) = 0$ and $[B](t=0) = 20$. The parameters we used for the simulations of the non rescaled model are: $d_A = 1$; $d_B = 1$; $K_A = 1$; $K_B = 1$; $\alpha_A = 15.6$; $\alpha_B = 3.12$; $\beta_A = 2.5$; $\beta_B = 1$. In order to rescale by 3 the stationary state concentration of the protein A, we rescale the parameters d_A and K_A such that: $d_A = 1/3$ and $K_A = 3$. The relation between the original and scaled toggle switch can be observed in Figure 7.3

7.3.4 Storage capacity in the Weisse model

In living cells, adenosine triphosphate (ATP) represents one of the main energy currencies, and as such, it can represent a possible storage location for living organisms. However, because of the elevated rates of ATP-dependent processes, it would quickly be depleted in the absence of ATP synthesis processes such as glycolysis or respiration [19]. This makes ATP a short sighted form of energy storage; indeed, living cells -which include bacteria- store energy under other chemical forms, such as polyglucoses or lipids, that are not directly substrates of biosynthetic processes [183].

In the model from [182], the chemical species that perform catalytic functions are the proteins: ribosomes, transporters, metabolic enzymes, and housekeeping proteins (whose functions are not described, but they are nonetheless assumed to be essential). The two nutrients, s_i and a, are biomass precursors, and, as such, can be considered as molecules that store energy. As we are interested in observing the interplay between storage and resource allocation, and seeing that allocation to different proteic sectors depends on the levels of protein precursors a, we choose a, rather than s_i, as the main molecule under which storage takes place.

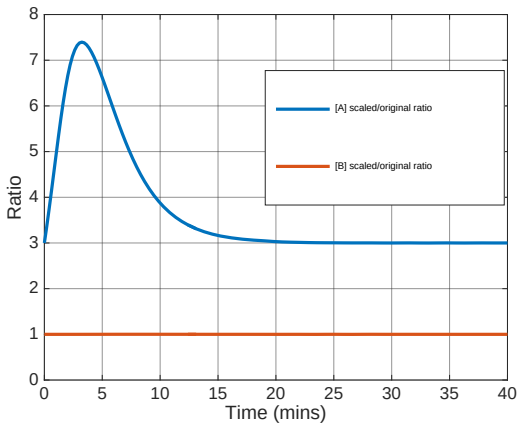


Figure 7.3: An example of scaling a Hill-kinetic rate function: simulation of the toggle switch (model adapted from [71]) before and after scaling species A by a factor of 3. The genetic toggle switch is a synthetic, bistable gene-regulatory network, composed of two repressors: proteins A and B, and two constitutive promoters. Each promoter is inhibited by the repressor which mRNA is transcribed from the opposing promoter. The concentrations of proteins A and B are denoted by $[A]$ and $[B]$. The plotted ratio is the concentration of a protein in the rescaled model divided by the concentration of the same protein in the non rescaled model. As expected, the rescaling preserves the stationary state value of $[B]$ (ratio is equal to 1) while rescaling by 3 the stationary state value of $[A]$. Note that even though the stationary state values for $[A]$ and $[B]$ are preserved, respectively rescaled by a factor of 3 (as predicted by Theorem 1), this is not necessarily the case for the transition regime.

Consequently, we define the *storage capacity* of a cell as the scaling factor α of the steady state amount of protein/biomass precursor a. This definition will allow us to modulate the size of the a pool at stationary state by tuning the storage capacity α: increasing the cell's storage capacity by a factor of α is achieved by scaling the model along species a by the same factor.

In principle, applying our scaling methodology to the the pool of a in the model will not affect the steady state fluxes of the model. This in turn would allow the study of transient responses to environmental fluctuations, in cell models that behave identically at stationary state. The parameters of the original paper [182] being fitted on stationary state data of *E.coli* growing on different substrates [167], such a rescaling does not impact the quality of the fitting - at least in a certain regime, that we present in this section. Without dynamic data that could discriminate between models, said parameters are consequently equally apt at describing the stationary behavior of *E.coli*.

Rescaling the amount of protein precursor a at steady state according to the procedure described previously implies rescaling the parameters of all reaction fluxes that are functions of a, so that they become invariant by modulation of the storage capacity α. In the Weisse cell model, three reaction rates are functions (in the mathematical sense) of a:

- $\gamma(a)$, the rate at which translation occurs,

- $\omega_x(a)$, the transcription rates of each mRNA,
- $\lambda(a)$, the dilution rate of a via growth[7].

For convenience, when adjusting the stationary state concentration of a, we denote with $*$ the reference parameters and concentrations, *i.e.*, their values when $\alpha = 1$.

7.3.4.1 Translation

For $\gamma(a)$, we note that consumption of a through translation follows Michaelis-Menten kinetics, see rate (f_6). According to Example r_1, we scale the affinity of the ribosome for a^* when translating proteins by a factor of α:

$$K_\gamma = \alpha \cdot K_\gamma^* \tag{7.16}$$

We can check that this scaling of γ preserves the fluxes of all species' concentrations, save for a.

In other words, we verify that the consumption rate of the protein precursor a via translation is independent of the storage capacity α.

If we denote the total amount of mRNA-Ribosome complexes by $R_t = \sum_x c_x$, then the consumption rate of a by translation is equal to $R_t \cdot \gamma(a; \gamma_{max}, K_\gamma)$, where $\gamma(a; \gamma_{max}, K_\gamma) = \dfrac{\gamma_{\max}}{\dfrac{K_\gamma}{a} + 1}$ is the rate of elongation per translating ribosome.

It then follows that:

$$\forall \alpha \in \mathbb{R}^n_{\geq 0}, R_t \cdot \gamma(a^*; \gamma^*_{max}, K^*_\gamma) = R_t \cdot \gamma(\alpha a^*; \gamma^*_{max}, \alpha \cdot K^*_\gamma) = R_t \cdot \frac{\gamma^*_{\max}}{\dfrac{K^*_\gamma}{a^*} + 1} \tag{7.17}$$

7.3.4.2 Transcription

In a similar fashion, the transcriptional rate $\omega_x(a)$ also follows Michaelis-Menten kinetics, see rate (f_4). Therefore, we scale the transcriptional thresholds by α:

$$\theta_x = \alpha \cdot \theta_x^* \tag{7.18}$$

Once again, we check that this scaling of θ preserves the fluxes of all species' concentrations, save for a, or that the mRNA production rate for each gene x at exponential growth (i.e.,

[7]We note that in order to be coherent with the notations in the Definitions 8.2.1-7.3.2, the reaction fluxes need to be written as a function of both the species concentrations and the parameters they depend on (*i.e.* write $\gamma(a; \gamma_{max}, K_\gamma)$ instead of simply $\gamma(a)$), but for concision purposes, the shorter notation is preferred in the manuscript, unless formal proofs using the definition notations are involved.

$\omega_x(a) = \dfrac{w_x}{\dfrac{\theta_x}{a} + 1}$) is invariant to the storage capacity α:

$$\forall \alpha \in \mathbb{R}^n_{\geq 0}, \omega_x(a^*; w_x^*, \theta_x^*) = \omega_x(\alpha a^*; w_x^*, \alpha \cdot \theta_x^*) = \frac{w_x^*}{\dfrac{\theta_x^*}{a^*} + 1} \tag{7.19}$$

7.3.4.3 A discussion on dilution and growth

In order to scale the model according to the method of Definition 7.3.2, the dilution rate of a by growth, *i.e.* $\lambda(a) \cdot a$, should be scaled as well. Then Theorem 7.3.1 would ensure the preservation of the scaled model's behavior at steady state.

However, the scaling of the dilution reaction will result in either the coexistence of two different growth rates (which contradicts the intuition behind the concept of growth rate itself, which is a unique property of the cell), or in a modified growth rate that would no longer preserve the reaction fluxes.

The reason for the emergence of this contradictory behavior is that in order to scale the dilution reaction (i.e., $a \xrightarrow{\lambda} \varnothing$) by a factor of α, one would need to scale down λ^* such that the rescaled growth rate becomes $\lambda(\alpha) = \frac{\lambda^*}{\alpha}$. Indeed, at first glance, such a rescaling would keep the rate of a dilution invariant with the rescaling, as $\frac{\lambda^*}{\alpha} \cdot \alpha \cdot a^* = \lambda^* \cdot a^*$, where $a = \alpha \cdot a^*$. However, since the growth rate is a unique property of the cell (a cell cannot have two growth rates simultaneously), rescaling the dilution reaction for a would also change the dilution rate of every other chemical species in the model, thereby breaking the preservation of the fluxes by the rescaling of every chemical species other than a.

This interdiction to scale the dilution reactions stems from the observation that said dilution reactions are nothing more than a practicality that allows the description of the growing cell model, whilst keeping its mass constant, and are thus artificial pseudo-reactions that are added to the actual biological model. More specifically, when the model has reached the parametrized mass (in this case 10^8), for each unit of total mass added by growth, the dilution reaction removes as much mass as was created, ensuring that the mass remains constant. The contribution of each chemical species to the mass may still vary, as the composition of the mass added by growth is not necessarily the same than the composition of the mass that is removed by the dilution pseudo-reaction (such is the case, if the model is not at stationary state).

Indeed, the growth rate and its associated dilution pseudo-reactions can be formally shown to be an emergent property of the growth model, instead of being *sensu stricto* chemical reactions (for the derivation, see Section 7.4), which justifies their unscalability.

However, this "partial" model scaling is sufficient to preserve the steady state model behavior over a large range of values of the scaling parameter/storage capacity α. Indeed, we have seen above that the scaling of γ and θ by definition preserve the fluxes of all species' concentrations, save for a. What's more, we can numerically show that for a wide range of values of α, the dynamics of species a in the Weisse model remains virtually unchanged, despite the fact that we choose not to apply the scaling to its dilution reaction (which means that Theorem 7.3.1 does not apply to a). For this, Figure 7.4 proves that, for values of α in the interval

$[1, \approx 10^5]$, the contribution of the dilution pseudo-reaction to the depletion of a is negligible when compared to that from its consumption by translation:

$$R_t \cdot \gamma(a) \approx R_t \cdot \gamma(a) + \lambda \cdot a; \tag{7.20}$$

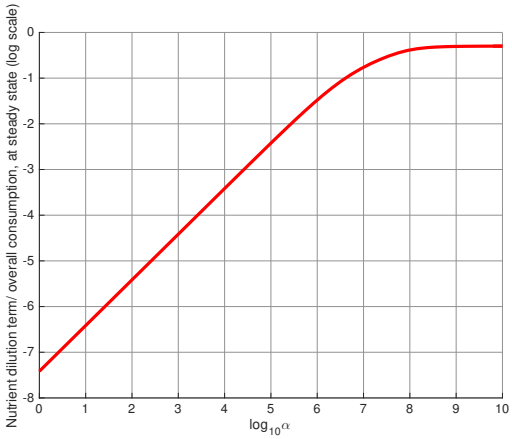


Figure 7.4: For values of the scaling parameter $\alpha \in [1, \approx 10^5]$, the contribution of the dilution pseudo-reaction to the depletion of a is negligible when compared the consumption of a by translation.

For example, at $\alpha = 10^5$, the dilution accounts for less than 0.3% of the total consumption of a. Therefore, for $\alpha \in [1, \approx 10^5]$, the time derivative of a can neglect the contribution of dilution:

$$\frac{da}{dt} = n_s \cdot \nu_{cat} - R_t \cdot \gamma(a) - \lambda \cdot a \approx n_s \cdot \nu_{cat} - R_t \cdot \gamma(a), \tag{7.21}$$

which means its dynamics remains unchanged in the partially scaled model: $\frac{da^*}{dt} = \frac{d\alpha a^*}{dt}$.

This concludes the proof for the preservation of steady state behavior for all species involved in the WS model, for small enough values of α. This statement will be reinforced by one of the numerical observations of Section 7.5, namely that the growth rate does not change for $\alpha < 10^4$, and only decreases by a few percents as long as $\alpha < 10^6$.

7.4 The Growth Rate λ is an emergent property of the model

In the Weisse model, cell division is modeled *implicitly*, by keeping the mass M at a constant value, and diluting all intracellular species at a rate λ (i.e., the *growth rate*, which connects cellular processes and growth). In this section, we show that the use of λ as a dilution term can be explained by using the 1-homogeneity property of vector fields. More specifically, we show that the growth rate λ, with the associated pseudo-reactions of dilution, are emergent behaviors when switching from an extensive, volume-independent description of a chemical system, to an intensive, volume-dependent one (i.e., when switching from describing the system using a vector of species' copy numbers to a description that uses species' concentrations).

Let $F\colon \mathbb{R}^n \mapsto \mathbb{R}^n$ be a vector field.

We say that F is 1-homogeneous, if:

$$\forall \alpha, X \in \mathbb{R}, \qquad F(\alpha X) = \alpha F(X). \tag{7.22}$$

In the case of chemical reaction networks, we suppose a 1-homogeneous vector field $F\colon \mathbb{R}^n \mapsto \mathbb{R}^n$ describing the system's dynamics.

We write $X \models F$, if X is a solution of F, and $\varphi_t^F(X_0)$ to denote the value at t of X, the unique solution of the Initial Value Problem:

$$\begin{cases} \forall t, \frac{dX(t)}{dt} = F(X(t)) \\ X(0) = X_0 \end{cases} \tag{7.23}$$

The intuition behind choosing the vectors X are that they denote vectors of species' copy numbers. In order to switch to an intensive, concentration-based description of the system, let $v \in \mathbb{R}^n$ be a vector of the same dimension as the vectors X, that denotes the *volume occupancy* of each species.

Then define:

$$x \stackrel{\text{def}}{=} \frac{X}{\langle v, X\rangle} \tag{7.24}$$

to be the description in terms of vectors of *species concentrations*, where $\langle , \rangle$ denotes the scalar product of two vectors. This is the case, as $\langle v, X\rangle$ denotes the *total mass of the system.*

One can then compute the *intensive form* dynamics of the system:

$$\frac{dx}{dt} = \frac{d}{dt}\left(\frac{X}{\langle v, X\rangle}\right) = \tag{7.25}$$

$$= \frac{dX}{dt} \cdot \frac{1}{\langle v, X\rangle} - \frac{X}{\langle v, X\rangle^2} \cdot \frac{d\langle v, X\rangle}{dt} = \tag{7.26}$$

$$= F(X) \cdot \frac{1}{\langle v, X\rangle} - \frac{X}{\langle v, X\rangle^2} \cdot \langle v, \frac{dX}{dt}\rangle = \tag{7.27}$$

$$= F(\frac{X}{\langle v, X\rangle}) - \frac{X}{\langle v, X\rangle} \cdot \frac{\langle v, F(X)\rangle}{\langle v, X\rangle} = \tag{7.28}$$

$$= F(x) - \langle v, F(x)\rangle \cdot x \tag{7.29}$$

By defining $\lambda(x) \stackrel{def}{=} \langle v, F(x)\rangle$, one observes that in the differential equation we obtain for x (i.e., $\frac{dx}{dt} = F(x) - x \cdot \lambda(x)$), the species' consumption (or "dilution") term is indeed denoted by λ. Thus, the dilution term (with its associated dilution pseudo-reactions) is indeed an emergent property of the *intensive* form, which is insensitive to division, through scaling by a constant v.

We remind the reader that in the general case, v is an arbitrary linear form, which we suppose to be non-null along the trajectories (this is indeed the case if the vector field is given by a mass action reaction network). For our purpose, the following interpretation of variables holds: v denotes the *volume occupancy of each species*, X is the species' copy number vector, $\langle v, X\rangle$ (which we can note as V) is the *total volume of the system* (or mass, if the density is constant), and $x = \frac{X}{V}$ - the *species' concentration* vector. Using this interpretation, we have shown that growth rate is an emergent property of the intensive vector field representation of the system.

We have also shown how, given the ODEs for the extensive form $\frac{dX}{dt} = F(X)$ of a chemical system, as well as v its volume occupancy, one can derive the ODEs for its intensive form: $\frac{dx}{dt} = F(x) - \langle v, F(x)\rangle \cdot x$, which we will note as $F_v(x)$.

The reverse transformation (from intensive vector field to extensive vector field) is also possible. For that, suppose $\frac{dx}{dt} = F_v(x)$, and fix X_0 an initial species copy number vector, v the volume occupancy vector, and $V_0 = \langle v, X_0\rangle$. Then, if one defines the growth rate as $\lambda_v = \langle v, F(x)\rangle$, the volume as V (and note that its dynamics is described by the equation $\frac{dV}{dt} = \lambda_v \cdot V$), and the extensive vector field as $X = V \cdot x$, we show that:

$$\frac{dX}{dt} = \frac{dx}{dt} \cdot V + x \cdot \frac{dV}{dt} = \tag{7.30}$$

$$= F(x) \cdot V - x \cdot \langle v, F(x)\rangle \cdot V + x \cdot \langle v, F(x)\rangle \cdot V = \tag{7.31}$$

$$= F(V \cdot x) = \tag{7.32}$$

$$= F(X) \tag{7.33}$$

which is exactly the initial ODE for the extensive form.

The above derivations show that, more generally, the following proposition (which relates the solutions of the intensive and extensive forms of the system) holds.

Proposition 7.4.1
If F is a 1-homogeneous vector field, and v a vector in the support of F, then:

$$\forall t, \forall X_0, \quad \varphi_t^{F_v}\left(\frac{X_0}{\langle v, X_0\rangle}\right) = \frac{\varphi_t^F(X_0)}{\langle v, \varphi_t^F(X_0)\rangle}$$

The proof of Prop. (7.4.1) is immediate, by using the constructions defined above.

7.5 The Single Cell model: Evolutionary Trade-offs in Cellular Resource Storage

Our first series of experiments aims at studying the impact of resource storage strategies on cellular growth, in the context of a *single cell* model. Consequently, in Section 7.5.1, we start by investigating the effect of different storage capacity strategies on the growth of a cell evolving in a fixed environment (*i.e.*, with a constant value of the nutritional capacity n_s). We repeat this experiment for three environments of different nutritional capacities n_s, and find that, on the one hand, storage capacity can be modulated over several orders of magnitude without significantly impacting growth, and that on the other hand, the negative impact of storage on growth is directly proportional to the environment's nutritional richness.

In 7.5.2, we extend the analysis to fluctuating environments that result in quasi-instantaneous changes of the sugar yield, in order to study the impact of storage strategies on the cell's ability to adapt to such changes in the environment. The results suggest that growth during environmental transitions is positively affected by an increased (up to a certain threshold) storage capacity, and that low storage capacities come with high sensitivity of transcriptional regulation, which in turn result in overshoot regulation during transitions. We finish this first series of experiments by analyzing the trade-offs stemming from the two different types of experiments: on the one hand, the fact that increased storage capacities enable smoother transitions during environmental up-shifts, and on the other hand, the detrimental impact of significantly high storage strategies on the exponential growth rate.

7.5.1 The effects of metabolite storage on exponential growth rate

We start by defining the notion of *storage cost*, in terms of the fraction of protein precursor a that, instead of being employed for biomass production, is diluted via the increase of cell volume:

Definition 7.5.1 (**Storage cost**)
*Let $\mathcal{A}_{Weisse} = (\mathcal{S}, \mathcal{R}, f, \kappa)$ denote the Weisse model as a scalable BRN along the protein precursor species $'a'$, and let α denote a fixed scaling factor. The term "α-**storage cost**" refers to the cost of scaling $\mathcal{A}_{Weisse}$ along species $'a'$ by a factor of α, and is defined as:*

$$\eta(\alpha) \equiv \frac{\lambda \cdot [a]}{\lambda \cdot [a] + \gamma \cdot R_t} \tag{7.34}$$

$$\approx \frac{\alpha \cdot [a^*]}{M + \alpha \cdot [a^*]} \tag{7.35}$$

where λ denotes the growth rate, $R_t = \sum_x c_x$ denotes the total number of translating ribosomes, and $[a]$ (respectively $[a^*]$) denotes the concentration of protein precursor in the scaled (respectively original) model.

We note that the approximation of (7.34) by (7.35) is justified by the steady-state equality of (f_8) and by Theorem 7.15.

With all this in place, we can proceed with the numerical experiments in the single-cell context, the results of which allow us to make a number of observations.

The cost of storage depends on the richness of the environment.

Equation (7.35) of Definition 7.5.1 suggests that high storage capacities - which translate into higher a concentrations - come with significant storage costs. The cost of high storage strategies can also be explained intuitively, provided that the notion of "cost" is interpreted as opposing that of "contribution to growth": if one molecule of protein precursor a is invested into biomass production at time t_0, its contribution to growth will be $\frac{1}{M_0}$, where M_0 denotes the amount of protein precursors that have to be invested in order to replicate the entire cell. However, if the same a molecule is to be invested at a later time, $t_1 > t_0$, and if the growth rate is positive - *i.e.*, the mass of cell at t_1 is $M_1 > M_0$-, then its contribution to biomass growth will decrease: $\frac{1}{M_1} < \frac{1}{M_0}$. This suggests that the sooner after its production a molecule a is invested into mass growth, the more it contributes to growth rate. As high storage capacities imply a longer time time between the production and the consumption of protein precursors a, it becomes clear that significant storage capacities have high costs.

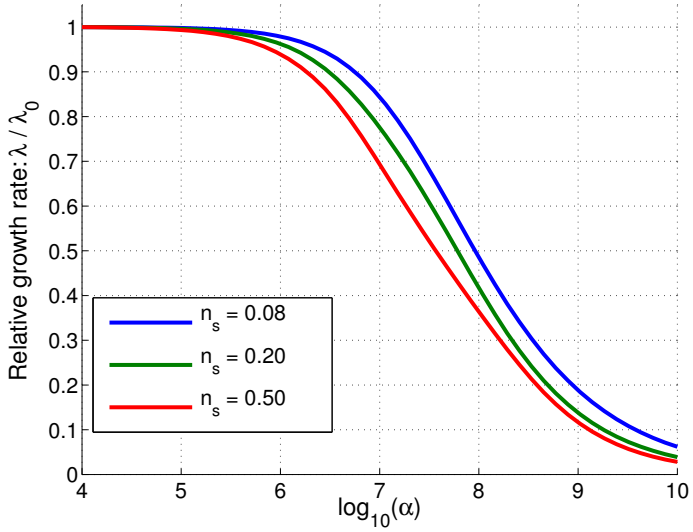


Figure 7.5: The model is simulated for increasing storage capacities α, for three different environments ($n_s = 0.08$, 0.20 and 0.5). We denote by λ_0 the growth rate obtained when the storage capacity is $\alpha = 1$, for any given n_s. The relative growth rate λ/λ_0 is defined as the growth rate obtained for a given storage capacity α, relative to the growth rate when $\alpha = 1$. One observes that the relative growth rate decreases, as the storage capacity α increases. The faster a cell grows, the more storage is detrimental to growth.

What's more, according to the Weisse model [182], the richer the medium, parameterised in the model by the nutrient quality n_s, the higher the concentration of metabolites a available for translation. This observation is experimentally validated, via an increase of tRNA concentration

[59], that in turn results in an increase of the ribosomal translation rate [22, 48, 187, 145, 66]. Consequently, one expects that for cells evolving in nutrient-rich environments, high storage capacities have a negative effect on cellular growth. The evolutionary pressure w.r.t. low storage capacity in rich environments is confirmed by the numerical simulations of Fig. 7.5, in which increasing storage capacities are tested in environments of different nutritional richness. Figure 7.5 also suggests that storage capacity can be modulated over several orders of magnitude without significantly affecting the exponential growth rate of our cell model. Consequently, cells may tune their storage capacity within that range, in order to maximize their biomass production in fluctuating environments, without impairing their ability to produce biomass in absence of these fluctuations.

We next show how variations in storage capacity affect cell biomass production upon environmental fluctuations.

7.5.2 Metabolite storage and adaptation to environmental fluctuations

Cells adjust their resource allocation depending on which medium they grow in. In bacteria for example, the ribosomal content increases as the medium gets more favourable to growth [58]. These adjustments are necessary in order to adapt biomass production in different growth conditions [167, 182]. However, the reallocation of cellular resources to different functions is not instantaneous, as it is constrained by the cellular mechanisms that sense and regulate the different processes; this has potential implications for biomass production during environmental transitions [44, 24]. A common assumption is that, on evolutionary time scales, cells seek to optimize their mean biomass production over time, which means that they also seek to optimize the dynamics of the transition between different physiological states, such that the biomass production over time is maximized.

Consequently, we analyze the behavior of growth rate during environmental transitions. To do so, we consider environmental up-shifts, which we model using an instantaneous change in n_s, a parameter that is a proxy for the *anabolic efficacy of the medium*: *i.e.*, how many metabolic steps are necessary in order to convert the nutrients in the environment into protein precursors. The biological interpretation of shifting the medium abruptly to a higher value of n_s is that nutrients that do not require a lot of metabolic processing before incorporation into biomass (*e.g.*, amino acids) are suddenly made available for the cell.

Storage capacity affects growth during environmental transitions

The results of our simulations, depicted in Fig. 7.6, show that *higher storage capacities result in smoother transitions from one physiological state to another.*

To understand this behaviour we next look at the **sensitivity of transcriptional regulation** to levels of the resource a, which we define as:

$$\sigma_x(a) = \frac{d\omega_x}{da}. \tag{7.36}$$

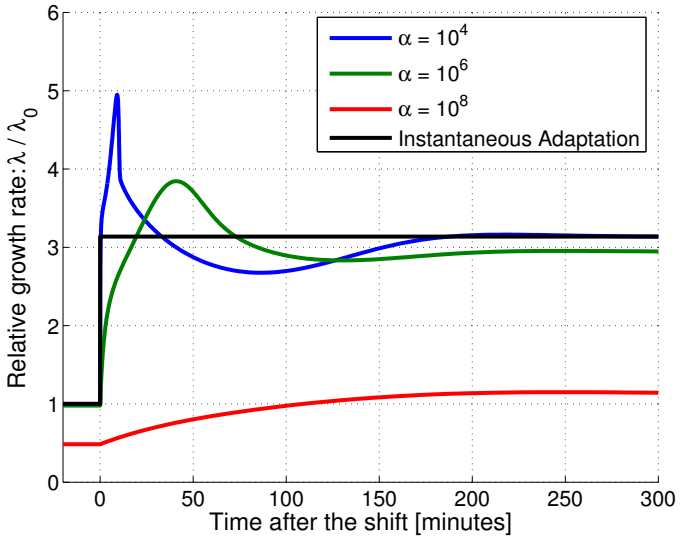


Figure 7.6: λ_0 denotes the growth rate of a model with storage capacity $\alpha = 1$ in a medium quality $n_s = 0.12$. Before $t = 0$, the model is growing steadily for a medium quality $n_s = 0.12$. At $t = 0$ the medium quality is shifted from 0.12 to $n_s = 0.5$. Different lines correspond to cell models with different storage capacities. The black line denotes the behaviour of a (theoretical cell) adapting instantaneously its internal composition to the new growth condition. Different dynamics of growth adaptations arise for different storage capacities: increasing the storage capacity α results in smoother transition of the growth rate following an up-shift. For high enough storage capacity, steady growth rate before the shift is severely impaired (red line), as expected from figure 7.5.

Under the scaling assumption $M \gg a$, which implies that $a \approx \alpha \cdot a^*$, one obtains:

$$\sigma_x \approx \frac{w_x \cdot \theta_x^*}{\alpha \cdot (\theta_x^* + a^*)^2}, \quad x \in \{t, m, r\}, \tag{7.37}$$

$$\sigma_q \approx \frac{w_q \cdot \theta_q^*}{\alpha \cdot \left(\theta_q^* + a^*\right)^2} \cdot \mathcal{I}(q). \tag{7.38}$$

The sensitivity of transcriptional expression thus decreases with increasing storage capacity. An intuitive explanation is that, although storage capacity impacts the steady state concentration of a, it does not affect its rate of production. Consequently, the rate of accumulation *or depletion* of a relative to its current concentration is decreasing, therefore making transcriptional regulation less sensitive to fluctuations in a.

High sensitivity of transcriptional regulation results in overshoot regulation during environmental up-shifts

With this in place, we analyze the reallocation dynamics of the Weisse model during an up-shift. The results depicted in Figure 7.8 suggest that high sensitivity of transcriptional regulation (achieved by low values of storage capacity α) results in a two-stage regulation:

1. **Cellular resources are reallocated from metabolic enzymes to ribosomes.**

   An increase in n_s causes the net production of a to augment(Fig. 7.8C). This is a consequence of the linear increase of the production flux with nutrient quality, $n_s \cdot \nu_{cat}$, while consumption by ribosomal mRNA complexes is saturated, which in turn causes the concentration of a to increase (Fig. 7.8D). Consequently, reallocation of cellular resources from metabolic enzymes to ribosomes occurs: see the increase of ribosomal concentration and decrease of metabolic enzymes concentration depicted in Fig. 7.8A and 7.8B.

2. **Cellular resources are reallocated from ribosomes to metabolic enzymes.**

   By the time the concentration of a decreases, ribosomal concentrations relative to those of metabolic enzymes are already high enough to create an imbalance between the production and the consumption of a. The concentration of a decreases significantly, which results in a reallocation of resources from ribosomes to metabolic enzymes.

Thus, the transcriptional regulation of low-storage cells during an up-shift behaves like a bang-bang control [170]: as described above, most of the available resources go to ribosomal production (relative to that of the exponential growth), which then leads to an excess in ribosomes, requiring to allocate most of the resources to metabolic enzymes. This behaviour is made possible because:

- **Reallocation of internal resources to different cellular functions is not instantaneous.**

  Indeed, allocation arises from competition for ribosomes, between different mRNAs. Since mRNAs have non-zero lifetimes, rewiring mRNA production to one cellular function does not result in a direct update of the mRNA repartitioning to different cellular functions. Additionally, the transcriptional regulation-induced stabilization of mRNA production to the new desired steady state levels takes time, as seen in Figure 7.7. Both arguments motivate the use -in living systems- of post-transcriptional regulation mechanisms that act directly at the translation level, instead of simply tuning transcription.

- **Protein production is not instantaneous.** The feedback coming from transcriptional regulation of ribosomal or metabolic enzyme expression on a concentration also takes time, and can thus cause the overshoot in regulation.

Finally, we analyze the trade-off caused by high storage strategies: on the one hand, the smoother transitions during up-shift, and on the other hand, their detrimental impact on the exponential growth rate.

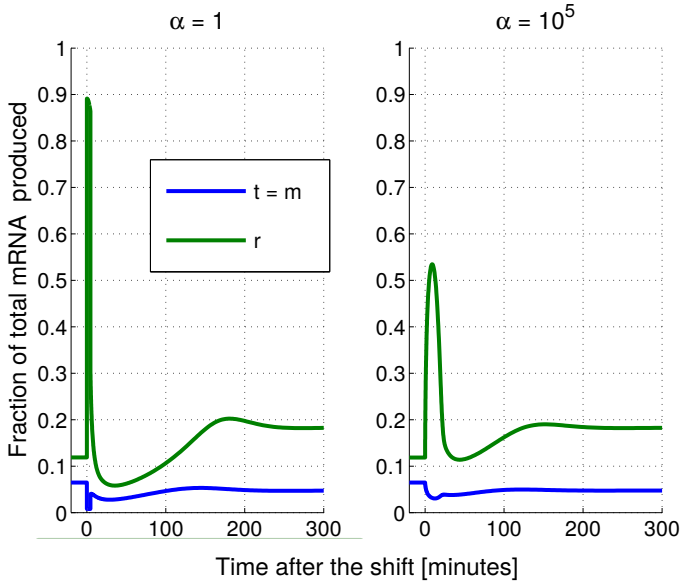


Figure 7.7: Before $t = 0$, the model is growing steadily for a medium quality $n_s = 0.12$. At $t = 0$, the medium quality is shifted from 0.12 to $n_s = 0.5$. The plot shows the fractions of transporter-and-metabolic (green) and of ribosomal (blue) mRNA being produced for two different storage capacity: (Left) $\alpha = 1$ and (Right) $\alpha = 10^5$. As the genetic regulation takes only place at the transcriptional level in the model, it shows the genetic regulation overshoot as a result of fluctuations in a levels during the up shift, (see also figure 7.8. **D**.). This overshoot is attenuated for higher storage capacity: high storage capacities and subsequent high steady state levels of a, act as buffers for the genetic regulation.

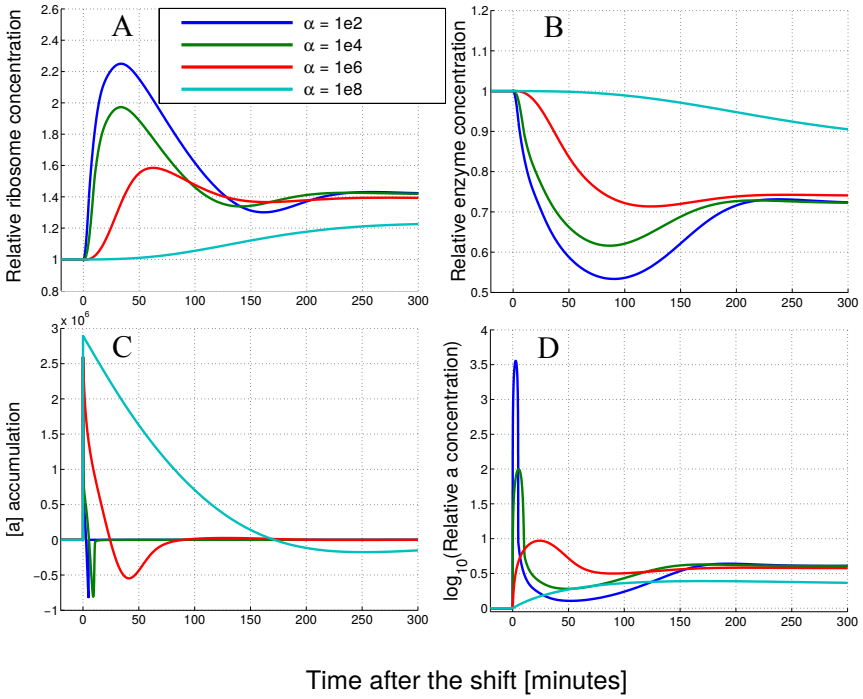


Figure 7.8: Before $t = 0$, the model is growing steadily for a medium quality $n_s = 0.12$. At $t = 0$ the medium quality is shifted from 0.12 to $n_s = 0.5$. Different lines correspond to cell models with different storage capacities. **(A)** Ribosome concentration relative to its concentration before the up-shift **(B)** Enzyme concentration relative to its concentration before the up-shift **(C)** Flux of a concentration: $\frac{da}{dt}$ **(D)** $\log_{10}$ of a concentration relative to its concentration before the up-shift.

Trade-offs between biomass production during transitions and during exponential growth

For analysis purposes, let $B(t_0, T, \alpha)$ denote the biomass production between two time points t_0 and T, for a storage capacity α, and let B_0 denote the biomass at time t_0.

Then:

$$B(t_0, T, \alpha) = B_0 \cdot e^{\int_{t_0}^{T} \lambda(\alpha) \cdot dt}. \tag{7.39}$$

The relative cumulative growth rate δ measures how much the mean growth rate with storage capacity α deviates from the growth rate with the reference parameter ($\alpha = 1$), between two time points t_0 and T. It is defined as:

$$\delta = \frac{\ln(B(t_0, T, \alpha))}{\ln(B(t_0, T, 1))} = \frac{\int_{t_0}^{T} \lambda(\alpha) \cdot dt}{\int_{t_0}^{T} \lambda(1) \cdot dt} \tag{7.40}$$

The biomass production and the relative cumulative growth rate are then related through:

$$B(t_0, T, 1)^{\delta} = B(t_0, T, \alpha). \tag{7.41}$$

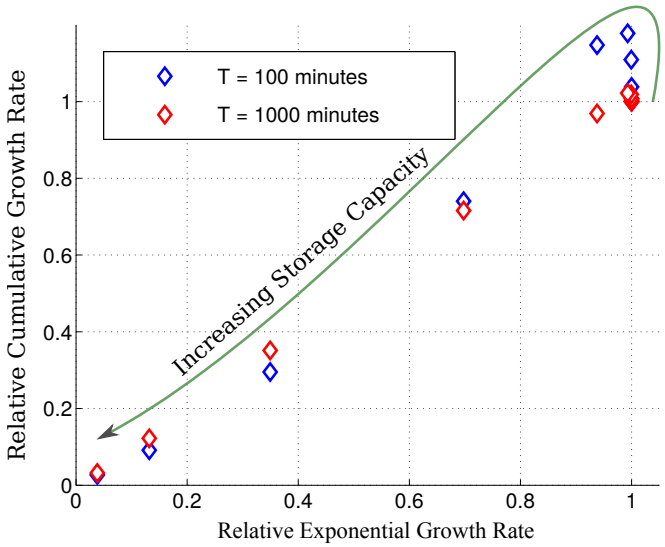


Figure 7.9: Diamonds correspond to different up-shift experiments from $n_s = 0.12$ to $n_s = 0.5$ for increasing the storage capacity α from 1 to 10^{10}. The relative exponential growth rate is the ratio of the growth rate of the cell model when $n_s = 0.5$ for one given value of the storage capacity α over the growth rate of the cell model when $n_s = 0.5$ and $\alpha = 1$. The relative cumulative growth rate defined in Equation (7.40) is computed for $t_0 = -20$ minutes and $T = 100$ or $T = 1000$ minutes. Increasing the storage capacity α results in decrease of the relative exponential growth rate. The benefits from a smoother transition coming from an increased stock of metabolites result in a maximized relative cumulative growth rate at intermediary storage capacities.

Fig. 7.9 illustrates the trade-off between smoother transitions during an up-shift and the detrimental impact of increased storage on the exponential growth rate (*i.e.*, the growth rate once the cell has reached the physiological state corresponding to the n_s after the shift). As long as $M \gg a$, the numerical results of Fig. 7.9 suggest that increasing the storage capacity in order to maximize biomass production during the up-shifts is a convenient strategy. However, when the storage capacity gets too high, the exponential growth rate starts decreasing sharply, hence annihilating the gains coming from higher biomass production during the up-shift. Cells may thus tune their storage capacity as a result of this trade-off.

7.5.3 Environmental dynamics and resource storage strategies

So far, we have only considered one up-shift intensity. For the last series of experiments in the single-cell context, we will now consider several shift intensities: $n_{s,1} - n_{s,0}$ where $n_{s,0}$ and $n_{s,1}$ are respectively the quality of the medium before and after the shift. The results of this type of experiment are as follows.

High storage strategies are favoured by sharper environmental shifts

In Fig. 7.10 (Left), we see that an increase in up-shift intensity results in an increase of the optimal storage capacity α.

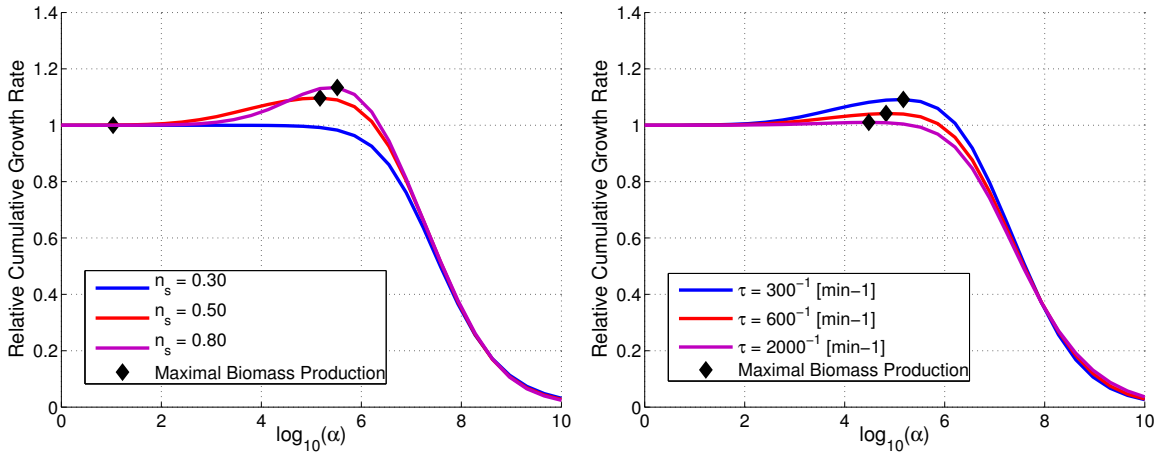


Figure 7.10: (Left) The cumulative growth rate for one given storage capacity α relative to that of when $\alpha = 1$ is maximized for greater storage capacities when the up-shifts are sharper. $\int_{t_0}^{T} \lambda(\alpha) \cdot dt / \int_{t_0}^{T} \lambda(1) \cdot dt$ is evaluated for $t_0 = -20$ and $T = 300$ minutes. The model is shifted from $n_s = 0.12$ to $n_s = 0.30$, $n_s = 0.50$ and $n_s = 0.80$. (Right) The cumulative growth rate for one given storage capacity α relative to that of when $\alpha = 1$ is maximized for lower storage capacities when the frequency τ of up-shift occurrences decreases. The more frequent the up-shifts, the higher the optimal storage capacity that maximises biomass production.

Frequent environmental fluctuations favor high storage

We now consider environments that fluctuate repeatedly, and define the frequency of an environmental shift by $\tau = T^{-1}$. The results of Fig. 7.10 (Right) show that, for high frequencies, growth during transitions becomes non-negligible when compared to exponential steady state growth. Consequently, the optimal storage capacity, *i.e.*, the one that maximizes cumulative growth, thus increases with the frequency of environmental change.

The final observation of the single-cell experiment section justifies our modelling approach, as it relates experimental data to the numerical predictions of our scaling method:

Experimentally observed *E. coli* ATP concentrations fall close to the predicted storage capacity that maximizes biomass production

Living cells such as *E. coli* use ATP as their main energy currency. Because ATP is needed not only for assembling molecules into amino acids (the main building block of proteins), but also for protein elongation during translation, it is considered one of the main protein precursors - as such, in the Weisse model, species a can be considered a proxy for ATP.

In a slow growing *E. coli* cell, which has a volume of approximatively 1×10^{-15} L [179], the average ATP concentration is 1.54×10^{-3} mol.L^{-1} [186], and the number of amino acids - whether they are or not constitutive of proteins- is 5.6×10^{8} [22].

In the Weisse cell model, the mass (counted in amino acids) is 10^8 . Therefore, a slow growing *E. coli model cell* contains $(1.54 \times 10^{-3} \cdot 1 \times 10^{-15} \cdot N_A)/5.6 \approx 1.7 \times 10^5$ ATP molecules per cell model mass, where $N_A = 6.02 \times 10^{23}$ is the Avogadro constant.

It is known that in *E. coli*, the average ATP cost per amino acid is approximatively 25 ATP.aa^{-1} [121]. Therefore, the number of ATP per *cell model mass*, in amino acid cost equivalent, is $(1.7 \times 10^5)/25 \approx 7 \times 10^3$.

This number falls close to the range of protein precursor $a = \alpha \cdot a^*$ which maximizes biomass production in fluctuating environments, as depicted by the results of Figure 7.10; therein, the optimal range is given by $10^4 \cdot a^* < \alpha \cdot a^* < 10^6 \cdot a^*$, where $a^* \approx 1$ under slow growing conditions, and a^* is the quantity of the protein precursor when the storage capacity is $\alpha = 1$.

7.6 The Ecological perspective: Storage Strategies for Competitive Growth

7.6.1 Motivation

The results of Section 7.5 allowed us to study the optimal storage strategy of a cell in different environments, fluctuations of which are modeled as a quasi-instantaneous change in sugar yield (denoted by n_s). Thus, changing the environment "quality" is achieved by varying n_s. More precisely, higher values of n_s are proxies for environments more propitious to growth, as a higher value of n_s means that for an equivalent amount of transporter/metabolic enzymes more protein precursor will be produced per unit of time, which will in turn lead to an increased growth rate in the model.

As one can notice in Table 7.1, the numerical experiments of 7.5 were performed under the assumption of a fuel-rich environment, such that the idealized cell experienced no sugar limitations: $s \gg K_t$, with s denoting the quantity of external nutrient (sugar), and K_t denoting the half-saturate constant of transporters.

In this section, we perform a complementary series of experiments, in order to analyze how storage strategies fare in a *competitive context*. Consequently, we will investigate the impact of storage strategies on the long term growth rate of *a population of cells* in fluctuating environments that *(i)* no longer dispose of unlimited amounts of external nutrient/sugar, and *(ii)* have constant sugar yields, *i.e.,* constant values of n_s. Fluctuating environments will no longer by modelled through a change in sugar yield, but will manifest as fluctuating sugar availability in the medium. We parametrize the sugar availability via the frequency of two superimposed probabilistic series of high and low nutrient/sugar pulses. We test populations of low- and high-storage strategies against each other in such fluctuating environments. Our results indicate the existence of a convex boundary separating a domain characterized by high infrequent pulses,in which the faster growing, low-storage strategies win, from the rest of the considered environments, where the fast growers are driven to extinction by the slow growing, high-storage cells. Suprisingly, this boundary is the only place where lasting co-existence between slow and fast growers can be observed.

7.6.2 Experiment setup: a mixed population model in a fluctuating environment

In this second series of experiments, we model a closed environment in which two types of competing cell populations co-exist:

- "thin" cells, which have low storage capacity: $\alpha = 1$;
- "fat" cells, which have high storage capacity: $\alpha = 2 \times 10^5$.

Each of the two population has an associated *death rate*. We assume an identical death rate for both cell populations, and we set its value to $d_N = 0.01\text{min}^{-1}$, meaning that 1% of each population is removed every minute.

Remark 7.6.1. We stress the importance of constructing a *mixed population* experimental framework, in which both types of cells are present *simultaneously*. If one were to study each type of cell model in isolation, like in the previous section 7.5, the two cell populations would have identical long-term growth rates: if thin and fat cells have the same growth yield n_s, as hypothesized above, and identical death rate, and what's more, if the amount of sugar being delivered in a time interval δt is limiting, both types of cell will have identical average growth rate during δt, for an identical amount of consumed sugar. However, in a *mixed population* setup, different amounts of sugar can be internalized by the two populations, leading to faster overall growth of the cell type that is better at resource assimilation. All in all, it turns out that maximizing growth rate in isolation does not necessarily translate to maximizing growth when more than one type of cells are competing for the same resources [67].

If N_i, with $i \in \{1, 2\}$, denotes the population of thin, respectively fat cells, its growth is given by the ODE:

$$\frac{dN_i}{dt} = N_i \cdot (\lambda_i - d_i). \tag{7.42}$$

The constant external sugar assumption of the initial Weisse model [182] is relaxed, in order to assume that external sugar can be consumed proportionally to the size of the population and its import activity:

$$\frac{ds}{dt} = -\sum_i N_i \cdot \nu_{imp,i}, \tag{7.43}$$

where $\nu_{imp,i}$ denotes the amount of sugar imported per cell of type i per unit of time, and $\frac{ds}{dt}$ is the change in s quantities imputable to import by the cells.

As mentioned above, the competitions are modeled in a closed environment, whose only interference with the exterior world is the addition of medium (sugar). To model the fluctuating availability of sugar in the environment, we introduce two different random nutrient pulses: Every $dt = 1$ minutes, there is a probability p_1 of adding a quantity Q_1 of sugar, and a probability p_2 of adding a quantity Q_2 of sugar. We consider Q_1 to be a *rare* pulse, that yields a high amount of sugar, while Q_2 is considered to be a more frequent, less sugar-rich pulse, that yields I times less sugar than Q_1: $Q_2 = Q_1/I$. Consequently, we refer to Q_1 and Q_2 as *high intensity*, respectively *low intensity* pulses.

We fix the expected number of cells in the environment: $\overline{N_{tot}} = \overline{\sum_i N_i}$. Assuming that all the added sugar contributes to cell growth ($M \gg \alpha \cdot a\star$), the average amount of cells produced per unit of time in a fix environment writes as:

$$\overline{\left(\frac{dN_{Tot}}{dt}\right)_P} = \overline{\left(\frac{d\left(\sum_i N_i\right)}{dt}\right)_P} = \overline{\sum_i N_i \cdot \lambda_i} = \frac{n_s \cdot (p_1 \cdot Q_1 + p_2 \cdot Q_2)}{M} \tag{7.44}$$

At steady state, one has that:

$$\overline{\frac{dN_{tot}}{dt}} = \overline{\left(\frac{dN_{Tot}}{dt}\right)_P} - \sum_i \overline{N_i} \cdot d_i = 0 \tag{7.45}$$

where $(dN_{Tot}/dt)_P$ is the growth term, and $\sum_i \overline{N_i} \cdot d_i$ is the death term.

Assuming identical death rates d_N for every species i, we get that:

$$\overline{\sum_i N_i} = \overline{N_{Tot}} = \frac{n_s \cdot (p_1 \cdot Q_1 + p_2 \cdot Q_2)}{M \cdot d_N} \tag{7.46}$$

For given values of the average number of cells in the environment ($\overline{N_{Tot}}$), and of pulse probabilities (p_1, p_2), the equation above constrains the choice of (Q_1, Q_2) to :

$$p_1 \cdot Q_1 + p_2 \cdot Q_2 = \frac{\overline{N_{Tot}} \cdot M \cdot d_N}{n_s} \tag{7.47}$$

By introducing the intensity ratio parameter constrain $I = Q_1/Q_2$ in Eq. (7.47), one obtains the values of (Q_1, Q_2) required to get an average total number of cells $\overline{N_{Tot}}$, for a fixed intensity ratio between the two pulses I and the frequencies associated to those pulses (p_1, p_2):

$$\begin{cases} Q_2 &= \dfrac{\overline{N_{Tot}} \cdot M \cdot d_N}{n_s \cdot (p_1 \cdot I + p_2)} \\ Q_1 &= I \cdot Q_2 \end{cases} \tag{7.48}$$

For the numerical experiments, we take $I = 100$ and assume that the mean number of cells that can be supported by the environment, *i.e.*, its carrying capacity, is given by $\overline{N_{Tot}} = 100$. The choice of this last parameter is verified *a posteriori* in the *in-silico* experiments, as can be observed in Figure 7.11. We consider a cell population to be extinct whenever its number of cells is 10 times lower than that of its competitor. In experiments for which the extinction condition was not fulfilled by any of the two populations, after an hour (real-time) of simulation, the two populations were considered to co-exist, and the experiment was stopped.

7.6.3 Results

With all this in place, we carried out two types of experiments, which we detail below. In the first experiment, we simulated competition between thin and fat cells, in two environments that differ only in their probability of low-intensity pulses. By setting the probability of high-intensity pulses to a low value, we are able to analyze how storage strategies fare during *starvation periods*. In the second experiment, we carry out competitions in a range of environments that differ through both high and low intensity pulse probability.

7.6.3.1 Experiment 1: Environments with different low-intensity pulse probabilities

We start by carrying out competitions in two different environments, which have an identical probability of type 1 pulses to occur ($p_1 = 10^{-3}$), and differ only by the probability of low intensity pulses ($p_2 \approx 0.11$, respectively $p_2 \approx 0.18$).

From Eq. (7.48), it is apparent that each low intensity pulse with probability $p_2 \approx 0.11$ will deliver $\approx 9.5 \times 10^8$ sugar units, whereas if $p_2 \approx 0.18$, each pulse delivers $\approx 7.1 \times 10^8$ sugar units. As parametrized, the high intensity pulses Q_1 deliver $I = 100$ times that amount.

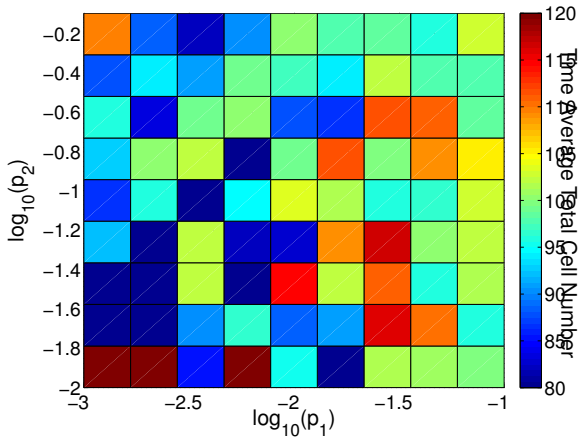


Figure 7.11: Average of the total number of cells along the entire competition experiment. As expected from the constraint we imposed, the average total number of cells sits around 100. The high variability in the average number of cells that appears when decreasing p_1 may be due to the fact that the simulations didn't last long enough so that the total number of cells reaches wide sense stationarity. The time it takes to reach stationarity is indeed expected to be longer as events that significantly impact the number of cells become rarer, as it is the case when the probability of pulse 1 decreases.

Remark 7.6.2. In order to make sense of these numbers, one should compare them to how many sugar units are needed to build a new cell. Since the yield is set to $n_s = 0.5$ across all simulations, it means that 50% of the incorporated sugar will be converted into cell mass. For a cell weighing 10^8 mass units, this means that a low intensity pulse delivers enough sugar to build ≈ 4.7, respectively ≈ 3.5 new cells (again, a high intensity pulse will build 100 times more cells).

Figure 7.12 depicts the results of this experiment. The first observation we were able to make is that in absence of high-intensity pulses (what we refer to as the *starvation period*), the nutrient influx provided by low-intensity pulses of is most often insufficient for cells to grow faster than they die, hence the decrease in total number of cells. This decrease is particularly prominent following a high-intensity pulse, during which the total number of cells grows significantly: after the sugar is exhausted, the large number of resulting cells are left to feed off the limited amount of sugar provided by low-intensity pulses. This behavior is illustrated in Figure 7.12 (Middle Left) - where one can observe the total amount of cells, as well as in Figure 7.12 (Middle Right), which depicts the concomitant dynamics of sugar content. Since the death rate is linearly-dependent on the total number of cells, and since growth depends on sugar availability, it then becomes evident that *the more a population grows following a sugar delivery, the faster it will become extinct when that supply is exhausted.*

The competition outcome depends on the frequency of low-intensity pulses

Our experiments indicate that for the time frame following a high-intensity pulse, survival of cells that have a low storage capacity will be favored. Thus, in Fig. 7.12 (Top Right), one observes an increase in the fraction of $\alpha = 1$ cells that immediately follows high-intensity pulses (which are visible in figure 7.12 (Middle Right) via the sugar spikes). The number of transporters in the same time frame is shown for the two different cell populations, as well as for the two different probabilities of low intensity pulses, in figure 7.12 (Bottom Left) and (Bottom Right) (they correspond to $p_2 \approx 0.11$, respectively $p_2 \approx 0.18$). For the thin cells (low storage capacity), the number of transporters fluctuates considerably less than it does for the fat ones, and this holds for the two different probabilities of low intensity pulses. Also, whereas during starvation periods (defined as time frames that do not follow a high intensity pulse, visible by the sugar spikes in figure 7.12 (Middle Right)), fat cells have more transporters than thin ones, the situation is reversed during periods of nutrient abundance (defined as the time frame that follows a high intensity pulse).

Fat cells allocate their resources better than thin cells

Figure 7.13 shows the probability density plots of the number of transporters as a function of the sugar in the environment for low (Left) and high (Right) storage capacity cells, as well as for $p_2 \approx 0.11$ (Top) and $p_2 \approx 0.18$. Colors indicate the density of data points: red regions map high probability density pairs of {Number of transporter, Amount of sugar in the environment}, whereas blue regions map low probability densities. In contrast with what is shown in figure 7.12, we now consider not only data from a selected time frame, but rather across the entire competition (*i.e.*, until one of the populations goes extinct).

There are three apparent regions. On the very left, one observes the time points at which the sugar was completely depleted from the environment. The second region, further on the right - around $\log_{10}(\text{sugar}) = 9$, corresponds to the data points that immediately follow the addition of a low intensity pulse. The third region, even further on the right, consists of data points that follow a high intensity pulse. There is a consistent pattern, mostly visible for the fattest cell (see Fig. 7.13 (Right)), indicating that the amount of sugar in the environment is inversely proportional to the amount of transporters. This is expected, since in nutrient-abundant environments, cells upregulate their internal allocation to biosynthetic functions, rather than to transport/metabolic functions [182, 167]. Nonetheless, this effect is hardly visible for the thinnest cell - see figure 7.13 (Left)- for which the amount of transporters is relatively constant with respect to variations in the amount of extracellular sugar.

Remark 7.6.3. The reason for which thin and fat cells allocate the resource differently can be understood by conducting the following thought experiment.

Thin cells have low storage capacity, and the amount of protein precursors in poor medium (low n_s) can be as small as 1 per cell, whereas in rich medium (high n_s), it can reach values between 10 and 20.

Consequently, following any kind of sugar pulse (low or high intensity), a thin cell

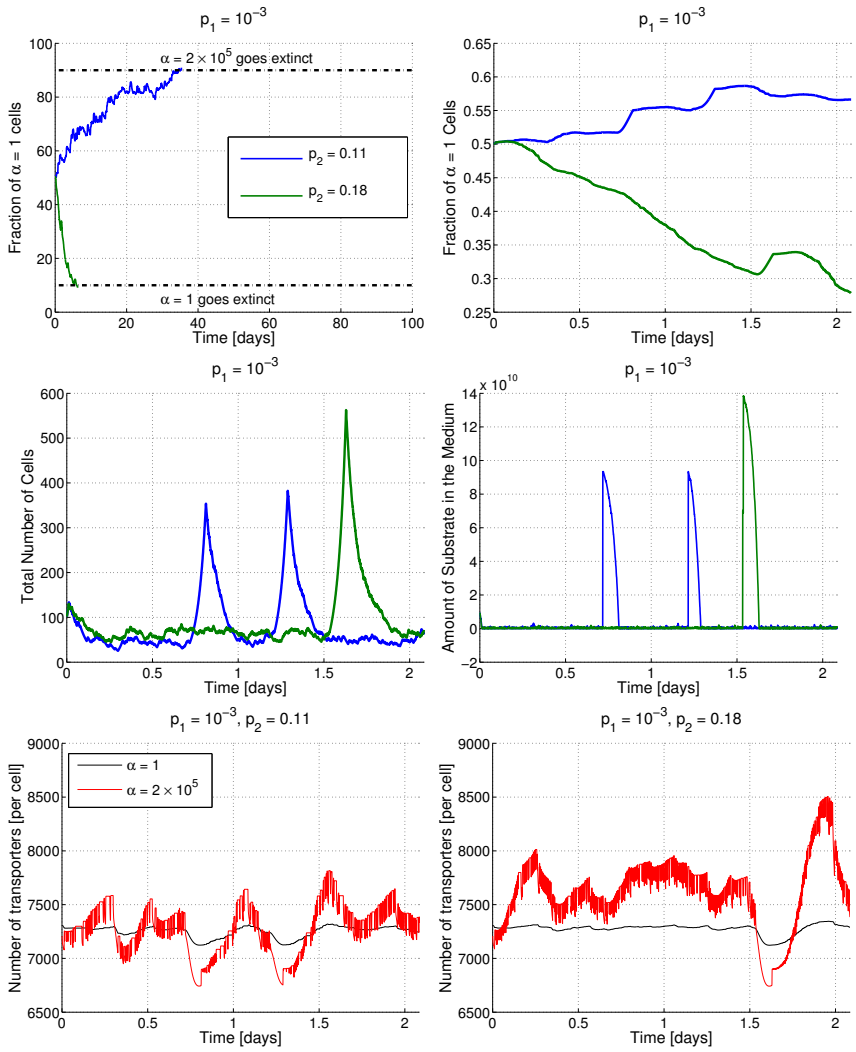


Figure 7.12: Modifying the way an equivalent amount of sugar is delivered in time can result in the extinction of either the fattest or the thinnest cell models. (Top Left) Example of two competitions ran for $p_1 = 10^{-3}$. In the case where $p_2 \approx 0.11$ (probability of low intensity pulses), the fattest ($\alpha = 2 \times 10^5$) go extinct. When $p_2 \approx 0.18$, it is the opposite, and the thinnest ($\alpha = 1$) go extinct. For the rest of the figures, the timed axis is a zoom in on the time interval $[0, 2]$ days from the figure on the left. (Top Right) For this time interval, identically to the left figure, the fraction of $\alpha = 1$ cells is displayed as a function of time. (Middle Left) Concomitant total number of cells as a function of time. (Middle Right) Corresponding amount of sugar as a function of time. (Bottom) Number of transporters in the same time interval for the fattest, in red, and the thinnest, in black, in environments where the probability of type 2 pulses is (Left) $p_2 \approx 0.11$ and (Right) $p_2 \approx 0.18$.

will incorporate enough sugar for it to "sense" the environment as being abundant.

Indeed, consider a low intensity pulse of $Q_2 \approx 7 \times 10^8$ sugar units (for $p_1 = 10^{-3}$ and $p_2 \approx 0.18$), that is distributed uniformly between all cells; then, for a population of 100 cells, each cell will get 7×10^6 sugar units, and consequently build approximatively 3.5×10^6 protein precursors (as $n_s = 0.5$). Because transporters and metabolic enzymes are already present in the cell models, prior to any kind of shift, the influx of a will be fast, and, for the thin cells, will almost instantaneously reach the $a > 10$ cap that maps to an allocation fit for rich environment, which in turn prioritizes biosynthetic processes over transporter/metabolic enzyme production.

One could argue that during starvation periods, thin cells could still shift their allocation to make more transporters, but the almost-null amount of sugar available during this time, coupled with the cells' lack of intracellular storage pools, makes this virtually impossible.

Therefore, we argue that thin cells only produce proteins when nutrients are present in the environment; in this case, they react as if the environment were resource-abundant, and allocate a considerable amount of resources to biosynthetic processes over transporters.

In contrast to their competitors, fat cells are able to adapt their number of transporters to the amount of sugar present in the environment. By having a storage capacity of $\alpha = 2 \times 10^5$, they typically require the amount of protein precursors per cell to be greater than 10^6 (a considerably larger quantity than that demanded by thin cells), in order to have an allocation status that prioritizes biosynthetic processes (*e.g.*, production of ribosomes).

Concretely, fat cells will pass through a continuum of allocation following a sugar pulse, that can be decomposed in three distinct phases: *(i)* for simplicity purposes, assume we start off an extracellular sugar amount equal to 0, as well as a null content in protein precursors. A pulse occurs, and sugar is made available in the environment. The fattest takes up the sugar, but contrary to the thinnest, it will, during the first phase of protein precursor accumulation (as long as it is schematically lower than 10^6), prioritize resource allocation towards transport/metabolic processes. *(ii)* If the supply is large enough, the fattest may build up sufficient amount of protein precursors to start allocating more resources to biosynthetic processes (schematically for amounts of protein precursors greater than 10^6). *(iii)* Once the sugar in the medium is depleted, the amount of protein precursors decreases, and when passing below 10^6, the fattest will again prioritize synthesis of transporters/metabolic enzymes.

In environments in which the sugar supply varies, the fattest may therefore be able to map its allocation status to a time average of the sugar supply, and avoid allocating too much resources to biosynthetic processes in environments that frequently run dry of nutrients.

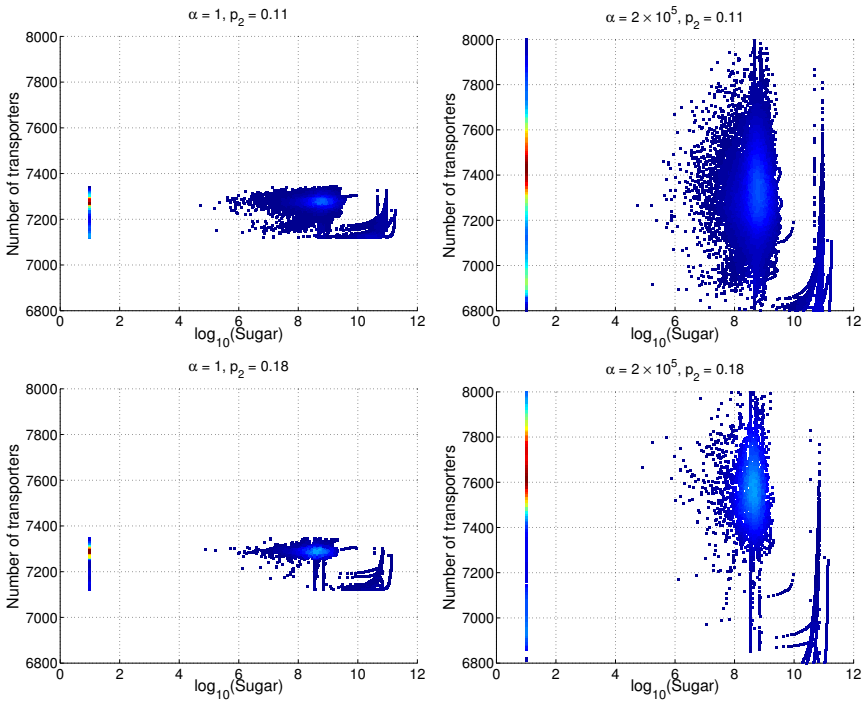


Figure 7.13: Fat cells have a larger dynamic range of their number of transporters. Colors indicate the density of the data points (how frequent a certain number of transporters occurs during the time interval the competition experiment lasts). All the figures correspond to competition experiments during which the probability of type 1 pulse (high intensity pulses) is $p_1 = 10^{-3}$. (Top) The probability of type 2 pulses is $p_2 \approx 0.11$; (Left) for the thinnest, $\alpha = 1$, while (Right) the fattest has $\alpha = 2 \times 10^5$. (Bottom) The probability of type 2 pulses is $p_2 \approx 0.18$, (Left) for the thinnest and (Right) for the fattest. The pairs of probabilities (p_1, p_2) correspond to those used in Figure 7.12. Note that the region left of the plots seems to indicate that the extracellular sugar content is equal to $1 = (\log_{10}(10))^{-1}$. This is actually around 0 in the simulations. We set it to the arbitrary value of 1 for easier plotting, since log scale does not take as input negative values (between -1 and 0), that arise from small numerical errors of the integrator.

7.6.3.2 Experiment 2: Environments with different low- and high-intensity pulse probabilities

So far, we have restricted our analysis to two environments, which differ only through their low-intensity pulse probability. We now systematically observe the outcome of the competition for a 10×10 parameter grid, in which each component corresponds to a different pair of high and low intensity pulse probabilities (p_1, p_2).

A first observation is that thin cells have almost identical amounts of transporters in all studied environments, see Fig. 7.15 (Bottom Left). On the contrary, fat cells have higher amounts of transporters for either high probabilities of low intensity pulses or, more surprisingly, for high probabilities of high intensity pulses, see Fig. 7.15 (Bottom Right).

However, by construction of pulse delivery, increasing the probability of high intensity pulses p_1 means decreasing the amount of sugar delivered per high intensity pulse Q_1, see Eq. (7.48) and Fig. 7.14. This reduces the amount of time during which extracellular sugar content is enough for cells to grow without sugar limitation.

Indeed, if one fixes $p_1 = 0.1 and p_2 = 0.01$, then by Eq. (7.48), the amount of sugar delivered per high intensity pulse is $Q_1 \approx 2 \times 10^9$, which is enough to build 10 new cells and is almost comparable to a low intensity pulse when $p_1 = 10^{-3}, p_2 \approx 0.11$, where the amount of sugar delivered per low intensity pulse is $Q_2 \approx 9.5 \times 10^7$.

Thus, for high p_1, the distribution of sugar in time is similar to that of an environment bearing low p_1 and high p_2 (albeit, without the long spawned abundance periods).

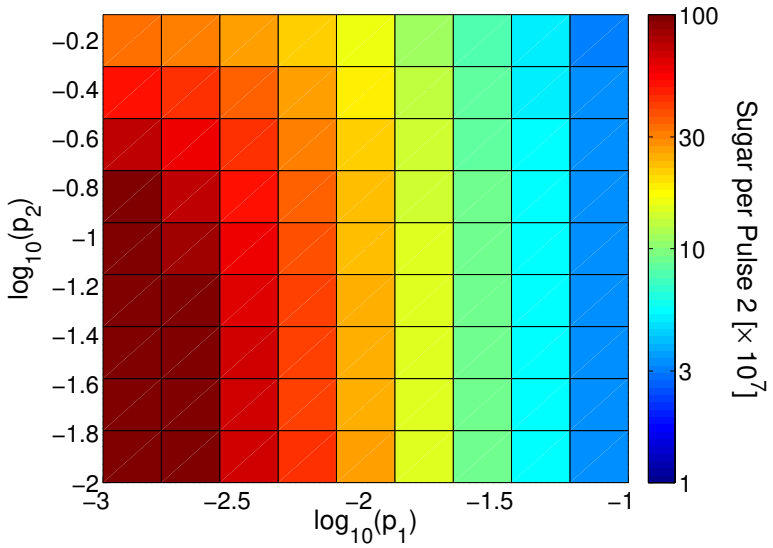


Figure 7.14: Quantity of sugar added per pulse of type 1 as a function of (p_1, p_2), the two pulses' probabilities, upon the constraints that: *(i)* the average amount of sugar provided per minute is $k_{in} = 2 \times 10^8$ and *(ii)* that $Q_1 = I \cdot Q_2$, with $I = 100$.

Transporter abundance predicts the competition outcome

A surprising observation is that the mean number of transporters of the fat cells, as depicted in Fig. 7.15 (Bottom Right), seems to entirely predict the outcome of the competition. This can be observed in Fig. 7.15 (Top Right), for which the convention is as follows: in blue, the fattest wins, in red the thinnest wins, in green, the numerical experiments were interrupted due to excessive computation time, and the two populations are assumed to co-exist.

Indeed, the simple rule of thumb: *"whichever type of cell synthesizes the most transporters wins the competition in an environment where the nutrient supply is limiting"*, is verified empirically.

Although this seems to hold mathematically and experimentally at steady state - see 7.6.4-, it is not trivial to see why that should be the case in a fluctuating environment.

Remark 7.6.4. We note that the growth rate, as defined in Eq. (7.13), depends on the yield factor n_s, the amount of sugar s, the number of transporters e_t, and the kinetic properties of the transporter (K_t and v_t), at steady state.

Under the assumption that different types of cells, competing in the same environment, have the same yield n_s and the same type of transporters (identical K_t and v_t), and given that the amount of sugar in the environment is the same for all cells, it then follows that their growth rate is actually parametrized by the amount of transporters they express *at steady state*.

This implies that at steady state, in a competition between two cell models having negligible storage capacities, the one that allocates the most of its mass fraction to transporters will win the competition, in accordance with [67, 182].

It is important to emphasize that this does not mean that a cell should only express transporters, and not ribosomes. Indeed, if a cell were to do this, the sugar transported inside the cell could not be consumed by biosynthetic processes (translation by ribosomes) at a sufficient rate, which would lead to an accumulation of metabolites until steady state is reached, *i.e.,* a state in which the rates of transport and biosynthetic processes balance out.

This could paradoxically lead to a lower amount of transporters per mass unit, since a higher fraction of the mass would be occupied by metabolites.

The fattest is the fittest in nutrient poor environments

Perhaps a good explanation for why storers outcompete fast growers in some conditions (*i.e.*, nutrient poor environments) - see Figure 7.15 (Top Right), is that the latter run out of resources suddenly and keep a frozen image (in their internal resource allocation) of an abundant past which has ceased to exist - consequently expressing an insufficient amount of transporters. Storers, on the other hand, have a much more sensible picture of the conditions which are prevalent in their environment.

Considering two types of fluctuations of different intensities reveals an intricate competitive landscape. Coexistence of the two species can be long, see figure 7.15 (Top Left), notably on

the convex boundary, which is evocative of phenomena of critical slow downs near bifurcations of dynamical systems. The result of the competition is quite different from what one would expect by exclusively considering the growth of the two populations in isolation.

Indeed, for subtle reasons, the apparent competitive disadvantage of the fattest is compensated by mechanisms that appear to be orthogonal to the ones explored in Section 7.5 (i.e. higher growth rate during transitions from one growth medium to another), and sometimes lead to extinction of the low-storage fast growers.

Again, it is interesting to emphasize that the trade-offs between storage and growth are emergent properties that build up on well known biological mechanisms – which is different from a game theoretical approach that would consist in implementing this trade-off in an exogenous way.

For a more thorough numerical investigation, one would need to sample behaviors with a finer grain, as well as explore other temporal patterns for nutrient delivery.

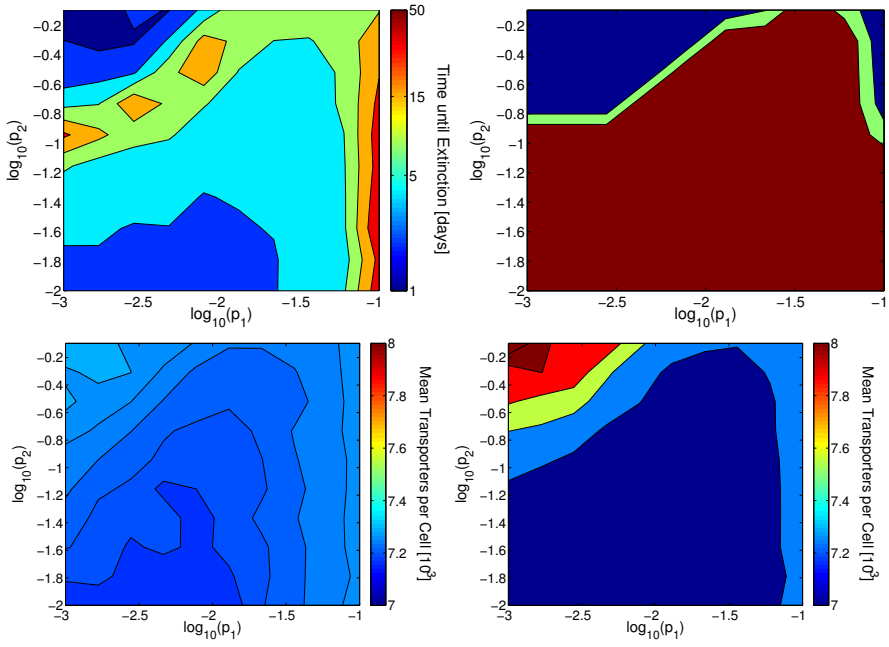


Figure 7.15: The distributions of the low and high intensity pulses underly various differences in the number of transporters between fat and thin cells. This difference in the number of transporters predicts which of the fattest or the thinnest will win the competition. (Top Left) Time until one of the two population goes extinct. (Top Right) The red area depicts that the population of the fattest (high storage capacity) goes extinct, whereas the blue area means that the thinnest (low storage capacity) goes extinct. The green area corresponds to environments wherein, after 1 hour (real time) of simulations, no population was extinct. (Bottom) Number of transporters across the different types of environments (characterized by the frequencies of the pulses 1 and 2) for (Left) the thinnest and (Right) the fattest.

7.7 Conclusion

In the work presented in this chapter, we investigated the impact of different resource storage strategies on cellular growth, using a recent mathematical model (the Weisse model) of the coarse-grained mechanisms that drive cellular growth, which we modify slightly in order to clarify some assumptions relative to growth and cellular mass.

In order to model resource storage, we introduce a novel technique of reparametrization of deterministic models of Biochemical Reaction Networks, which allows one to tune the concentration of certain chemical species, while preserving the network's behavior at steady state. This, in turn, allows one to symbolically navigate natural lines of iso-cost in parameter space, provided cost functions only depend on steady-state constraints. When applied to the Weisse model, the scaling method guarantees by construction that various storage strategies preserve approximatively goodness-of-fit to the original data, and therefore correctly match growth conditions to sectorial resource allocation.

In a first series of experiments, we analyze how the level of resource storage can impact cellular growth, and thus impose fitness costs or benefits on cells, for a *single cell* model. We first test different storage strategies in different environments, parametrized by their respective nutritional efficiency coefficient n_s, and find that there is a cost associated with high levels of storage, resulting from the dilution-driven loss of stored resources. Our results also show that there is large window of concentrations for the universal precursor (considered as a conflation of all the various energy and carbon needs of the cell) in which the growth rate is essentially unchanged at steady state. This first experiment suggests that constant environments favour low levels of resource storage, as no additional fitness can be obtained by storing.

In order to extend our analysis to fluctuating environments, we implement shifts in the sugar yield by modulating the nutritional coefficient parameter n_s. We find that high levels of storage can benefit cells in variable environments, by increasing biomass production during transitions from one medium to another. Consequently, fluctuating environments appear to favour high levels of resource storage, *i.e.*, additional fitness can be obtained by storing, with little detrimental effect on stationary growth.

These two results open up a significant possibility for selective pressure driving cells to a judicious choice of storage level, in order to extract the benefits during shifts in nutrient availability and quality.

We also find that the cost of storage appears to increase as environments become more favourable to growth. This can justify the use of regulatory systems that tune the storage capacity according to growth conditions: in the case of *E.coli*, it has been shown that proteins involved in glycogen (the main *E.coli* storage molecule) synthesis are up-regulated in poor growth conditions [61].

We follow up on the single cell model analysis by performing a series of complementary experiments, aimed at investigating how storage strategies fare in a *competitive context* as well as on how the operational impact of intermittent energy sources may be mitigated by storage of immediate growth precursors. We argue that while most biologists focus on studying organisms in environments where nutrient supply is virtually infinite, and thus *steady* (*e.g.*, during the exponential phase of a growth culture), these conditions do not reflect the environments most

living systems actually experience.

In that sense, our rescaling technique allows us to generate a family of models that behave differently when nutrient availability fluctuates - but which are nearly indistinguishable when environmental conditions are steady. In this second series of experiments, fluctuations in the availability of substrates fueling the biosynthetic process are entirely exogeneous to the cell, and are parametrized by two superimposed probabilistic trains of high, respectively low, pulses of sugar [8]. We test populations of low- and high- storage strategies against each other in a variety of such fluctuating environments. Our results suggest the existence of a convex boundary that separates a domain of high and infrequent sugar pulses regimes - in which the fast growing, low storage cells survive-, from the rest of the environmental landscape - in which the fast growers are driven to extinction by the high-storage, slow growers.

Another, peculiarly interesting, yet completely unexpected behavior of the model is that storage seems to allow cells to "measure" the amount of resources in the environment on longer time scales. This behavior is reminiscent of the concept of memory and it provides a mechanistic explanation of how storage could be used as a primitive source of information about the life history of an organism: one could hypothesize that it may have provided an incentive for living systems to evolve to greater size over evolutionary time scales.

All in all, our analysis suggests that cells may face trade-offs in their maintenance of resource storage, based on the frequency of environmental change. Quantitative models such as the one we considered here could be used to reverse-engineer the ecology of cellular species — much of which we only know in laboratory conditions —, using existing data about metabolic resource concentrations [15].

Finally, although our analysis was performed on a model for unicellular growth, the mechanisms described may be applicable to any system growing in an environment where resources are fluctuating and scarce.

[8] as opposed to the first series of experiments, where environmental fluctuations were modeled as variations in the sugar yield, herein we use constant mean amounts of sugar per time unit

Chapter 8
Abstractions for cellular growth: towards a new Petri net semantics

The work presented in this chapter has been published in the proceedings of the 5th Hybrid Systems Biology (HSB) workshop [11].

8.1 Motivation

The previous discussion of Chapter 7 with respect to the incorporation of cellular growth in classical models of Biochemical Reaction Networks motivates a deeper analysis of the subject. We are thus now interested in working towards a novel executable model of biochemical reaction networks, in which cellular growth can be characterized as an emergent property of the execution semantics, thus circumventing the need of adding artificial constructs such as dilution reactions to a growth model.

As seen in Section 4.1, Biochemical Reaction Networks can be represented using Petri Nets, in a straightforward manner: place denote species, while transitions denote reactions. For general Petri nets, the execution of the net proceeds according to the usual *interleaving semantics*: at each execution step, an enabled transition is selected non-deterministically, then fired. The interleaving semantics describes totally asynchronous behavior, and consequently fails to capture the inherently concurrent nature of cellular behavior, in which multiple reactions can happen in parallel. What's more, intracellular processes rarely work in isolation, but rather in continuous interaction with the rest of the cell. The cell has finite resources, so committing resources to one task reduces the amount of resources available to others. All cells experience these trade-offs, which potentially modify all cellular processes, but are often overlooked.

We have also seen that in the case of "biochemically interpreted" Petri nets, the usual chemical kinetics (be it stochastic or deterministic) is imposed as the execution semantics: transitions in stochastic Petri nets are executed according to the standard firing rule of the interleaving semantics, while continuous Petri nets execution proceeds according to the system of ODEs imposed on the net by the firing rate function. However, one could argue that the construction of these net dynamics demeans artificial, and limited in scope: with it in place, Petri nets are nothing more than a structured description of the biochemical reaction networks they model. In this case, the existing structural Petri net analysis methods can be harnessed for

biological purposes, but no supplementary information (with respect to the usual deterministic or stochastic model execution) about the biological system is gained via the execution semantics.

Consequently, in this chapter, we work towards defining a general Petri net execution semantics that recreates the mass-action dynamics of deterministic BRNs models. More specifically, our method consists in a piecewise synchronous approximation of the dynamics of a "growth" [1] BRN: a resource-allocation-centered Petri net with maximal-step execution semantics. With our method in place, the trade-offs caused by finite resource allocation between reactions are put front stage, and the mass action run of the system can be rephrased as an optimization problem: the inter-phase between synchronous runs defines an unknown, the *resource split*, that can be exploited in order to find the best split (e.g., the one which minimizes parallel completion time, or maximizes growth rate).

For BRNs comprised exclusively of unimolecular reactions, we prove the correctness of our method and show that it can be used either as an approximation of the dynamics, or as a method of constraining the reaction rate constants or refuting models. Indeed, we show that for unimolecular reaction networks, the proposed PN execution semantics enables the definition of a formal notion of growth rate, which can serve as an improved "biomass objective function"[142] for a constraint method similar to flux balance analysis (FBA)[142].

Related work. While most intracellular growth processes are well characterized, the manner in which they are coordinated under the control of a scheduling policy is not well understood. When fast replication is sought, a schedule that minimizes the completion time is naturally desirable. But when resources are scarce, in the worst case it is computationally hard to find such a schedule [72],[169]. The scheduling problem of a self-replicating bacterial cell is studied in [153].

The concept of maximally parallel execution already appears in the literature on P-systems [144], and in Levy's family reductions [115], while in [110], the authors use it to develop a Petri Net execution semantics that resembles biology. The scheduling policy of cells is also tackled in [65], where the notion of *bounded asynchrony* is introduced. In [148], the author introduces a constraint method that generalizes FBA to the stochastic case, allowing models to be discriminated using second order moments.

This chapter is organised as follows. In the next section, we first define the max-parallel execution of a PN, after which we introduce the piecewise synchronous execution semantics and show it encompasses max-parallel execution. We then demonstrate that, at least in the case of unimolecular reactions, our method recreates the usual ODE system dynamics, and that it can be used either as an approximation of the dynamics, or as an alternative to flux balance analysis. The final section concludes with a summary and outlook regarding further work on the subject.

[1] we associate growth with the existence of an exponential stationary phase

8.2 Split-Burst: towards a piecewise-synchronous execution of Biochemical Reaction Networks

From now on, assume a Petri net $\mathcal{N} = (P, T, f, m_0)$ that models a given BRN $\mathcal{A} = (\mathcal{S}, \mathcal{R}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ with initial species abundance given by the vector $\boldsymbol{x}_0$, *i.e.* the components of $\mathcal{N}$ verify the conditions:

(i) $P = \mathcal{S}$;

(ii) $T = \mathcal{R}$;

(iii) $\forall (p,t) \in P \times T, \qquad f(p,t) = \alpha_{tp}$;

(iv) $\forall (t,p) \in T \times P, \qquad f(t,p) = \beta_{tp}$;

(v) $m_0 = \boldsymbol{x}_0$.

The incidence matrix of $\mathcal{N}$ writes as:

$$\nabla \equiv \nabla^+ - \nabla^-,$$

with $\nabla^+ \equiv \boldsymbol{\beta}^T$, and $\nabla^- \equiv \boldsymbol{\alpha}^T$.

8.2.1 Max-parallel execution semantics of Petri Nets

In [110], the authors introduce a Petri net execution semantics that accounts for the inherent concurrency of biological systems, and as such, proves to be better suited for the modeling of BRNs than the usual interleaving semantics: the *max-parallel* execution semantics, which can be described informally as "execute greedily as many transitions as possible in one step"[110].

This description allows us to provide a formal definition of the max-parallel execution:

Definition 8.2.1 (**Max-parallel execution step of a Petri net**)
A max-parallel execution step of the Petri net $\mathcal{N}$ at state $\boldsymbol{m}$ is a positive T-vector $\boldsymbol{v}$ such that:

1. *v is **compatible** with m (i.e., there are enough tokens to do everything, in any order):*

$$\mathbf{0} \leq \boldsymbol{m} - \nabla^- \boldsymbol{v}$$

2. *v is **exhaustive** (i.e., no reaction is enabled after firing):*

$$\forall j \in T, \boldsymbol{m} - \nabla^- \boldsymbol{v} \not\geq \nabla^-(:, j),$$

where $\nabla^-(:, j)$ denotes the j^{th} column of ∇^-.

For example, Figure 8.1 depicts the Petri net of the BRN (we ignore the products):

$$\begin{aligned} 3A + 2B &\xrightarrow{\kappa_0} \cdots \\ 5B + 3C &\xrightarrow{\kappa_1} \cdots \\ C &\xrightarrow{\kappa_2} \cdots \end{aligned}$$

with initial marking $m_0 = (9, 9, 9)$, and its possible max-parallel strategies.

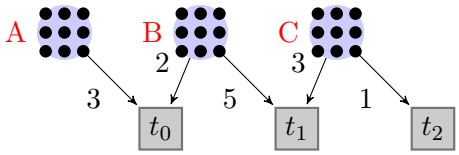


Figure 8.1: A network with exactly 2 possible maximally parallel steps: $\{t_0 \times 3, t_2 \times 9\}$ and $\{t_0 \times 2, t_1, t_2 \times 6\}$

8.2.2 Piecewise-synchronous execution semantics

In order to deal with cellular resource allocation, we construct a Petri net execution semantics that is piecewise synchronous (and which includes the maximally parallel strategies): among all traces of execution of a PN, we single out a subset of semi-synchronous ones.

Execution proceeds in an alternation of resource allocation (*"split"*) and depletion (*"burst"*). Between the depletion step and the next allocation, we add a phase of collection of products.

Allocation of tokens to their possible transitions is done via a $|T| \times |P|$ matrix $\boldsymbol{\alpha}^{\boldsymbol{split}}$, where $\boldsymbol{\alpha}^{\boldsymbol{split}}_{ij}$ denotes the fraction of resource j being allocated to reaction i, meaning that:

$$\forall j \in P, \qquad \sum_{i \in T} \boldsymbol{\alpha}^{\boldsymbol{split}}_{ij} \leq 1 \tag{8.1}$$

The "burst" phase consists of the execution of all transitions in parallel, until the available input are reduced to a small constant fraction of the initial amount (for reasons explained below). This small constant remainder we impose on our semantics is both the reason for the inequality sign in (8.1), and the reason our executions will be in the spirit of max-parallel executions, rather than max-parallel in the strict sense: whereas the max-parallel execution seeks to deplete all available resources, ours consumes them up to a fixed level (noted ϵ in the following).

Resource allocation: relation to max-parallel execution

For a PN $\mathcal{N}$, assume $\boldsymbol{\alpha}^{\boldsymbol{split}} \in \mathbb{R}_+^{|T| \times |P|}$ a resource allocation matrix defined as above, $\boldsymbol{m} \in \mathbb{R}^{|P| \times 1}$ a marking of $\mathcal{N}$ (i.e., a resource array), and $\boldsymbol{v} \in \mathbb{R}^{|T| \times 1}$ a (potentially max-parallel) reaction vector. We note that zero-order reactions ($\emptyset \to \ldots$) are not taken into account, as the question of resource allocation does not apply to them.

We define the operation $\star$ as:

Definition 8.2.2

$$(\boldsymbol{\alpha}^{\boldsymbol{split}} \star \boldsymbol{m})_j \equiv \min_{i \in P} \left(\frac{\alpha_{ji}}{\nabla^-_{ij}} \cdot m_i \right)$$

Then Theorem 8.2.1 states that our execution semantics encompasses the max-parallel strategy (each max-parallel strategy is associated with a resource allocation matrix $\boldsymbol{\alpha}^{\boldsymbol{split}}$:

Theorem 8.2.1
For every reaction vector $\boldsymbol{v}$ that is compatible with a resource vector $\boldsymbol{m}$ (and potentially max-parallel), there exists a resource allocation matrix $\boldsymbol{\alpha}^{\boldsymbol{split}}$ such that:

$$\boldsymbol{v} = \boldsymbol{\alpha}^{\boldsymbol{split}} \star \boldsymbol{m}$$

Furthermore, if the BRN modeled by $\mathcal{N}$, is unary, there is unicity of $\boldsymbol{\alpha}^{\boldsymbol{split}}$.

Proof. Given a resource array $\boldsymbol{m}$, the (potentially max-parallel) resource vector $\boldsymbol{v}$ is compatible with $\boldsymbol{m}$ iff

$$\nabla^{-}\boldsymbol{v} \leq \boldsymbol{m} \tag{8.2}$$

Then we can construct $\boldsymbol{\alpha}^{\boldsymbol{split}} \in \mathbb{R}_{+}^{|T|\times|P|}$:

$$\boldsymbol{\alpha}_{ji}^{\boldsymbol{split}} \equiv \frac{\nabla_{ij}^{-} \cdot v_j}{m_i} \tag{8.3}$$

s.t.

$$\begin{aligned}(\boldsymbol{\alpha}^{\boldsymbol{split}} \star \boldsymbol{m})_j &= \min_{i\in P}\{\frac{\alpha_{ji}}{\nabla_{ij}^{-}} \cdot m_i\} \\ &= \min\{\frac{\nabla_{ij}^{-} \cdot v_j}{m_i} \cdot \frac{m_i}{\nabla_{ij}^{-}} \mid \nabla_{ij}^{-} \neq 0\} \\ &= v_j\end{aligned}$$

Furthermore, cf. (8.2), $\forall j \in P : \sum_{i\in T} \boldsymbol{\alpha}_{ij}^{\boldsymbol{split}} = \frac{\sum_{i\in T} \nabla_{ji}^{-} \cdot v_i}{m_j} \leq 1$, *i.e.* $\boldsymbol{\alpha}^{\boldsymbol{split}}$ is indeed a resource-allocation matrix.

If all reactions of the BRN are unimolecular, then :

$$\forall j \in T, \exists! i_j \in P \text{ s.t. } \nabla_{i_j j}^{-} \neq 0 \implies \forall j \in T, (\boldsymbol{\alpha}^{\boldsymbol{split}} \star \boldsymbol{m})_j = \frac{\alpha_{ji_j}}{\nabla_{i_j j}^{-}} \cdot m_{i_j} \tag{8.4}$$

hence the uniqueness of α. ■

For bimolecular reactions, $\boldsymbol{\alpha}^{\boldsymbol{split}}$ defined as above is no longer the unique solution of $\boldsymbol{v} = \boldsymbol{\alpha}^{\boldsymbol{split}} \star \boldsymbol{m}$; intuitively, for a bimolecular reaction $r_k : A + B \to \ldots$, a different resource allocation matrix $\boldsymbol{\alpha'}$ can be created by allocating to r_k whatever amount of species A is not allocated in $\boldsymbol{\alpha}^{\boldsymbol{split}}$. The use of min in Def. 8.2.2 ensures that $v = \alpha' \star m$.

Example 8.2.1
Reconsider the BRN of Figure 8.1: $\boldsymbol{m} = \begin{bmatrix}9\\9\\9\end{bmatrix}$, $\nabla^{-} = \begin{bmatrix}3 & 0 & 0\\2 & 5 & 0\\0 & 3 & 1\end{bmatrix}$, *and* $\boldsymbol{v} = \begin{bmatrix}2\\1\\6\end{bmatrix}$, *and one of the 2 possible maximally parallel steps:* $\{t_0 \times 2, t_1, t_2 \times 6\}$.

Then, $\exists \boldsymbol{\alpha}^{\boldsymbol{split}} = \begin{bmatrix} \frac{6}{9} & \frac{4}{9} & 0 \\ 0 & \frac{5}{9} & \frac{3}{9} \\ 0 & 0 & \frac{6}{9} \end{bmatrix}$*, defined as in (8.3), s.t.*

$$\boldsymbol{\alpha}^{\boldsymbol{split}} \star \boldsymbol{m} = \begin{bmatrix} \frac{6}{9} \cdot 9 \cdot \frac{1}{3} \wedge \frac{4}{9} \cdot 9 \cdot \frac{1}{2} \wedge \infty \\ \infty \wedge \frac{5}{9} \cdot 9 \cdot \frac{1}{5} \wedge \frac{3}{9} \cdot 9 \cdot \frac{1}{3} \\ \infty \wedge \infty \wedge \frac{6}{9} \cdot 9 \cdot \frac{1}{1} \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \\ 6 \end{bmatrix} = \boldsymbol{v}.$$

By re-allocating the excess of species A to the first reaction, we get $\boldsymbol{\alpha'} = \begin{bmatrix} 1 & \frac{4}{9} & 0 \\ 0 & \frac{5}{9} & \frac{3}{9} \\ 0 & 0 & \frac{6}{9} \end{bmatrix}$*,*

a resource-allocation matrix that also verifies

$$\boldsymbol{\alpha'} \star \boldsymbol{m} = \begin{bmatrix} 1 \cdot 9 \cdot \frac{1}{3} \wedge \frac{4}{9} \cdot 9 \cdot \frac{1}{2} \wedge \infty \\ \infty \wedge \frac{5}{9} \cdot 9 \cdot \frac{1}{5} \wedge \frac{3}{9} \cdot 9 \cdot \frac{1}{3} \\ \infty \wedge \infty \wedge \frac{6}{9} \cdot 9 \cdot \frac{1}{1} \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \\ 6 \end{bmatrix} = v,$$

hence the non-uniqueness of $\boldsymbol{\alpha}^{\boldsymbol{split}}$ *in the bimolecular case.*

8.3 Linear Reaction Networks

8.3.1 Growth rate as an emergent property

Consider a BRN comprised exclusively of unimolecular reactions. Then, for $\boldsymbol{m}$ an initial marking, ∇ the composite change matrix, and $\boldsymbol{\alpha}^{\boldsymbol{split}}$ a resource allocation matrix, the state of the system after one execution with the $\boldsymbol{\alpha}^{\boldsymbol{split}}$ split is given by the matrix
$(I + \nabla \cdot \boldsymbol{\alpha}^{\boldsymbol{split}}) \cdot \boldsymbol{m}$, with I the identity matrix.

More generally, after k iterations of the "split-burst" execution with the same split $\boldsymbol{\alpha}^{\boldsymbol{split}}$, the state of the system is:

$$D_\alpha^k \cdot \boldsymbol{m}, \text{ with } D_\alpha = I + \nabla \cdot \boldsymbol{\alpha}^{\boldsymbol{split}} \tag{8.5}$$

Let $\lambda_1 > \lambda_2 > \cdots$, the eigenvalues of D_α, and $E(\lambda_i)$ the eigenspace associated to each λ_i. If the initial marking vector can be decomposed as $\boldsymbol{m} = \sum_i \boldsymbol{m_i}$, with $\boldsymbol{m_i} \in E(\lambda_i)$, then we can rewrite:

$$D_\alpha^k \cdot \boldsymbol{m} = \lambda_1^k \cdot [\boldsymbol{m_1} + \sum_{i \geq 2} (\frac{\lambda_i}{\lambda_1})^k \cdot \boldsymbol{m_i}] \tag{8.6}$$

If $\lambda_1 < 1$, given (8.6), the system will eventually go extinct. Also, if $\boldsymbol{m_1} \in E(\lambda_1)$ is not unique, one has redundancy of growth (i.e., growth on multiple species/sources). We thus assume that $\lambda_1 > 1$, alongside uniqueness of $\boldsymbol{m_1}$ and unidimensionality of eigenspaces.

Under these assumptions, as $\frac{\lambda_i}{\lambda_1} < 1$, for a big-enough k, the state of the system will converge to $\lambda_1^k \cdot \boldsymbol{m_1}$, meaning that the growth rate of the system is given by λ_1, the biggest eigenvalue of D_α.

8.3.2 Synchronous execution

Depletion time of unimolecular reactions

Consider the following unimolecular reaction:

$$S_i \to \ldots @k,$$

for which the time evolution of the concentration of species S_i is given by the ordinary differential equation

$$\frac{d[S_i]}{dt} = -k \cdot [S_i]$$

Then, at time t, the concentration of species S_i is:

$$[S_i](t) = [S_i](0) \cdot e^{-kt}$$

Equivalently, the mean time of depletion of the reaction (i.e., bringing the level of species S_i to a specified amount $0 < s_i \leq [S_i](0)$) is given by:

$$\tau = k^{-1} \log\left(\frac{[S_i](0)}{s_i}\right) \tag{8.7}$$

We note that s_i is a convention (the remainder of the reaction), or rather $\frac{[S_i](0)}{s_i}$ is the relative amount we consume off the input (e.g., $s_i = 1\%S_i(0)$); the point being that we cannot deplete the whole amount of S_i, as that would take time $\tau = \infty$. Herein lies the main difference between our method and max-parallel execution.

Now suppose n unary reactions with the same input :

$$S_i \to \ldots @k_j, \tag{8.8}$$

with $j \in N_i, |N_i| = n$.

We then allocate in parallel the available amount of species S_i between the n reactions, according to our execution semantics: $\forall j \in N_i$, reaction j receives $\alpha_{ji} \cdot [S_i](0)$ input, and has a remainder of $s_{i,j}$.

Then, the depletion time of reaction j is given by:

$$\tau_j = k_j^{-1} \cdot \log\left(\alpha_{ji} \cdot \frac{[S_i](0)}{s_{i,j}}\right) \tag{8.9}$$

Isochronicity and iso-remainder assumptions

In order to have a synchronous execution, we fix the same depletion time, τ for all reactions of the unary CRN.

Then, from (8.9):

$$\alpha_{ji}^{split} = \beta^{k_j} \cdot \epsilon_{i,j}, \tag{8.10}$$

where $\beta = e^{\tau}$ and $\epsilon_{i,j} = \frac{s_{i,j}}{[S_i](0)}$.

In this notation, $\epsilon_{i,j}$ is the remaining *percentage* (relative amount) of the total amount of S_i available in the beginning of the split round (i.e., $[S_i](0)$), after reaction j is executed.

Furthermore, if we assume the same relative amount, s_i, remains after executing all n reactions, we have that:

$$\forall j \in N_i, \alpha_{ji}^{split} = \epsilon_i \cdot \beta^{k_j}, \tag{8.11}$$

with $\epsilon_i = \frac{s_i}{S_i(0)}$.

Under these assumptions, the dynamics of the system in state $\boldsymbol{m}$, for species S_i, is given by:

$$\Delta m(\tau) \equiv \nabla \cdot (\alpha_{\_i}^{split} - \epsilon_{\_i}) \cdot \boldsymbol{m}, \tag{8.12}$$

where $\alpha_{\_i}^{split}$ denotes the i^{th} column of the resource allocation matrix $\boldsymbol{\alpha^{split}}$:

$$\forall j \in T, \alpha_{ji} = \begin{cases} \epsilon_i \cdot \beta^{k_j}, & \text{if } j \in N_i \\ 0, & \text{otherwise} \end{cases},$$

and

$$\forall j \in T, \epsilon_{ji} = \begin{cases} \epsilon_i, & \text{if } j \in N_i \\ 0, & \text{otherwise} \end{cases}.$$

Then, from (8.1) [2] and (8.11), we have that:

$$\epsilon_i = \frac{1}{\sum\limits_{j \in N_i} 1 + \beta^{k_j}},$$

and we can define:

$$\begin{aligned} \hat{\alpha}_{S_i}(\tau, \hat{k_{S_i}}) &\equiv [\alpha_{\_i} - \epsilon_{\_i}] \\ &= [\frac{e^{\tau \cdot k_j} - 1}{\sum_j 1 + e^{\tau \cdot k_j}}] \end{aligned} \tag{8.13}$$

with $[\frac{e^{\tau \cdot k_j} - 1}{\sum_j 1 + e^{\tau \cdot k_j}}]$ denoting the T-vector that has 0 in the components representing reactions $j \notin N_i$.

Then, by injecting equation (8.13) into equation (8.12), and by using the fact that when $x \to 0, e^x \approx 1 + x$, one can easily observe that when $\tau \to 0$, $\Delta m(\tau)$ recreates the usual ODE dynamics for unimolecular BRNs:

[2] the inequality of (8.1) is here explicitly expressed via the remainder ϵ : $\sum_{i \in T} \alpha_{ij} \leq 1$ is the same as $\sum_{i \in T} (\alpha_{ij} + \epsilon_j) = 1$

$$\frac{\Delta m}{\tau} \approx \nabla \cdot \hat{\boldsymbol{k}} \cdot \boldsymbol{m} \tag{8.14}$$

8.4 Possible Applications

Based on formulae given in (8.6) and (8.14), our method can be interpreted either as an approximation of the real system's dynamics (and be used for simulation purposes), or in an abstract way, as an alternative to flux balance analysis.

8.4.1 Approximation of system dynamics

As an approximation of the dynamics, under the unary/isochronous/iso-remainder assumptions, ours is a temporised discrete execution dynamics, that, when $\tau \to 0$, recreates the usual ODE dynamics. If we fix τ the execution time-step, and $\hat{\boldsymbol{k}}$ the reaction rate vector, we can determine $\boldsymbol{\alpha}^{\boldsymbol{split}}$, the resource allocation matrix, and ϵ, the remainder percentage (cf. (8.13)).

We note that the "iso-" assumptions represent a way of decoupling production and consumption in the biochemical network, in the spirit of Karr's modular systems [104]; intuitively, it can be interpreted as: "in a parallel execution of a reaction set, there is no waiting for the slowest reaction to complete".

As a simulation method, it can be viewed as a big-step approximation of an integrator, resembling a deterministic τ-leaping [76].

8.4.2 Synchronous Balanced Analysis

Conversely, if the resource allocation matrix α is fixed, our execution semantics can be interpreted as an alternative to Flux Balance Analysis (FBA)[142], in order to determine the limitations of a metabolic system. This is a second possible utility of the "split-burst" Petri net execution semantics, and arguably the most important application of our method.

Flux Balance Analysis is a widely used constraint-based mathematical approach for analyzing the flow of metabolites through a metabolic network, thereby making it possible to predict the growth rate of the system. The constraints lying at the heart of FBA come from two sources. Firstly, the system is assumed to be at steady state, a condition which is translated into flux (or mass) balance constraints imposed on the system by its stoichiometry matrix. Secondly, one assumes inequalities that impose bounds on the system; these inequalities translate the possibility of defininining minimum and maximum allowable reaction fluxes. These two types of constraints (*balances* and *bounds*) define the space of allowable flux distributions of a system, *i.e.*, the rates at which every metabolite/species is consumed or produce by each reaction [142]. In FBA, this allowable space is then further reduced via optimization of the sytem phenotype. This is achieved by *defining* a biological objective relevant to the system in question, and then translating it into a mathematical *objective function* that indicates how much each reaction contributes to the phenotype, and which is to be maximized. Together, the mathematical representations of the constraints and of the phenotype define a system of linear equations, which are solved

in FBA using linear programming. In the case of predicting growth, the objective is biomass production: the rate at which metabolic compounds are converted into biomass constituents. For this, a *biomass reaction* is constructed by the modeler and then artificially introduced in the metabolic reaction system. The biomass reaction drains precursor metabolites from the system at their relative stoichiometries, in order to simulate biomass production [142]. This reaction is then scaled such that its flux be equal to the exponential growth rate (λ) of the organism.

A mathematical presentation of FBA, taken from [142], is given below.

Flux Balance Analysis:
Assume a metabolic reaction network given by $M = (\mathcal{S}, \mathcal{R}, \boldsymbol{\alpha}, \boldsymbol{\beta})$, with stoichiometry matrix $\nabla = \boldsymbol{\beta} - \boldsymbol{\alpha} \in \mathbb{R}^{m \times n}$, where $\mid \mathcal{S} \mid = n, \mid \mathcal{R} \mid = m$. As usually for BRNs, the flux through all of the network's reactions is represented by a vector $\boldsymbol{f} \in \mathbb{R}^m$, while the system state is given by the vector of metabolite concentrations $\boldsymbol{x} \in \mathbb{R}^n$.

FBA assumes the system to be at steady state, which writes as:

$$\frac{d\boldsymbol{x}}{dt} = \nabla^T \boldsymbol{f} = 0 \tag{8.15}$$

Supplementary constraints on the system can be introduced by defining minimum and maximum allowable reaction fluxes:

$$f_i^{\min} \leq f_i \leq f_i^{\max}, \qquad 1 \leq i \leq m \tag{8.16}$$

The FBA objective function can be defined as any linear combination of fluxes:

$$Z = \boldsymbol{c}^T \boldsymbol{f}, \tag{8.17}$$

where $\boldsymbol{c}$ is a vector of weights that indicates how much each reaction contributes to the objective function.

In the case of growth simulation, only the biomass reaction is desired for maximization, *i.e.*, $\boldsymbol{c}$ is a vector of zeros, except for the element corresponding to the biomass reaction, which is set to a value of one.

Then, FBA proceeds by using linear programming techniques to identify a particular flux distribution $\boldsymbol{f}_{FBA}$ that maximizes the objective function of Equation (8.17), while observing the constraints imposed by the flux balance equations of (8.15) and the reaction bound equations of (8.16). Seeing how in the case of growth simulation, the objective function is given by the biomass reaction (whose flux in turn is given by the exponential growth rate), we see that FBA enables prediction (and maximization) of the growth rate of an organism.

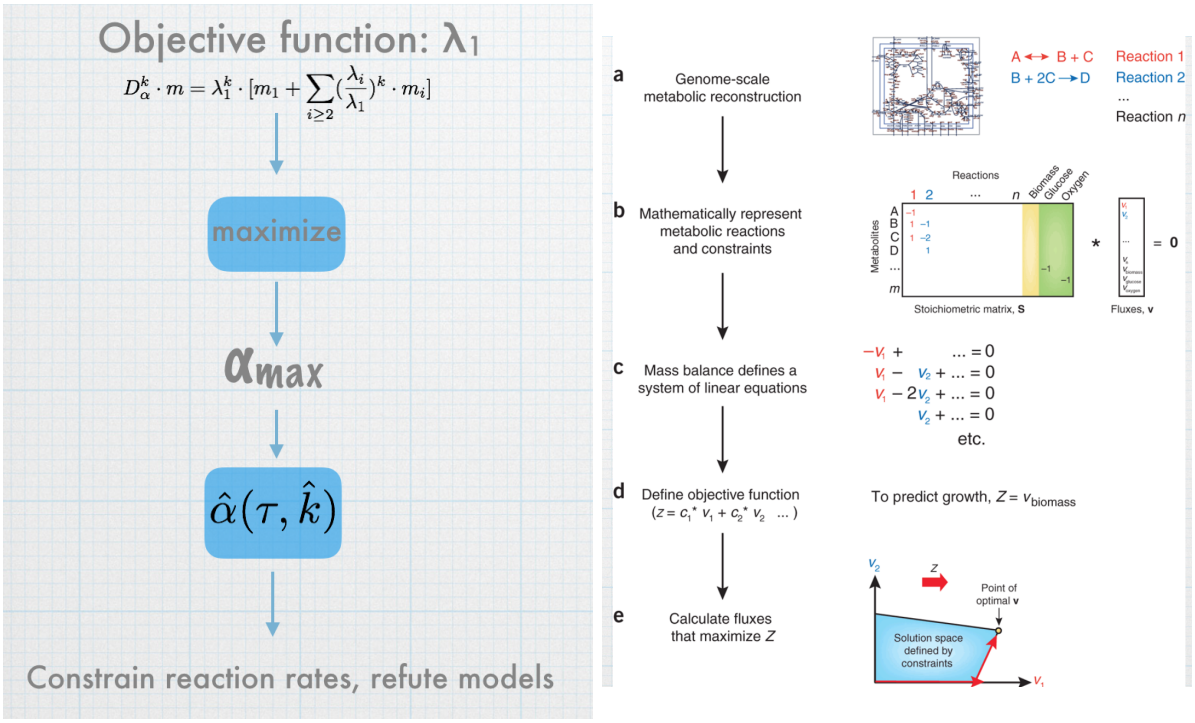


Figure 8.2: **Flux Balance Analysis versus "Synchronous Balanced Analysis"** (*Right: Flux Balance Analysis*) A workflow for FBA, as taken from [142]. When applied to growth maximization, the biomass objective function is defined by adding an artifical "biomass" reaction into the reaction network, the flux of which is set to be equal to the exponential growth rate λ. Thus, growth optimization using FBA is achieved by maximizing the flux of this "biomass reaction". (*Left:"Synchronous Balance Analysis"*) For BRNs comprised exclusively of unimolecular reactions, our Petri net execution semantics enables the mathematical characterization of the system's growth rate, which in turn avoids any artificial addition to the model: maximizing the growth rate becomes akin to finding the matrix D_α with the maximal biggest eigenvalue. Our method can also be used to constrain reaction rates and/or refute models, based on their parameter values.

The FBA method's main advantage lies in the fact that it does not require kinetic parameters. Morevoer, it can be computed quickly even for large reaction networks.

However, its lack of information on kinetic parameters means that FBA is not able to predict metabolite concentrations. Another important limitation consists in its ability of determining fluxes exclusively at *at steady state*.

We argue that our execution semantics can offer an alternative to FBA, that circumvents these issues. Indeed, through our method, the emergent definition of the growth rate λ can be used as an objective function. According to (8.6), maximizing λ is achieved by finding the resource allocation matrix α with the biggest eigenvalue. Once this $\boldsymbol{\alpha}^{\boldsymbol{split}}$ is fixed, Equation (8.13) can be used in order to constrain the reaction rate constants $\hat{\boldsymbol{k}}$, as well as the time-step

τ and the remainder ϵ. By constraining the reaction rate constants, our method could be used as a technique of refuting models.

The advantages, when compared to FBA, are twofold. Firstly, our method can be applied to growth systems: unlike FBA, the system is not considered to be at steady-state. This in turn means that it accounts for the real system kinetcs. Secondly, our approach preserves the ides of biomass maximization, but the artificial creation of an *objective biomass function* is no longer needed. Instead, the growth rate emerges directly from our method. The downside of our approach however lies in maximizing the biggest eigenvalue of matrix D_α.

8.5 Conclusion and future directions

In this chapter, we propose a piecewise synchronous approximation of the dynamics of a (growth) chemical reaction network: a parallel execution semantics of Petri Nets, based on resource allocation. For BRNs composed exclusively of unimolecular reactions, we show that our method can be interpreted either as an approximation of the real system's dynamics, or as a constraint method similar to Flux Balance Analysis. The main advantage of our approach resides in its ability of characterizing the behavior of a cell using only one construction: the resource allocation matrix α_{split}. Consequently, one can eliminate the mechanistic details that deal with resource allocation, and replace them by an abstract vector (α^{split}). Furthermore, when compared to flux balance analysis, our method is applicable to growth systems.

Future work

Our method can be extended into several directions, the most important of which deals with encompassing binary reactions into our execution semantics. Overcoming this obstacle will allow for the construction of untrivial examples based on real-life BRNs, in order to investigate how our method can be used to determine correlations between growth rate and different model parameters (such as reaction rate constants). Once the issue of bi-molecular reactions solved, the two possible interpretations of our approach, as presented in Section 8.4, imply that the quality of our "split-burst" PN semantics could be assessed by comparing it to other existing BRN simulation approaches, such as τ-leaping, or by comparing it to more biologically-inspired, constraint-based heuristics, such as the allocation method proposed by Karr [104] (in which regulation of metabolic protein is carried out via Flux Balance Analysis).

Consider a BRN that contains a bimolecular reaction:

$$\mathsf{r}_1 : A + B \to C$$

and assume a resource allocation (via a matrix $\boldsymbol{\alpha_{split}}$) that reserves 3 molecules of A and 2 molecules of B for consumption by r_1. Naturally, the number of times a reaction can be executed in a max-parallel execution step wil be constrained by the quantity of the least-abundant reactant species, as given by the resource allocation. For example, for the allocation given above, species A represents the "limiting species": r_1 can be executed at most $\min(2, 3) = 2$ times in a max-parallel execution step. Thus, the evolution of the system state is no longer

given by Equation 8.5. A similar equation for describing the evolution of a system containing bimolecular reactions should incorporate the computation of the limiting species' quantities, using the min operator.

The solution to this problem could reside in the use of tropical mathematics, (more specifically *max-plus* or *min-plus* algebras or semi-rings). Therein, matrix multiplication over the ring of real numbers $(\mathbb{R}, +, \times)$ is replaced with matrix multiplication over the idempotent semiring $(\mathbb{R} \cup \{-\infty\} \cup \{\infty\}, \bigoplus, \bigotimes)$, with $\bigoplus$ denoting either the max (for $(\max, +)$) or the min (for $(\min, +)$) operation, and $\bigotimes$ denoting the usual addition. Much of the initial interest in $(\max, +)$, $(\min, +)$ and other idempotent semiring structures was in connection with the modeling of discrete event systems typically arising in areas involving allocation of resources, scheduling and queuing theory [9]. The development of tropical analogues of several ideas from classical linear algebra (*e.g.*, spectral theory, linear independence) resulted in a wealth of results w.r.t. the dynamics of discrete event systems, including descriptions of periodic regimes, and asymptotic behaviors. The question arising in our case refers to how these results can be extended to the case of models of BRNs.

Indeed, (timed) Petri nets represent a common tool to describe discrete event systems (DES). It is known that conventional linear systems have "modes" related to their eigenstructures, which are reached asymptotically when said systems are stable. The $(\max, +)$ and $(\min, +)$ approaches enable the definition of similar notions for a subclass of Petri nets: in a nutshell, they enable a structural description of the net's *throughput* λ, which is shown to be equal to the *largest average weight* of a directed circuit of the net. The throughput λ is associated with the *cycle time* of the underlying system: if such a λ exists, then each transition of the network becomes active every λ units of time. Consequently, the notion of throughput can be used as a proxy of the net's growth rate.

The main issue with the tropical approach is related to the restrictions imposed on the net structure: the class of timed Petri nets which can be described by linear max-plus or min-plus equations is given by *Timed Event Graphs* (TEGs). TEGs enable modeling of *synchronization*, but not of *conflicts*: in a TEG, each place has exactly one upstream transition, and exactly one downstream transition. Whereas the latter restriction can be solved by splitting resources according to the resource allocation matrix $\boldsymbol{\alpha}^{\boldsymbol{split}}$, the "exactly one upstream transition" condition appears to be too restrictive for most models of BRNs (as it translates into "each species can be produced by exactly one reaction"). Relaxing this restriction leads to weaker results [42] w.r.t. the existence of the throughput (*i.e.* growth rate in our case). What's more, the results w.r.t. the throughput of a TEG assume the net to be *autonomous* and *strongly connected*.

Another issue is related to the *timing* aspect: in the case of Petri nets that model discrete event systems, a constant "duration" is associated to each place, in order to denote the minimum sojourn time of a token in a place. In order to apply the $(\min, +)$-algebra analysis to models of BRNs, the relation between chemical reaction rate constants and such sojourn times needs to formally established.

Despite these restrictions, the tropical approach has proven to be useful in the analysis of certain biological systems. For example, in [21], a $(\max, +)$ model for the dynamics of mRNA translation is proposed, which allows one to predict protein production rates and codon occupation densities. The results of [21] give an indication of how, under a suitable level of

abstraction, the results obtained via $(\max, +)$-algebra analysis can be harnessed for biological purposes. This *suitable level of abstraction* seems to consist in modeling a BRN using a Petri net in which each transition models a biological *event* in an abstract way, instead of the usual direct transition-to-reaction correspondence between PNs and BRNs. For example, in [21], the authors perform the $(\max, +)$ analysis on a TEG in which the firing of a transition models the event consisting in "a ribosome moving from codon i to codon *i+1*", without describing the mechanistic details behind this event. Consequently, future work includes studying the trade-off between the level of abstraction in DES models of BRNs and results obtained via tropical mathematics analysis.

Another way to potentially relieve the issue of biomolecular reactions is by assuming that no significant change in the concentration of one of the two reactants is being caused by any other reaction during one execution step. In this sense, Michaelis-Menten like reduction schemes could be considered.

Conclusion

Chapter 9
Conclusion and future directions

9.1 Contributions

In this thesis, we propose several contributions related to the modeling and analysis of Biochemical Reaction Networks. In the first part, we study how the inherent multi-scaleness of biological systems can be exploited for both model reduction heuristics, and model reduction error estimation techniques. The first work deals with the use of rule-based modeling for prototyping genetic regulatory networks: we argue that because of their capacity to unambiguously specify protein-protein interactions in general, and consequently mechanistic details such as DNA binding sites, dimerisation of transcription factors, or co-operative interactions, rule-based modeling languages are more appropriate for designing genetic circuits in a modular, transparent and easily modifiable fashion than reaction based languages.

Nonetheless, such a detailed description comes with complexity, as well as computationally costly executions. Consequently, we next propose a general reduction method for a subset of rule-based models, aimed at modeling genetic circuits, which exploits concentration and time-scale separation. The method is an adaptation of an existing algorithm [112], designed for reaction based models. Our version of the algorithm proceeds by static inspection of the rule-set: it scans the Kappa model in search for interaction patterns known to be amenable to equilibrium approximations (e.g. Michaelis-Menten scheme, as well as a number of other stoichiometry-simplifying techniques), and after performing additional checks in order to verify if the reduction is meaningful in the context of the full model, it proceeds with the elimination of intermediate species and with the adjustment of the rule rates. When tested on a detailed rule-based model of a λ-phage switch, our tool provides a dramatic reduction in simulation time of several orders of magnitude. Our method is an illustration of the fact that in general, the multiscaleness of biochemical reaction networks represents a feature that can be exploited for model reduction purposes.

The correctness of our approach relies on the fact that the approximate model is equal to the original one, in the artificial limit where certain reactions happen at a sufficiently larger time-scale than others, and they are seemingly equilibrated shortly upon the reactions' initiation. However, not unlike other scale-separation reduction methods, our method relies on a solid physical justification, yet (to the best of our knowledge) there is no precise method to quantify the error induced by time-scale separation approximations for biochemical reaction networks,

while avoiding to solve the original model.

This is why, we next propose an approximation method in the deterministic modeling framework that exploits the multiscaleness property, in which reduction guarantees represent the major requirement. Our method combines abstraction and numerical approximation, and aims at providing a better evaluation of model reduction methods that are based on time- and concentration- scale separation. The reduction guarantees of our method are a consequence of a carefully designed symbolic propagation of dominance constraints: given an ODE model of a BRN that exhibits time- and concentration- scale separation, we abstract the solution of the original system by a "box" that over-approximates the state of the original system and provides lower and upper bounds for the value of each variable of the system in its current state. The simpler equations (which we call *tropicalized*) that define the hyperfaces of the box are obtained by combining the dominance concept borrowed from tropical analysis [119] with symbolic bounds propagation. Mass invariants of the initial system of differential equations are used to refine the computed bounds, thus improving the accuracy of the method. The resulting (simplified) system provides a posteriori time- dependent lower and upper bounds for the concentrations of the initial model's species, and thus bounds on numerical errors stemming from tropicalization. This means that no information on the original system's trajectory is needed - the most important advantage of our approach.

In the second part of this manuscript, we address the formal modelling of relevant biological behaviours, such as intracellular resource storage and cellular growth, in the deterministic modeling framework of biochemical reaction networks. We first introduce a new reparametrisation technique of ordinary differential equations models, intended to model intracellular resource storage strategies. Our technique consists in defining a generic scaling transformation of biochemical reaction networks that allows one to tune the concentration of certain chemical species, while preserving the network's behaviour at steady state. We then employ this technique to study the effect of different storage strategies on cellular growth. This fundamental trade-off is best investigated using a mechanistic model of cellular growth, where costs are emergent and reflect architectural traits of the growth machinery. Consequently, we apply our method on such a recent "self-replicator" mathematical model [182] of the coarse-grained mechanisms that drive cellular growth. Our analysis is carried out first for a single-cell model, and afterwards in a competitive context. In the single-cell experiments, we compare storage strategies against different patterns of environmental changes. Our results suggest that on the one hand there is a cost associated with high levels of storage, which results from the loss of stored resources through dilution, and that on the other hand, high levels of storage can benefit cells in variable environments, by increasing biomass production during transitions from one medium to another. In the competitive experiments, we test populations of low- and high-storage strategies against each other, in a variety of environments parametrized by the frequency of two superimposed probabilistic trains of high and low pulses of sugar. All in all, we are able to observe a rich interplay of storage levels, growth rates, growth yield, and resource variability. Our results indicate that the specificities of environmental changes play a decisive role in deciding which storage strategy is deemed the most beneficial (with respect to accumulating biomass over time). This is even more so the case when species content for the same resource: the combined effect of storage strategies and competition lead to extracting less biomass out of the same amount of

resources.

In the final work of this manuscript, we work towards a characterization of cellular growth as an emergent property of model execution semantics. In this sense, we work towards a novel Petri net execution semantics representing a piecewise-synchronous approximation of the deterministic dynamics. To achieve this, we propose to model a biochemical reaction network using a resource-allocation-centered Petri net, with parallel maximal-step execution semantics. We argue that this semantics is better-suited for modeling biochemical reaction networks, when compared to the classical interleaving semantics, as it takes into account the inherently concurrent nature of biological processes. In the case of unimolecular chemical reactions, we prove the correctness of our method and show that it can be used either as an approximation of the dynamics, as a technique of refuting models, or as a method of constraining the reaction rate constants - consequently, as an alternative to flux balance analysis, using an emergent formally defined notion of "growth rate" as the objective function.

9.2 Future works

The works presented in this manuscript can be extended in several directions. The detailed description of each considered extension is contained in the chapters corresponding to each project. We give a final brief summary of the planned extensions below.

- **"Prototyping genetic circuits using rule-based models"**:
  - the reduction algorithm is currently being extended in order to ensure a sound reduction in the case of "don't care don't write" Kappa patterns;
  - we also plan on studying how the set of approximation patterns can be extended in order to obtain good reductions for complex models of signaling pathways.
- **"Tropical Abstraction of BRNs with guarantees"**: we plan on extending our method as to take into account reaction networks with no mass conservation laws. In this sense, we plan on looking into tropical equilibrations and the permanency concept [140].
- **"Abstractions for cellular growth"**: we plan on analyzing how our piecewise-synchronous Petri net execution semantics can be extended to bimolecular reaction networks, either by using the $(\max, +)$-algebra approach, or by using Michaelis-Menten like reduction schemes. We also plan on comparing our method to the τ-leaping simulation method, and to the allocation method propsed by Karr [104].

Appendices

Chapter A
Probability Theory

Herein, we review the basic definitions and concepts of elementary probability theory, which are required for defining the *stochastic* model semantics of biochemical reaction networks.

The usual notations from set theory are used. For subsets $A, A_k, B, \ldots$ of some abstract space Ω, the set *union* writes as $A \cup B$ or $\bigcup_k A_k$, *intersection* writes as $A \cap B$ or $\bigcap_k A_k$, *complement* writes as A^C, and *difference* writes as $A \setminus B = A \cup B^C$. We denote by $\mathcal{P}(\Omega) \equiv \{A \mid A \in \Omega\}$ the set of all subsets of Ω (*i.e.*, the *power-set*), by $|\Omega|$ the number of elements in Ω, and by $\mathcal{R}(f)$ the range of values taken by a function $f : A \mapsto B$.

Basic notions of measure theory

We start by presenting the basic notions and results of measure theory, for which a simple definition is that it is a theory about the distribution of mass over a set $\mathbb{S}$. If the mass is uniformly distributed and $\mathbb{S}$ is an Euclidean space $\mathbb{R}^k$, it is the theory of Lebesgue measure on $\mathbb{R}^k$ (i.e., length in $\mathbb{R}$, area in $\mathbb{R}^2$, volume in $\mathbb{R}^3$, etc.). Probability theory is concerned with the case when $\mathbb{S}$ is the sample space of a random experiment and the total mass equals one. Thus, measure theory notions will be useful in defining concepts from probability theory.

Definition A.1 (σ**-algebra**)
Let Ω be a set. A collection of sets $\mathcal{F} \in \mathcal{P}(\Omega)$ is called a ***σ-algebra*** *on Ω if it is nonempty and closed under countable set operations:*

(i) $\emptyset \in \mathcal{F}$,

(ii) $A \in \mathcal{F} \Rightarrow A^C \in \mathcal{F}$,

(iii) $A_n \in \mathcal{F}$ *for* $n \geq 1 \Rightarrow \bigcap_{n \geq 1} A_n \in \mathcal{F}, \bigcup_{n \geq 1} A_n \in \mathcal{F}$.

Example A.1
Let $\mathcal{A}$ be any class of subsets of a set Ω. The set of all subsets of Ω is a σ-algebra containing $\mathcal{A}$. The intersection of any collection of σ-algebras is again a σ-algebra. The collection of σ-algebras containing $\mathcal{A}$ is therefore non-empty and its intersection is a σ-algebra $\sigma\langle\mathcal{A}\rangle$, which is called ***the σ-algebra generated by*** *$\mathcal{A}$:*

$$\sigma\langle\mathcal{A}\rangle = \bigcap_{\mathcal{F}\in\mathcal{I}(\mathcal{A})} \mathcal{F},$$

where $\mathcal{I}(\mathcal{A}) \equiv \{\mathcal{F} \mid \mathcal{A} \subset \mathcal{F}$ *and* $\mathcal{F}$ *is a* σ*-algebra on* $\Omega\}$ *is the collection of* σ*-algebras containing* $\mathcal{A}$*.*

A particularly useful class of σ-algebras are those generated by open sets of a topological space. These are called **Borel σ-algebras**. We recall that a *topological space* is a pair $(\mathbb{S}, \mathcal{T})$, where $\mathbb{S}$ is a nonempty set and $\mathcal{T}$ is a collection of subsets of $\mathbb{S}$ such that *(i)*$\mathbb{S} \in \mathcal{T}$, *(ii)*$\mathcal{O}_1, \mathcal{O}_2 \in \mathcal{T} \Rightarrow \mathcal{O}_1 \cap \mathcal{O}_2 \in \mathcal{T}$, and *(iii)*$\{\mathcal{O}_\alpha \mid \alpha \in I\} \subset \mathcal{T} \Rightarrow \bigcup_{\alpha\in I} \mathcal{O}_\alpha \in \mathcal{T}$. Elements of $\mathcal{T}$ are called **open sets**.

Similarly, a *metric space* is a pair $(\mathbb{S}, d)$, where $\mathbb{S}$ is a nonempty set and d is a function $d : \mathbb{S} \times \mathbb{S} \mapsto \mathbb{R}^+$ satisfying *(i)* $d(x, y) = d(y, x), \forall x, y \in \mathbb{S}$, *(ii)* $d(x, y) = 0$ iff $x = y$, and *(iii)* $d(x, z) \leq d(x, y) + d(y, z), \forall x, y, z \in \mathbb{S}$. The function d is called a *metric on* $\mathbb{S}$.

Definition A.2 (**Borel σ-algebra**)
The ***Borel σ-algebra*** *on a topological space* $\mathbb{S}$ *(in particular, on a metric space or an Euclidean space) is defined as the* σ*-algebra generated by the collection of open sets in* $\mathbb{S}$*.*

A *set function* is an extended real valued function defined on a class of subsets of a set Ω. *Measures* are nonnegative set functions that, intuitively speaking, measure the content of a subset of Ω. Consequently, they have to satisfy certain natural requirements, such as "the measure of the union of a countable collection of disjoint sets is the sum of the measures of the individual sets":

Definition A.3 (**Measure space**)
Let Ω *be a nonempty set and* $\mathcal{F}$ *a* σ*-algebra on* Ω*. The pair* $(\Omega, \mathcal{F})$ *is called a* ***measurable space****, and each* $A \in \mathcal{F}$ *is called a* ***measurable set****. A* ***measure*** μ *on* $(\Omega, \mathcal{F})$ *is a function* $\mu : \mathcal{F} \mapsto [0, \infty)$ *which satisfies the countable additivity property:*

(i) $\mu(\emptyset) = 0$*,*

(ii) $A_1, A_2, \ldots \in \mathcal{F}$ *a sequence of disjoint sets* $\Rightarrow \mu(\bigcup_{k\geq 1} A_k) = \sum_{k\geq 1} \mu(A_k)$*.*

The triple $(\Omega, \mathcal{F}, \mu)$ *is called a* ***measure space****.*

A measure μ is said to be **finite** if $\mu(\Omega) < \infty$, respectively **infinite** if $\mu(\Omega) = \infty$. A finite measure with $\mu(\Omega) = 1$ is called a **probability measure**. A measure μ on a σ-algebra $\mathcal{F}$ is called **σ-finite** if there exist a countable collection of sets $A_1, A_2, \ldots \in \mathcal{F}$ such that *(a)* $\bigcup_{n\geq 1} A_n = \Omega$ and *(b)* $\mu(A_n) < \infty, \forall n \geq 1$.

Armed with the definition of a *measure space*, one can proceed to define a basic probability space, a notion that provides the basic apparatus for modelling randomness:

Definition A.4 (**Probability space**)
A ***probability space*** *is a measure space* $(\Omega, \mathcal{F}, P)$*, with* P *a probability measure (i.e.,* $P(\Omega) =$

1*). A probability space provides a model for an experiment whose outcome is subject to chance. In the probabilistic context, the interpretation of the underlying measure space components are as follows:*

(i) Ω *is the set of possible outcomes of the experiment, called* ***samples****;*

(ii) $\mathcal{F}$ *is a set of* ***observable sets of outcomes****, and sets* $A \in \mathcal{F}$ *are called* ***events****;*

(iii) and $P(A)$ *is called the* ***probability*** *of event* $A \in \mathcal{F}$*.*

Other useful examples of measures are:

Example A.2 (**The counting measure**)
When Ω *is a finite countable set, the measure on* $(\Omega, \mathcal{P}(\Omega))$ *that assigns to each measurable set* A *the number of elements it contains, i.e.* $m(A) = |A|$*, is called the counting measure on* Ω*. If* Ω *is not countable, one works instead with Borel sets.*

Example A.3 (**The Lebesgue measure**)
Given a set Ω*, the Borel* σ*-algebra of* Ω *(i.e., the* σ*-algebra generated by the set of open sets in* Ω*) writes as* $\mathcal{B}(\Omega)$*. The elements of* $\mathcal{B}(\Omega)$ *are called* ***Borel sets****. If* $\Omega = \mathbb{R}$*, and* $\mathcal{A} = \{(a, b) \mid a, b \in \mathbb{R}, a < b\}$*, then* $\mathcal{B}$ *denotes the* ***Borel*** σ***-algebra on*** $\mathbb{R}$*. It can be shown that there is a unique measure* μ *on* $(\mathbb{R}, \mathcal{B})$ *such that:*

$$\mu(a, b) = a - b, \qquad \forall a, b.$$

This measure μ *is called the* ***Lebesgue measure****.*

Example A.4 (**Discrete probability measures**)
Let $\omega_1, \omega_2, \ldots \in \Omega$*, and* $p_1, p_2, \ldots \in [0, 1]$*, such that* $\sum\limits_{i \geq 1} p_i = 1$*. Define for any* $A \in \Omega$*:*

$$\mathrm{P}(A) = \sum_{i \geq 1} p_i 1_A(\omega_i),$$

with 1_A *the indicator function of a set* A*, as defined in Example A.5. For any disjoint collection of sets* $A_1, A_2, \ldots \in \mathcal{P}(\Omega)$*,*

$$\begin{aligned}
\mathrm{P}(\bigcup_{i \geq 1} A_i) &= \sum_{j \geq 1} p_j I_{\bigcup\limits_{i \geq 1} A_i}(\omega_j) \\
&= \sum_{j \geq 1} p_j \Big(\sum_{i \geq 1} I_{A_i}(\omega_j) \Big) \\
&= \sum_{i \geq 1} \Big(\sum_{j \geq 1} p_j I_{A_i}(\omega_j) \Big) \\
&= \sum_{i \geq 1} \mathrm{P}(A_i),
\end{aligned}$$

which shows that P *is a* ***probability measure*** *on* $\mathcal{P}(\Omega)$*.*

Oftentimes, when dealing with probabilities, one is not interested in the full details of a measure space $(\Omega, \mathcal{F}, \mu)$, but only on certain functions defined on Ω: if Ω represents the outcome of 10 tosses of a fair coin, one may only be interested in knowing the number of heads in the 10 tosses. As it turns out, in order to assign measures (*probabilities*) to sets (*events*) involving such functions, one can only allow certain functions, called *measurable functions*:

Definition A.5 (**$\langle \mathcal{F}, \mathcal{B} \rangle$-measurable function**)
*Let $(\Omega, \mathcal{F})$ be a measurable space. A function $f : \Omega \mapsto \mathbb{R}$ is said to be $\langle \mathcal{F}, \mathcal{B} \rangle$-**measurable**, or simply $\mathcal{F}$-**measurable**, if for each $a \in \mathbb{R}$:*

$$f^{-1}((-\infty, a]) \equiv \{\omega \mid f(\omega) \leq a\} \in \mathcal{F}$$

This definition can be generalized to maps between any two measurable spaces:

Definition A.6 (**Measurable function**)
*Let $(\Omega_1, \mathcal{F}_1)$ and $(\Omega_2, \mathcal{F}_2)$ be two measurable spaces. A function $f : \Omega_1 \mapsto \Omega_2$ is $\langle \mathcal{F}_\infty, \mathcal{F}_\in \rangle$-**measurable** if the inverse image of any $A \in \mathcal{F}_2$ lies in $\mathcal{F}_1$:*

$$\forall A \in \mathcal{F}_2, f^{-1}(A) \in \mathcal{F}_1, \text{ with } f^{-1}(A) := \{a \in \Omega_1 \mid f(a) \in A\}.$$

A measurable function is often called a **measurement** on $(\Omega_1, \mathcal{A}_1)$. A measurable function on a Borel σ-algebra is called a Borel function.

Example A.5
A measurement on $(\Omega, \mathcal{A})$ is the indicator function of the set $A \in \mathcal{A}$, given by :

$$1_A(a) = \begin{cases} 1, & \text{if } a \in A \\ 0, & \text{otherwise} \end{cases} .$$

Then, a random variable on a probability space $(\Omega, \mathcal{F}, \mathrm{P})$ may be defined as an $\langle \mathcal{F}, \mathcal{B} \rangle$-measurable function on Ω, and is interpreted as reducing the probability space to the subset of observations of interest:

Definition A.7 (**Random variable**)
*Let $(\Omega, \mathcal{F}, \mathrm{P})$ be a probability space. Then a function $X : \Omega \mapsto \mathbb{R}$ is called a **random variable**, if the event*[1]

$$X^{-1}((-\infty, a]) \equiv \{\omega \mid X(\omega) \leq a\} \in \mathcal{F}$$

for each $a \in \mathbb{R}$, i.e., a random variable is a real valued $\langle \mathcal{F}, \mathcal{B} \rangle$-measurable function on a probability space $(\Omega, \mathcal{F}, \mathrm{P})$.

A random variable X can be either **discrete**, if $\mathcal{R}(X)$ is a countable set, or **continuous**, otherwise.

Definition A.8
A random variable X is called:

[1] or equivalently if $X^{-1}(A) \in \mathcal{F}, \forall A \in \mathcal{B}(\mathbb{R})$

- ***discrete**, if there exists a countable set $A \in \mathbb{R}$ such that $\mathrm{P}(X \in A) = 1$,*
- ***continuous**, if $\mathrm{P}(X = x) = 0, \forall x \in \mathbb{R}$.*

If X is a random variable defined on a probability space $(\Omega, \mathcal{F}, \mathrm{P})$, then P governs the probabilities assigned to events such as $X^{-1}([a, b]), -\infty < a < b < \infty$. Since X takes values in the real line, one would like to express such probabilities only as a function of the set $[a, b]$. Since X is $\langle \mathcal{F}, \mathcal{B} \rangle$-measurable, $X^{-1}(A) \in \mathcal{F}, \forall A \in \mathcal{B}(\mathbb{R})$, and the function:

$$\mathrm{P}_X(A) \equiv \mathrm{P}(X^{-1}(A))$$

is a set function defined on $\mathcal{B}(\mathbb{R})$.

Proposition A.1
Let $(\Omega_1, \mathcal{F}_1)$ and $(\Omega_2, \mathcal{F}_2)$ be two measurable spaces, and let $T : \Omega_1 \mapsto \Omega_2$ be a $\langle \mathcal{F}_\infty, \mathcal{F}_2 \rangle$-measurable function. Then, for any measure μ on $(\Omega_1, \mathcal{F}_1)$, the set function μT^{-1}, defined by

$$\mu T^{-1}(A) \equiv \mu(T^{-1}(A)), A \in \mathcal{F}2$$

is a measure on $\mathcal{F}_2$.

Definition A.9
The measure μT^{-1} is called the measure induced by T (or the induced measure of T) on $\mathcal{F}_2$.

In particular, if $\mu(\Omega_1) = 1$, then $\mu T^{-1}(\Omega_2) = 1$.

The, P_X turns out to be a probability measure on $\mathcal{B}(\mathbb{R})$, called *the probability distribution of X*:

Definition A.10 (**Probability distribution**)
*For a random variable X defined on a probability space $(\Omega, \mathcal{F}, \mathrm{P})$, the **probability distribution of** X (or **law of** X) is given by the induced measure of X under P on $\mathbb{R}$, i.e., $\mathrm{P}_X \equiv \mathrm{P}X^{-1}$.*

We note that when defining probabilities of events like '$X \in [a, b]$', it is common practice to use the **probability mass function** when dealing with *discrete random variables* and the **probability density function** for *continuous random variables*. The measure-theoretic definition above allows one to treat both these cases, as well as the case of "mixed" distributions, under a unified framework.

Definition A.11 (**Cumulative distribution function**)
*The cumulative distribution function (or **cdf** in short) of a random variable X is defined as:*

$$\begin{aligned} F_X(x) &\equiv \mathrm{P}_X((-\infty, x]) \\ &\equiv \mathrm{P}(\{\omega \mid X(\omega) \leq x\}), x \in \mathbb{R}. \end{aligned}$$

Integral of a measurable function

Let $(\Omega, \mathcal{F}, \mu)$ be a measure space and $f : \Omega \mapsto \mathbb{R}$ be an arbitrary measurable function. One wants to define the *integral* of f with respect to the measure μ, which will be used to define the *expectancy* of a random variable.

If $(\Omega, \mathcal{F})$ is a measurable space, denote by $m\mathcal{F}$ the set of measurable functions $f : \Omega \mapsto \mathbb{R}$. Then, $m\mathcal{F}$ is a vector space. Let $m\mathcal{F}^+$ denote the set of non-negative measurable functions $f : \Omega \mapsto [0, \infty]$, where one takes on $[0, \infty]$ the σ-algebra generated by the open intervals (a, b). Then, $m\mathcal{F}^+$ is a cone:

$$(f, g \in m\mathcal{F}^+, \alpha, \beta \geq 0) \Rightarrow \alpha f + \beta g \in m\mathcal{F}^+,$$

which is closed under countable suprema:

$$(f_i \in m\mathcal{F}^+, i \in I) \Rightarrow \sup_i f_i \in m\mathcal{F}^+.$$

It follows that, for a sequence of functions $f_n \in m\mathcal{F}^+$, both $\limsup_n f_n$ and $\liminf_n f_n$ lie in $m\mathcal{F}^+$ (so does $\lim_n f_n$, when it exists). It can be shown that there is a unique map $\tilde{\mu} : m\mathcal{F}^+ \mapsto [0, \infty]$, such that:

(i) $\tilde{\mu}(1_A) = \mu(A), \forall A \in \mathcal{F}$;

(ii) $\tilde{\mu}(\alpha f + \beta g) = \alpha\tilde{\mu}(f) + \beta\tilde{\mu}(g), \forall f, g \in m\mathcal{F}^+, \alpha, \beta \geq 0$;

(iii) $(f_n \in m\mathcal{F}^+, n \in \mathbb{N}) \Rightarrow \tilde{\mu}(\sum_n f_n) = \sum_n \tilde{\mu}(f_n)$.

For $f \in m\mathcal{F}$, let $f^+ = \max f, 0$ and $f^- = \max -f, 0$. Then, $f^+, f^- \in m\mathcal{F}^+$, $f = f^+ - f^-$, and $|f| = f^+ + f^-$. If $\tilde{\mu}(|f|) < \infty$, then f is to be *integrable*, and one can set:

$$\tilde{\mu}(f) = \mu(f^+) - \mu(f^-).$$

We call $\tilde{\mu}(f)$ **the integral** of f, but by convention, this notation can be replaced by one of the alternatives:

$$\mu(f) = \int_\Omega f d\mu = \int_{x\in\Omega} f(x)\mu(dx).$$

In the case of the Lebesgue measure μ, one simply writes $\int_{x\in\mathbb{R}} f(x)dx$.

Definition A.12 (**Expected value**)
*Let X be a random variable on a probability space $(\Omega, \mathcal{F}, \mathrm{P})$. The **expected value** (or **mean), or expectation** of X, denoted by $\mathrm{E}[X]$, is defined as:*

$$\mathrm{E}[X] \equiv \int_\Omega X d\mathrm{P},$$

Definition A.13 (**Variance**)
*The **variance** of a random variable X is defined as:*

$$\mathrm{Var}(X) \equiv \mathrm{E}[(X - \mathrm{E}[X])^2].$$

Example A.6 (**Bernoulli random variable**)
The simplest example of a discrete random variable is the ***Bernoulli random variable****,* $\mathrm{X_p} : \Omega \mapsto \{0, 1\}$*, which models an experiment with only two possible outcomes (denoted by 0 or 1), which occur with probabilities* $\mathrm{P_{X_p}}(1) = p$*, and* $\mathrm{P_{X_p}}(0) = 1 - p$*. The mean of a Bernoulli variable is* $\mathrm{E}[\mathrm{X_p}] = p$*, and its variance is* $\mathrm{Var}(\mathrm{X + p}) = p(1 - p)$*.*

Example A.7 (**Exponential, Poisson random variables**)
The two most important random variables for constructing a continuous Markov chain are:

(i) the ***exponential random variable****,* $\mathrm{Exp}_\lambda : \Omega \mapsto [0, \infty)$*, with probability density function* $f_{\mathrm{Exp}_\lambda}(x) = \lambda e^{-\lambda x}$*, mean* $\mathrm{E}[\mathrm{Exp}_\lambda] = \frac{1}{\lambda}$*, and variance* $\mathrm{Var}(\mathrm{Exp}_\lambda) = \frac{1}{\lambda^2}$

(ii) the ***Poisson random variable****,* $\mathrm{Poisson}_\lambda : \Omega \mapsto \{0, 1, 2, \ldots\}$*, with probability mass function* $\mathrm{P}(\mathrm{Poisson}_\lambda = \mathrm{k}) = \frac{\lambda^k e^{-k}}{k!}$*, mean* $\mathrm{E}[\mathrm{Poisson}_\lambda] = \lambda$*, and variance* $\mathrm{Var}(\mathrm{Poisson}_\lambda) = \lambda$*.*

Definition A.14 (**Memoryless property**)
For $x, y \in \mathcal{R}(\mathrm{X})$*,* $X : \Omega \mapsto \mathbb{R}$ *is said to have the* ***memoryless property****, if*

$$\mathrm{P}\{\mathrm{X} > x + y \mid \mathrm{X} > x\} = \mathrm{P}\{\mathrm{X} > x\}$$

.

The only continuous random variables to satisfy the memoryless property are exponential random variables.

Markov chains

Definition A.15 (**Stochastic process**)
A ***stochastic process****, or* ***random process****, with state space* S *and parameter set* T *is a collection of random variables* $\{X_t, t \in T\}$*, defined on a common probability space* $(\Omega, \mathcal{A}, \mathrm{P})$*. The process is said to be "discrete" if* T *is countable, and "continuous", otherwise.*

Usually, t is to be interpreted as an indicator of *time*, in which case X_t represent the state of the process at time t. For every fixed $\omega \in \Omega$, the mapping $t \mapsto X_t(\omega)$ defines a *trace/trajectory/realization/sample path* of the process.

In order to facilitate analysis, additional properties are assumed on a stochastic model:

Definition A.16 (**Markov property, time-homogeneity**)
For a given stochastic process $\{X_t\}$ *on a countable state space* S*, let* $\mathcal{H}_{X(t)}$ *denote all the information about the process until time* $t \in T$*. Then, the process* $\{X_t\}$ *is said to* ***satisfy the Markov property****, if the conditional (on both past and present states) probability distribution of future states of the process depends only upon the present state, not on the sequence of events that preceded it. That is, if for all states* $s, s' \in S$*, and for all times* $t + h > t$*, one has that:*

$$\mathrm{P}\{X_{t+h} = s' \mid \mathcal{H}_{X_t}\} = \mathrm{P}\{X_{t+h} = s' \mid X_t\}.$$

*The process is said to be **time-homogeneous** if the transition probability between any two state values at two given times depends only on the difference between those state values:*

$$\mathrm{P}\{X_{t+h} = s' \mid X_t = s\} = \mathrm{P}\{X_h = s' \mid X_0 = s\}$$

.

Definition A.17 (**Transition probability matrix**)
*For a stochastic process with a finite state space S, for some ordering of the elements of S, the **transition probability matrix** for step t is defined as $\boldsymbol{P}^{(t)}$, a $S \times S$-matrix with entries $\boldsymbol{P}^{(t)}(s, s') = \mathrm{P}\{X_t = s' \mid X_0 = s\}$.*

Then, any Markov time-homogeneous process satisfies the Chapman-Kolmogorov equations:

$$\boldsymbol{P}^{(t+h)} = \boldsymbol{P}^{(t)}\boldsymbol{P}^{(h)}, \forall t, h \in T$$

.

Then, a discrete (respectively continuous) time *Markov chain* is defined as a discrete (respectively continuous) time random process $\{X_n\}_{n\in\mathbb{N}}$ (respectively $\{X_t\}_{t\in\mathbb{R}_{\geq 0}}$ that satisfies the Markov and time-homogeneity properties.

Another definition of Markov chains consists in assigning a Markov process to a Markov graph:

Definition A.18 (**Markov graph**)
*A **Markov graph** is a triple (S, w, p_0), with:*

(i) *S a countable state space,*

(ii) *$w : S \times S \mapsto \mathbb{R}$, the transition weight function*

(iii) *$p_0 : S \mapsto [0, 1]$ such that $\sum_{s_i} p_0(s_i) = 1$.*

Definition A.19 (**Discrete-time Markov Chain (DTMC)**)
*A discrete-time Markov graph $M = (S, w, p_0)$ is such that $\forall s \in S, w(s, :)$ is a probability. Then, a process $\{X_t\}$ assigned to M is called a **discrete-time Markov chain (DTMC)**, and is such that $\forall s, s' \in S$, the following conditions are satisfied:*

(i) $\mathrm{P}(X_0 = s) = p_0(s)$,

(ii) $\mathrm{P}(X_1 = s' \mid X_0 = s) = w(s, s')$.

Reasoning about continuous-time Markov chains proves to be a bit more subtle: the parameter set T is uncountable, and one cannot assign a probability to an uncountable union of marginal distributions. Instead, the probability of transitions will be defined in a small interval $[0, h)$. In this case, one talks about a restriction to *right-continuous processes*: $\forall w \in \Omega, \forall t \geq 0, \exists \epsilon > 0$ such that $X_s(w) = X_t(w)$, for $s \in [t, t + \epsilon]$.

Definition A.20 (**Continuous-time Markov Chain (CTMC)**)
A continuous-time Markov graph $M = (S, w, p_0)$ *is such that* $\forall s \in S, s \neq s' \Rightarrow w(s, s') \geq 0$, *where* w *is also called the rate function. For any state* $s \in S$, *the activity of state* s *can be defined as* $a(s) = \sum_{s' \in S} w(s, s')$. *Also, for each state* $s \in S$, *the rate function is set to* $w(s, s) = a(s)$. *A process* $\{X_t\}$ *assigned to* M *is called a* ***continuous-time Markov chain (CTMC)****, and is such that* $\forall s, s' \in S$, *and* $\forall t \geq 0$ *the following conditions are satisfied:*

(i) $\mathrm{P}(X_0 = s) = p_0(s)$,

(ii) $\mathrm{P}(X_{t+h} = s' \mid X_t = s) = \begin{cases} w(s, s')h + o(h), & \text{if } s \neq s' \\ 1 - a(s)h + o(h), & \text{otherwise} \end{cases}$, *with* $f \in o(h)$ *if* $\lim_{h \to 0} \frac{f(h)}{h} = 0$.

Another definition of a CTMC, which can prove more useful for its simulation, can be given by associating each right-continuous process with the random variables:

(i) $\xi_0, \xi_1, \ldots \in \mathbb{R}_{\geq 0}$, the *jump times* of $\{X_t\}$, defined as the (absolute) time instances at which jumps occur [2]:

$$\begin{cases} \xi_0 = 0 \\ \xi_{n+1} = \inf\{t > \xi_n \mid X_t \neq X_{\xi_n}\} \end{cases}$$

(ii) $\tau_0, \tau_1, \ldots \in \mathbb{R}_{\geq 0}$, the waiting times relative to the last jump: $\tau_i = \xi_{i+1} - \xi_i$

(iii) $Z_0, Z_1, \ldots \in S$ the sequence of states visited by jumps ($Z_i = X_{\xi_i}$) which defines the *embedded discrete process.*

Then, a CTMC $\{X_t\}$ is such that:

(i) $\mathrm{P}(X_0 = s) = p_0(s), \forall s \in S$,

(ii) $\mathrm{P}(\xi_i < t \mid Z_0 = s) = 1 - e^{-a(s)t}, \forall s, s' \in S$,

(iii) $\mathrm{P}(Z_1 = s' | Z_0 = s) = \frac{w(s,s')}{a(s)}, \forall s, s' \in S$.

[2] We are working under the assumption of non-explosive processes, *i.e.* in all finite intervals $[0, t)$ only finitely many jumps can occur. Stochastic models of BRNs are trivially non-explosive.

Chapter B
A DNA model: equation for the derivative of the lower bound on the concentration of x_2

According to Def.6.3.7, the derivative of the lower bound on the concentration of x_2 is computed by selecting the minimum region-dependent (*i.e.*, local) lower bound, out of the 9 possible cases:

$$\frac{d\underline{x_2}}{dt} = \min(t_{\downarrow}^{1,1}, t_{\downarrow}^{1,2}, t_{\downarrow}^{1,3}, t_{\downarrow}^{2,1}, t_{\downarrow}^{2,2}, t_{\downarrow}^{2,3}, t_{\downarrow}^{3,1}, t_{\downarrow}^{3,2}, t_{\downarrow}^{3,3})$$

with

$$\begin{aligned} t_{\downarrow}^{1,1} = &\max(k_1 \underline{x_1}^2, k_{-2}(\frac{DNA_0}{\epsilon} - \frac{k_{-1}}{k_2}))- \\ &(1+\epsilon)k_{-1} \cdot \min(\underline{x_2}, \frac{M_0 - \underline{x_1}}{2}, \frac{M_0}{2} - DNA_0 - \frac{\underline{x_1}}{2} - \epsilon\frac{k_{-1}}{k_2}); \end{aligned}$$

$$\begin{aligned} t_{\downarrow}^{1,2} = &\max(k_1 \underline{x_1}^2, k_{-2}\frac{DNA_0 - \overline{x_3 u}}{\epsilon})- \\ &(1+\epsilon)k_2 \cdot \min(\underline{x_2}, \frac{M_0 - \underline{x_1}}{2}, \frac{M_0 - 2DNA_0 - \underline{x_1} + 2\overline{x_3}}{2})\cdot \\ &\cdot \min(\overline{x_3}, DNA_0) \end{aligned}$$

$$\begin{aligned} t_{\downarrow}^{1,3} = &\max(k_1 \underline{x_1}^2, (\frac{k_{-2}}{\epsilon}(DNA_0 - \frac{k_{-1}}{\epsilon k_2}), \frac{k_{-2}}{\epsilon}(DNA_0 - \overline{x_3}))- \\ &\min(\underline{x_2}, \frac{M_0 - \underline{x_1}}{2} + \min(\overline{x_3} - DNA_0, \frac{k_2}{\epsilon k_{-1}} - DNA_0)))\cdot \\ &(k_2 \cdot \min(\overline{x_3}, DNA_0) + k_{-1}); \end{aligned}$$

$$t_{\downarrow}^{2,1} = k_{-2} \cdot \max(\underline{x_4}, DNA_0 - \frac{\epsilon k_{-1}}{k_2}) - (1+\epsilon) k_{-1} \cdot \min(\underline{x_2}, \frac{M_0}{2} - \max(\underline{x_4}, DNA_0 - \epsilon \frac{k_{-1}}{k_2}))$$

$$t_{\downarrow}^{2,2} = k_{-2} \cdot \max(\underline{x_4}, DNA_0 - \overline{x_3}) - (1+\epsilon) k_2 \cdot \min(\underline{x_2}, \frac{M_0}{2} - \underline{x_4}, \frac{M_0}{2} - DNA_0 + \overline{x_3}) \cdot \min(\overline{x_3}, DNA_0 - \underline{x_4})$$

$$t_{\downarrow}^{2,3} = k_{-2} \cdot \max(\underline{x_4}, DNA_0 - \overline{x_3}, DNA_0 - \frac{k_{-1}}{\epsilon k_2})) - \min(\underline{x_2}, \frac{M_0}{2} - \max(\underline{x_4}, DNA_0 - \overline{x_3}, DNA_0 - \frac{k_{-1}}{\epsilon k_2})) (k_{-1} + k_2 \cdot \min(\overline{x_3}, DNA_0 - \underline{x_4}));$$

$$t_{\downarrow}^{3,1} = \max(\epsilon k_{-2} \underline{x_4}, k_1 {\underline{x_1}}^2) + \max(\epsilon k_1 {\underline{x_1}}^2, k_{-2} \cdot (DNA_0 - \epsilon k_{-1} \frac{\underline{x_2}}{k_2})) - (1+\epsilon) k_{-1} \cdot \min(\underline{x_2}, \frac{M_0 - \underline{x_1}}{2} - \epsilon k_1 \cdot {\underline{x_1}}^2 k_{-2});$$

$$t_{\downarrow}^{3,2} = \max(k_1 {\underline{x_1}}^2, \epsilon k_{-2} \underline{x_4}, \epsilon k_{-2}(DNA_0 - \overline{x_3})) + k_{-2} \cdot \max(\underline{x_4}, DNA_0 - \overline{x_3}, \epsilon k_1 \frac{{\underline{x_1}}^2}{k_{-2}}) - (1+\epsilon) k_2 \cdot \min(\underline{x_2}, \frac{M_0 - \underline{x_1}}{2} - \max(\underline{x_4}, \epsilon k_1 \frac{{\underline{x_1}}^2}{k_{-2}}, DNA_0 - \overline{x_3}))) \cdot \min(\overline{x_3}, DNA_0 - \max(\underline{x_4}, \epsilon k_1 \frac{{\underline{x_1}}^2}{k_{-2}}));$$

$$
\begin{aligned}
t_{\downarrow}^{3,3} =& k_1 \underline{x_1}^2 + k_{-2} \cdot \max(\underline{x_4}, DNA_0 - \overline{x_3}) - \\
& k_2 \cdot \min(\underline{x_2}, \frac{M_0 - \underline{x_1}}{2} - \max(\underline{x_4}, DNA_0 - \overline{x_3})) \cdot \min(\overline{x_3}, DNA_0 - \underline{x_4}) - \\
& k_{-1} \cdot \min(\underline{x_2}, \frac{M_0 - \underline{x_1} - 2 \cdot \underline{x_4}}{2});
\end{aligned}
$$

Bibliography

[1] Computer assisted proofs in dynamics (capd) library. `http://capd.ii.uj.edu.pl`. [Online; accessed 14-January-2019].

[2] R. Albert and H. G. Othmer. The topology of the regulatory interactions predicts the expression pattern of the segment polarity genes in drosophila melanogaster. *Journal of Theoretical Biology*, 223(1):1 – 18, 2003.

[3] R. Alur, C. Courcoubetis, T. A. Henzinger, and P.-H. Ho. Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems. In *Hybrid systems*, pages 209–229. Springer, 1993.

[4] D. F. Anderson. Incorporating postleap checks in tau-leaping. *The Journal of Chemical Physics*, 128(5), 2008.

[5] D. F. Anderson and T. G. Kurtz. Continuous time markov chain models for chemical reaction networks. In *Design and analysis of biomolecular circuits*, pages 3–42. Springer, 2011.

[6] S. Andrews. Spatial and stochastic cellular modeling with the smoldyn simulator. *Methods in molecular biology (Clifton, N.J.)*, 804:519–42, 10 2012.

[7] P. Arányi and J. Tóth. A full stochastic description of the michaelis-menten reaction for small systems. *Acta biochimica et biophysica; Academiae Scientiarum Hungaricae*, 12(4):375–388, 1977.

[8] A. Arkin, J. Ross, and H. H. McAdams. Stochastic kinetic analysis of developmental pathway bifurcation in phage λ-infected escherichia coli cells. *Genetics*, 149(4):1633–1648, 1998.

[9] F. Baccelli, G. Cohen, G. J. Olsder, and J.-P. Quadrat. Synchronization and linearity: an algebra for discrete event systems. 1992.

[10] E. Bartocci and P. Lió. Computational Modeling, Formal Analysis, and Tools for Systems Biology. *PLOS Computational Biology*, 12(1):1–22, 01 2016.

[11] A. Beica and V. Danos. Synchronous balanced analysis. In E. Cinquemani and A. Donzé, editors, *Hybrid Systems Biology*, pages 85–94, Cham, 2016. Springer International Publishing.

[12] A. Beica, J. Feret, and T. Petrov. Tropical abstraction of biochemical reaction networks with guarantees. *arXiv preprint arXiv:1812.11405*, 2018.

[13] A. Beica, C. C. Guet, and T. Petrov. Efficient reduction of kappa models by static inspection of the rule-set. In *International Workshop on Hybrid Systems Biology*, pages 173–191. Springer, 2015.

[14] A. Beica and T. Petrov. Kared, 2014. `https://github.com/andreeabeica/KappaRed`.

[15] B. Bennett. Absolute metabolite concentrations and implied enzyme active site occupancy in escherichia coli. *Nature Chemical Biology*, 2009.

[16] A. Beyer. *Network-Based Models in Molecular Biology*, pages 35–56. Birkhauser, 02 2009.

[17] A. T. Bittig and A. M. Uhrmacher. Ml-space: Hybrid spatial gillespie and particle simulation of multi-level rule-based models in cell biology. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 14(6):1339–1349, Nov 2017.

[18] M. L. Blinov, J. R. Faeder, B. Goldstein, and W. S. Hlavacek. Bionetgen: software for rule-based modeling of signal transduction based on the interactions of molecular domains. *Bioinformatics*, 20(17):3289–3291, 2004.

[19] M. Bonora, S. Patergnani, A. Rimessi, E. De Marchi, J. M. Suski, A. Bononi, C. Giorgi, S. Marchi, S. Missiroli, F. Poletti, M. R. Wieckowski, and P. Pinton. Atp synthesis and storage. *Purinergic Signalling*, 8(3):343–357, Sep 2012.

[20] N. M. Borisov, N. I. Markevich, J. B. Hoek, and B. N. Kholodenko. Signaling through receptors and scaffolds: independent interactions reduce combinatorial complexity. *Biophysical journal*, 89(2):951–966, 2005.

[21] C. A. Brackley, D. S. Broomhead, M. C. Romano, and M. Thiel. A max-plus model of ribosome dynamics during mrna translation. *Journal of Theoretical Biology*, 303:128–140, 2012.

[22] H. Bremer, P. P. Dennis, et al. Modulation of chemical composition and other parameters of the cell by growth rate. *E Coli Salmonella Cell Mol Biol 01*, 1996.

[23] G. E. Briggs and J. B. S. Haldane. A note on the kinetics of enzyme action. *Biochemical journal*, 19(2):338, 1925.

[24] H. Brunschede, T. Dove, and H. Bremer. Establishment of exponential growth after a nutritional shift-up in escherichia coli b/r: accumulation of deoxyribonucleic acid, ribonucleic acid, and protein. *Journal of bacteriology*, 129(2):1020–1033, 1977.

[25] J. R. Burch, E. M. Clarke, K. L. McMillan, D. L. Dill, and L.-J. Hwang. Symbolic model checking: 10^{20} states and beyond. *Information and computation*, 98(2):142–170, 1992.

[26] J. Butcher. Numerical methods for differential equations and applications, 1997.

[27] C. Campbell, S. Yang, R. Albert, and K. Shea. A network model for plant–pollinator community assembly. *Proceedings of the National Academy of Sciences*, 108(1):197–202, 2011.

[28] F. Camporesi, J. Feret, and K. Q. Lỳ. K a de: A tool to compile kappa rules into (reduced) ode models. In *International Conference on Computational Methods in Systems Biology*, pages 291–299. Springer, 2017.

[29] Y. Cao, D. T. Gillespie, and L. R. Petzold. Avoiding negative populations in explicit poisson tau-leaping. *The Journal of chemical physics*, 123(5):054104, 2005.

[30] Y. Cao, D. T. Gillespie, and L. R. Petzold. Efficient step size selection for the tau-leaping simulation method. *The Journal of Chemical Physics*, 124(4):044109, 2006.

[31] Y. Cao, H. Li, and L. Petzold. Efficient formulation of the stochastic simulation algorithm for chemically reacting systems. *The Journal of Chemical Physics*, 121(9):4059–4067, 2004.

[32] Y. Cao and L. Petzold. Trapezoidal tau-leaping formula for the stochastic simulation of biochemical systems. *Proceedings of Foundations of Systems Biology in Engineering (FOSBE 2005)*, pages 149–152, 01 2005.

[33] Y. Cao, D. T Gillespie, and L. Petzold. Accelerated stochastic simulation of the stiff enzyme-substrate reaction. *The Journal of chemical physics*, 123:144917, 11 2005.

[34] Y. Cao, D. T Gillespie, and L. Petzold. The slow-scale stochastic simulation algorithm. *The Journal of chemical physics*, 122:14116, 02 2005.

[35] J. Carrera and M. W Covert. Why build whole-cell models? *Trends in cell biology*, 25, 10 2015.

[36] F. Cazals and P. Kornprobst. *Modeling in computational biology and biomedicine. A multidisciplinary endeavor.* Springer, 03 2013.

[37] K. Chanseau Saint-Germain and J. Feret. Conservative numerical approximations of the differential semantics in biological rule-based models, 2016.

[38] C. Chaouiya. Petri net modelling of biological networks. *Briefings in bioinformatics*, 8(4):210–219, 2007.

[39] A. Chatterjee, D. G. Vlachos, and M. A. Katsoulakis. Binomial distribution based τ-leap accelerated stochastic simulation. *The Journal of Chemical Physics*, 122(2), 2005.

[40] P. Chesson. Multispecies competition in variable environments. *Theoretical Population Biology*, 45(3):227 – 276, 1994.

[41] B. Choi, G. A. Rempala, and J. K. Kim. Beyond the Michaelis-Menten equation: Accurate and efficient estimation of enzyme kinetic parameters. *Scientific reports*, 7(1):17018, 2017.

[42] G. Cohen, S. Gaubert, and J.-P. Quadrat. Algebraic system analysis of timed Petri nets. *Idempotency*, Collection of the Isaac Newton Institute, Cambridge, University Press, 1998.

[43] E. Conrad and J. Tyson. *Modeling Molecular Interaction Networks with Nonlinear Ordinary Differential Equations*, chapter 6, pages 97–123. MITP, 2006.

[44] S. Cooper. Cell division and dna replication following a shift to a richer medium. *Journal of molecular biology*, 43(1):1–11, 1969.

[45] P. Cousot. Abstract interpretation based formal methods and future challenges. In *Informatics - 10 Years Back. 10 Years Ahead.*, pages 138–156, London, UK, 2001. Springer-Verlag.

[46] G. Craciun and M. Feinberg. Multiple equilibria in complex chemical reaction networks: Ii. the species-reaction graph. *SIAM Journal on Applied Mathematics*, 66(4):1321–1338, 2006.

[47] A. Crudu, A. Debussche, and O. Radulescu. Hybrid stochastic simplifications for multiscale gene networks. *BMC systems biology*, 3(1):89, 2009.

[48] D. G. Dalbow and R. Young. Synthesis time of β-galactosidase in escherichia coli b/r as a function of growth rate. *Biochemical Journal*, 150(1):13–20, 1975.

[49] V. Danos, J. Feret, W. Fontana, R. Harmer, and J. Krivine. Rule-based modelling of cellular signalling, invited paper. In *Proceedings of the Eighteenth International Conference on Concurrency Theory, CONCUR '2007, Lisbon, Portugal*, volume 4703 of *Lecture Notes in Computer Science*, pages 17–41, Lisbon, Portugal, 3–8 September 2007. Springer, Berlin, Germany.

[50] V. Danos, J. Feret, W. Fontana, and J. Krivine. Scalable simulation of cellular signaling networks. In *Asian Symposium on Programming Languages and Systems*, pages 139–157. Springer, 2007.

[51] V. Danos, J. Feret, W. Fontana, and J. Krivine. Abstract interpretation of reachable complexes in biological signalling networks. In *Proceedings of the 9th International Conference on Verification, Model Checking and Abstract Interpretation (VMCAI'08)*, volume 4905, pages 42–58, 2008.

[52] V. Danos and C. Laneve. Core formal molecular biology. *Theoretical Computer Science*, 325:69–110, 2003.

[53] V. Danos and C. Laneve. Formal molecular biology. *Theoretical Computer Science*, 325(1):69–110, 2004.

[54] T. A. Darden. *A pseudo-steady-state approximation for stochastic chemical kinetics*. PhD thesis, University of California, Berkeley, 1979.

[55] R. David and H. Alla. Continuous Petri Nets. *Proc. 8th Int. Work. on Applications and Theory of Petri Nets*, pages 275–294, 1987.

[56] R. David and H. Alla. Continuous and Hybrid Petri Nets. *Journal of Circuits, Systems, and Computers*, 8:159–188, 1998.

[57] D. Del Vecchio. Design and analysis of an activator-repressor clock in e. coli. In *American Control Conference, 2007. ACC'07*, pages 1589–1594. IEEE, 2007.

[58] P. P. Dennis and H. Bremer. Macromolecular composition during steady-state growth of escherichia coli b/r. *Journal of bacteriology*, 119(1):270–281, 1974.

[59] H. Dong, L. Nilsson, and C. G. Kurland. Co-variation of trna abundance and codon usage inescherichia coliat different growth rates. *Journal of molecular biology*, 260(5):649–663, 1996.

[60] M. R. Droop. Vitamin b12 and marine ecology. iv. the kinetics of uptake, growth and inhibition in monochrysis lutheri. *Journal of the Marine Biological Association of the United Kingdom*, 48(3):689–733, 1968.

[61] A. N. Edwards, L. M. Patterson-Fortin, C. A. Vakulskas, J. W. Mercante, K. Potrykus, D. Vinella, M. I. Camacho, J. A. Fields, S. A. Thompson, D. Georgellis, et al. Circuitry linking the csr and stringent response global regulatory systems. *Molecular microbiology*, 80(6):1561–1580, 2011.

[62] J. Feret, V. Danos, J. Krivine, R. Harmer, and W. Fontana. Internal coarse-graining of molecular systems. *Proceedings of the National Academy of Sciences*, 106(16):6453–6458, April 2009.

[63] J. Feret, T. Henzinger, H. Koeppl, and T. Petrov. Lumpability abstractions of rule-based systems. *Theor. Comput. Sci.*, 431:137–164, May 2012.

[64] J. Fisher and T. A. Henzinger. Executable cell biology. *Nature biotechnology*, 25(11):1239, 2007.

[65] J. Fisher, T. A. Henzinger, M. Mateescu, and N. Piterman. Bounded asynchrony: Concurrency for modeling cell-cell interactions. In *International Workshop on Formal Methods in Systems Biology*, pages 17–32. Springer, 2008.

[66] J. Forchhammer and L. Lindahl. Growth rate of polypeptide chains as a function of the cell growth rate in a mutant of escherichia coli 15. *Journal of molecular biology*, 55(3):563–568, 1971.

[67] S. A. Frank. The trade-off between rate and yield in the design of microbial metabolism. *Journal of Evolutionary Biology*, 23(3):609–613, 2010.

[68] M. Franzle, C. Herde, T. Teige, S. Ratschan, and T. Schubert. Efficient solving of large non-linear arithmetic constraint systems with complex boolean structure. *Journal on Satisfiability, Boolean Modeling and Computation*, 1:209–236, 2007.

[69] A. Ganguly, T. Petrov, and H. Koeppl. Markov chain aggregation and its applications to combinatorial reaction networks. *Journal of mathematical biology*, November 2013.

[70] S. Gao, S. Kong, and E. M. Clarke. dReal: An SMT solver for nonlinear theories over the reals. In *International conference on automated deduction*, pages 208–214. Springer, 2013.

[71] T. S. Gardner, C. R. Cantor, and J. J. Collins. Construction of a genetic toggle switch in Escherichia coli. *Nature*, 403(6767):339–342, Jan 2000.

[72] M. R. Garey, D. S. Johnson, and R. Sethi. The complexity of flowshop and jobshop scheduling. *Mathematics of operations research*, 1(2):117–129, 1976.

[73] C. E. Giacomantonio and G. J. Goodhill. A boolean model of the gene regulatory network underlying mammalian cortical area development. *PLOS Computational Biology*, 6(9):1–13, 09 2010.

[74] M. A. Gibson and J. Bruck. Efficient exact stochastic simulation of chemical systems with many species and many channels. *The Journal of Physical Chemistry A*, 104(9):1876–1889, 2000.

[75] D. Gilbert and M. Heiner. From Petri nets to differential equations - An integrative approach for biochemical network analysis. *Lecture Notes in Computer Science, Conference: 27th International Conference on Applications and Theory of Petri Nets*, pages 181–200, 01 2006.

[76] D. Gillespie. Approximate accelerated stochastic simulation of chemically reacting systems. *The Journal of Chemical Physics*, 115:1716–1733, 2001.

[77] D. T. Gillespie. A general method of numerically simulating the stochastic time evolution of coupled chemical reactions. *Journal of Computational Physics*, 22:403–434, 12 1976.

[78] D. T. Gillespie. *Markov Processes: An Introduction for Physical Scientists*. Academic Press, San Diego, 1992.

[79] D. T. Gillespie. Stochastic simulation of chemical kinetics. *Annual Review of Physical Chemistry*, 58(1):35–55, 2007. PMID: 17037977.

[80] D. T. Gillespie. Deterministic limit of stochastic chemical kinetics. *The Journal of Physical Chemistry B*, 113(6):1640–1644, 2009. PMID: 19159264.

[81] D. T. Gillespie, Y. Cao, K. R. Sanft, and L. R. Petzold. The subtle business of model reduction for stochastic chemical kinetics. *The Journal of chemical physics*, 130(6):064103, 2009.

[82] N. Giordano, F. Mairet, J.-L. Gouzé, J. Geiselmann, and H. de Jong. Dynamical allocation of cellular resources as an optimal control problem: Novel insights into microbial growth strategies. *PLOS Computational Biology*, 12(3):1–28, 03 2016.

[83] D. Green. Cellular automata models in biology. *Mathematical and Computer Modelling*, 13(6):69–74, 1990.

[84] R. Grima. Investigating the robustness of the classical enzyme kinetic equations in small intracellular compartments. *BMC systems biology*, 3(1):101, 2009.

[85] R. Grima. Noise-induced breakdown of the michaelis-menten equation in steady-state conditions. *Physical review letters*, 102(21):218103, 2009.

[86] R. Grima and A. Leier. Exact product formation rates for stochastic enzyme kinetics. *The Journal of Physical Chemistry B*, 121(1):13–23, 2016.

[87] C. C. Guet, M. B. Elowitz, W. Hsing, and S. Leibler. Combinatorial synthesis of genetic networks. *Science*, 296(5572):1466–1470, 2002.

[88] G. Hardin. The competitive exclusion principle. *Science*, 131(3409):1292–1297, 1960.

[89] D. Harel. Statecharts: A visual formalism for complex systems. *Science of computer programming*, 8(3):231–274, 1987.

[90] E. L. Haseltine and J. B. Rawlings. Approximate simulation of coupled fast and slow reactions for stochastic chemical kinetics. *The Journal of chemical physics*, 117(15):6959–6969, 2002.

[91] H.Conzelmann, J.Saez-Rodriguez, T.Sauter, B.N.Kholodenko, and E. Gilles. A domain-oriented approach to the reduction of combinatorial complexity in signal transduction networks. *BMC Bioinformatics*, 7, 2006.

[92] M. Heiner, D. Gilbert, and R. Donaldson. Petri Nets for Systems and Synthetic Biology. *Formal Methods for Computational Systems Biology*, pages 215–264, 2008.

[93] B. Hepp, A. Gupta, and M. Khammash. Adaptive hybrid simulations for multiscale stochastic reaction networks. *arXiv preprint arXiv:1402.3523*, 2014.

[94] W. Hlavacek, J. Faeder, M. L Blinov, A. Perelson, and B. Goldstein. The complexity of complexes in signal transduction. *Biotechnology and bioengineering*, 84:783–94, 12 2003.

[95] P. Hogeweg. The roots of bioinformatics in theoretical biology. *PLOS Computational Biology*, 7(3):1–5, 03 2011.

[96] J. S. Hogg, L. A. Harris, L. J. Stover, N. S. Nair, and J. R. Faeder. Exact hybrid particle/population simulation of rule-based models of biochemical systems. *PLoS computational biology*, 10(4):e1003544, 2014.

[97] S. Hui, J. M. Silverman, S. S. Chen, D. W. Erickson, M. Basan, J. Wang, T. Hwa, and J. R. Williamson. Quantitative proteomic analysis reveals a simple strategy of global resource allocation in bacteria. *Molecular Systems Biology*, 11(2), 2015.

[98] T. Ideker, T. Galitski, and L. Hood. A new approach to decoding life: Systems biology. *Annual review of genomics and human genetics*, 2:343–72, 02 2001.

[99] B. Ingalls. *Mathematical Modelling in Systems Biology: An Introduction*. MIT Press, 01 2013.

[100] ISB. `https://systemsbiology.org/about/what-is-systems-biology/`. [Online; accessed 14-January-2019].

[101] D. J Wilkinson. Stochastic modelling for quantitative description of heterogeneous biological systems. *Nature reviews. Genetics*, 10:122–33, 02 2009.

[102] T. Jahnke and W. Huisinga. Solving the chemical master equation for monomolecular reaction systems analytically. *Journal of Mathematical Biology*, 54(1):1–26, Jan 2007.

[103] H.-W. Kang and T. G. Kurtz. Separation of time-scales and model reduction for stochastic reaction networks. *The Annals of Applied Probability*, 23(2):529–583, 2013.

[104] J. R. Karr, J. C. Sanghvi, D. N. Macklin, M. V. Gutschow, J. M. Jacobs, B. Bolival, N. Assad-Garcia, J. I. Glass, and M. W. Covert. A whole-cell computational model predicts phenotype from genotype. *Cell*, 150(2):389 – 401, 2012.

[105] J. R. Karr, K. Takahashi, and A. Funahashi. The principles of whole-cell modeling. *Current opinion in microbiology*, 27:18–24, 06 2015.

[106] J. Keener and J. Sneyd. *Mathematical Phsiology I: Cellular Physiology*, volume 2. Springer, 2009.

[107] M. Kirkilionis and S. Walcher. On comparison systems for ordinary differential equations. *Journal of mathematical analysis and applications*, 299(1):157–173, 2004.

[108] H. Kitano. *Foundations of Systems Biology*. MIT Press, 01 2001.

[109] H. Kitano. Systems biology: A brief overview. *Science*, 295(5560):1662–1664, 2002.

[110] E. Krepska, N. Bonzanni, A. Feenstra, W. Fokkink, T. Kielmann, H. Bal, and J. Heringa. Design issues for qualitative modelling of biological cells with petri nets. In *Formal Methods in Systems Biology*, pages 48–62. Springer, 2008.

[111] T. G. Kurtz. Solutions of ordinary differential equations as limits of pure jump markov processes. *Journal of applied Probability*, 7(1):49–58, 1970.

[112] H. Kuwahara, C. J. Myers, M. S. Samoilov, N. A. Barker, and A. P. Arkin. Automated abstraction methodology for genetic regulatory networks. In C. Priami and G. Plotkin, editors, *Transactions on Computational Systems Biology VI*, pages 150–175, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[113] R. Kwok. Five hard truths for synthetic biology. *Nature*, 463(7279):288–290, 2010.

[114] C. H. Lee and P. Kim. An analytical approach to solutions of master equations for stochastic nonlinear reactions. *Journal of Mathematical Chemistry*, 50(6):1550–1569, Jun 2012.

[115] J.-J. Lévy. *Réductions correctes et optimales dans le lambda-calcul*. 1978. Reproduction de Thèse d'Etat Mathématiques. Informatique algébrique Paris 7 1978.

[116] F. Li, T. Long, Y. Lu, Q. Ouyang, and C. Tang. The yeast cell-cycle network is robustly designed. *Proceedings of the National Academy of Sciences*, 101(14):4781–4786, 2004.

[117] H. Li and L. Petzold. Logarithmic direct method for discrete stochastic simulation of chemically reacting systems. *The Journal of Chemical Physics*, 2006.

[118] S. Li, S. M. Assmann, and R. Albert. Predicting essential components of signal transduction networks: A dynamic model of guard cell abscisic acid signaling. *PLOS Biology*, 4(10):1–17, 09 2006.

[119] G. L. Litvinov. Maslov dequantization, idempotent and tropical mathematics: A brief introduction. *Journal of Mathematical Sciences*, 140(3):426–444, 2007.

[120] G. L. Litvinov. Tropical mathematics, idempotent analysis, classical mechanics and geometry. 2011.

[121] M. Lynch and G. K. Marinov. The bioenergetic costs of a gene. *Proceedings of the National Academy of Sciences*, 112(51):15690–15695, 2015.

[122] D. N. Macklin, N. A. Ruggero, and M. W. Covert. The future of whole-cell modeling. *Current Opinion in Biotechnology*, 28:111 – 115, 2014. Nanobiotechnology • Systems biology.

[123] MATLAB. *version R2015b*. The MathWorks Inc., Natick, Massachusetts, 2015.

[124] J. M. McCollum, G. D. Peterson, C. D. Cox, M. L. Simpson, and N. F. Samatova. The sorting direct method for stochastic simulation of biochemical systems with varying reaction execution behavior. *Computational Biology and Chemistry*, 30(1):39 – 49, 2006.

[125] A. McNabb. Comparison theorems for differential equations. *Journal of mathematical analysis and applications*, 119(1-2):417–428, 1986.

[126] D. A. McQuarrie. Stochastic approach to chemical kinetics. *Journal of Applied Probability*, 4(3):413–478, 1967.

[127] L. Michaelis and M. L. Menten. *Die kinetik der invertinwirkung*.

[128] R. Milner. *Communicating and mobile systems: the pi calculus*. Cambridge university press, 1999.

[129] R. Milo. What is the total number of protein molecules per cell volume? a call to rethink some published values. *BioEssays*, 35(12):1050–1055, 2013.

[130] D. Molenaar, R. van Berlo, D. de Ridder, and B. Teusink. Shifts in growth strategies reflect tradeoffs in cellular economics. *Molecular Systems Biology*, 5(1), 2009.

[131] J. Monod. The growth of bacterial cultures. *Annual Review of Microbiology*, 3(1):371–394, 1949.

[132] A. Moraes, R. Tempone, and P. Vilanova. Hybrid chernoff tau-leap. *Multiscale Modeling & Simulation*, 12(2):581–615, 2014.

[133] M. K. Morris, J. Saez-Rodriguez, P. K. Sorger, and D. A. Lauffenburger. Logic-based models for the analysis of cell signaling networks. *Biochemistry*, 49(15):3216–3224, 2010.

[134] J. D. Murray. Mathematical biology i: An introduction, vol. 17 of interdisciplinary applied mathematics, 2002.

[135] C. J. Myers. *Engineering genetic circuits*. CRC Press, 2011.

[136] N. Nedialkov. Vnode-lp: A validated solver for initial value problems in ordinary differential equations. 2006.

[137] N. S. Nedialkov. Interval tools for odes and daes. In *12th GAMM-IMACS International Symposium on Scientific Computing, Computer Arithmetic and Validated Numerics (SCAN 2006)*, pages 4–4. IEEE, 2006.

[138] V. Noël. Modèles réduits et hybrides de réseaux de réactions biochimiques : applications à la modélisation du cycle cellulaire. *PhD Thesis*, 2012.

[139] V. Noel, D. Grigoriev, S. Vakulenko, and O. Radulescu. Tropical geometries and dynamics of biochemical networks application to hybrid cell cycle models. *Electronic Notes in Theoretical Computer Science*, 284:75–91, 2012.

[140] V. Noel, D. Grigoriev, S. Vakulenko, and O. Radulescu. Tropicalization and tropical equilibration of chemical reactions. *Tropical and Idempotent Mathematics and Applications*, 616:261–277, 2014.

[141] M. J. North, N. T. Collier, and J. R. Vos. Experiences creating three implementations of the repast agent modeling toolkit. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 16(1):1–25, 2006.

[142] J. D. Orth, I. Thiele, and B. Ø. Palsson. What is flux balance analysis? *Nature biotechnology*, 28(3):245, 2010.

[143] A. P. Goldberg, B. Szigeti, Y. H. Chew, J. Sekar, Y. Roth, and J. Karr. Emerging whole-cell modeling principles and methods. *Current opinion in biotechnology*, 51, 10 2017.

[144] G. Păun and G. Rozenberg. A guide to membrane computing. *Theoretical Computer Science*, 287(1):73–100, 2002.

[145] S. Pedersen. Escherichia coli ribosomes translate in vivo with variable rate. *The EMBO journal*, 3(12):2895–2898, 1984.

[146] M. Peleg, D. Rubin, and R. B. Altman. Using petri net tools to study properties and dynamics of biological systems. *Journal of the American Medical Informatics Association*, 12(2):181–199, 2005.

[147] C. Petri. Kommunikation mit Automaten. *Bonn: Institut fur Instrumentelle Mathematik, Schriften des IIM*, 2, 1962.

[148] V. Picard. *Réseaux de réactions: de l'analyse probabiliste à la réfutation.* PhD thesis, Université Rennes 1, 2015.

[149] J. K. Pierre Boutillier, Jérôme Feret and W. Fontana. The kappa language and kappa tools: A user manual and guide, 2008-2018. `http://www.kappalanguage.org`.

[150] L. Popova-Zeugmann, M. Heiner, and I. Koch. Time Petri Nets for Modelling and Analysis of Biochemical Networks. *Fundamenta Informaticae*, 67:149–162, 2005.

[151] M. Ptashne. A genetic switch: Gene control and phage lambda. 1986.

[152] M. A. Ptashne. Genes & signals. 2001.

[153] R. Pugatch. Greedy scheduling of cellular self-replication leads to optimal doubling times with a log-frechet distribution. *Proceedings of the National Academy of Sciences*, 112(8):2611–2616, 2015.

[154] O. Radulescu, A. N. Gorban, A. Zinovyev, and V. Noel. Reduction of dynamical biochemical reactions networks in computational biology. *Frontiers in genetics*, 3:131, 2012.

[155] O. Radulescu, S. Vakulenko, and D. Grigoriev. Model reduction of biochemical reactions networks by tropical analysis methods. *Mathematical Modelling of Natural Phenomena*, 10(3):124–138, 2015.

[156] R. Ramaswamy and I. F. Sbalzarini. A partial-propensity variant of the composition-rejection stochastic simulation algorithm for chemical reaction networks. *The Journal of Chemical Physics*, 132(4):044102, 2010.

[157] C. V. Rao and A. P. Arkin. Stochastic chemical kinetics and the quasi-steady-state assumption: Application to the gillespie algorithm. *Journal of Chemical Physics*, 118(11):4999–5010, 2003.

[158] W. C. Ratcliff and R. F. Denison. Individual-level bet hedging in the bacterium sinorhizobium meliloti. *Current Biology*, 20(19):1740 – 1744, 2010.

[159] M. Rathinam, L. R. Petzold, Y. Cao, and D. T. Gillespie. Stiffness in stochastic chemically reacting systems: The implicit tau-leaping method. *The Journal of Chemical Physics*, 119(24):12784–12794, 2003.

[160] A. Regev, E. M. Panina, W. Silverman, L. Cardelli, and E. Shapiro. Bioambients: an abstraction for biological compartments. *Theoretical Computer Science*, 325(1):141–167, 2004.

[161] P. Richmond, D. Walker, S. Coakley, and D. Romano. High performance cellular level agent-based simulation with flame for the gpu. *Briefings in bioinformatics*, 11(3):334–347, 2010.

[162] J. Saez-Rodriguez, L. Simeoni, J. A. Lindquist, R. Hemenway, U. Bommhardt, B. Arndt, U.-U. Haus, R. Weismantel, E. D. Gilles, S. Klamt, and B. Schraven. A logical model provides insights into t cell receptor signaling. *PLOS Computational Biology*, 3(8):1–11, 08 2007.

[163] R. Samaga, J. Saez-Rodriguez, L. G. Alexopoulos, P. K. Sorger, and S. Klamt. The logic of egfr/erbb signaling: Theoretical properties and analysis of high-throughput data. *PLOS Computational Biology*, 5(8):1–19, 08 2009.

[164] K. Sanft, D. Gillespie, and L. Petzold. Legitimacy of the stochastic michaelis-menten approximation. *Systems Biology, IET*, 5:58 – 69, 02 2011.

[165] H. Sauro, D. Harel, M. Z. Kwiatkowska, C. Shaffer, A. Uhrmacher, M. Hucka, P. Mendes, L. Strömbäck, and J. J. Tyson. Challenges for modeling and simulation methods in systems biology. *Proceedings - Winter Simulation Conference*, pages 1720–1730, 12 2006.

[166] R. Schlatter, K. Schmich, I. Avalos Vizcarra, P. Scheurich, T. Sauter, C. Borner, M. Ederer, I. Merfort, and O. Sawodny. On/off and beyond - a boolean model of apoptosis. *PLOS Computational Biology*, 5(12):1–13, 12 2009.

[167] M. Scott, C. W. Gunderson, E. M. Mateescu, Z. Zhang, and T. Hwa. Interdependence of cell growth and gene expression: origins and consequences. *Science*, 330(6007):1099–1102, 2010.

[168] L. A. Segel and M. Slemrod. The quasi-steady-state assumption: a case study in perturbation. *SIAM review*, 31(3):446–477, 1989.

[169] O. Sinnen. *Task scheduling for parallel systems*, volume 60. John Wiley & Sons, 2007.

[170] L. Sonneborn and F. Van Vleck. The bang-bang principle for linear control systems. *Journal of the Society for Industrial and Applied Mathematics, Series A: Control*, 2(2):151–159, 1964.

[171] F. Spill, P. K. Maini, and H. M. Byrne. Optimisation of simulations of stochastic processes by removal of opposing reactions. *The Journal of Chemical Physics*, 144(8):084105, 2016.

[172] M. O. Stéfanini, A. J. McKane, and T. J. Newman. Single enzyme pathways and substrate fluctuations. *Nonlinearity*, 18(4):1575, 2005.

[173] J. Stelling. *System Modeling in Cellular Biology : From Concepts to Nuts and Bolts*. Cambridge: MIT Press, 2006.

[174] G. Terradot, A. Beica, A. Weiße, and V. Danos. Survival of the fattest: Evolutionary trade-offs in cellular resource storage. *Electr. Notes Theor. Comput. Sci.*, 335:91–112, 2018.

[175] P. Thomas, A. V. Straube, and R. Grima. Stochastic theory of large-scale enzyme-reaction networks: Finite copy number corrections to rate equation models. *The Journal of chemical physics*, 133(19):11B607, 2010.

[176] P. Thomas, A. V. Straube, and R. Grima. Communication: limitations of the stochastic quasi-steady-state approximation in open biochemical reaction networks, 2011.

[177] M. Tomita. Whole-cell simulation: A grand challenge of the 21st century. *Trends in biotechnology*, 19:205–10, 07 2001.

[178] J. J. Tyson. Modeling the cell division cycle: cdc2 and cyclin interactions. *Proceedings of the National Academy of Sciences*, 88(16):7328–7332, 1991.

[179] B. Volkmer and M. Heinemann. Condition-dependent cell volume and concentration of escherichia coli to facilitate data conversion for systems biology modeling. *PloS one*, 6(7):e23126, 2011.

[180] R.-S. Wang, A. Saadatpour, and R. Albert. Boolean modeling in systems biology: an overview of methodology and applications. *Physical Biology*, 9(5):055001, sep 2012.

[181] S. Watterson, S. Marshall, and P. Ghazal. Logic models of pathway biology. *Drug Discovery Today*, 13(9):447 – 456, 2008.

[182] A. Weisse, D. Oyarzún, V. Danos, and P. S. Swain. Mechanistic links between cellular trade-offs, gene expression, and growth. *PNAS*, 112, 02 2015.

[183] J. F. Wilkinson. Carbon and energy storage in bacteria. *Microbiology*, 32(2):171–176, 1963.

[184] J. Wilson-Kanamori, V. Danos, T. Thomson, and R. Honorato-Zimmer. Kappa rule-based modeling in synthetic biology. In *Computational Methods in Synthetic Biology*, pages 105–135. Springer, 2015.

[185] M. L. Wynn, N. Consul, S. D. Merajver, and S. Schnell. Logic-based models in systems biology: a predictive and parameter-free network analysis method. *Integrative biology*, 4(11):1323–1337, 2012.

[186] H. Yaginuma. Diversity in ATP concentrations in a single bacterial cell population revealed by quantitative single-cell imaging. *Scientific Reports*, 2014.

[187] R. Young and H. Bremer. Polypeptide-chain-elongation rate in Escherichia coli B/r as a function of growth rate. *Biochemical Journal*, 160(2):185–194, 1976.

List of Figures

List of Tables

RÉSUMÉ

Cette thèse vise á étudier deux aspects liés á la modélisation des Réseaux de Réactions Biochimiques.

Dans un premier temps, nous montrons comment la séparation des échelles de temps et de concentration dans les systèmes biologiques peut être utilisée pour la réduction de modèles. Nous proposons l'utilisation des modèles par régles de réécriture pour le prototypage de circuits génétiques, puis nous exploitons le caractère multi-échelle de tels systèmes pour construire une méthode générale d'approximation de modèles. La réduction est effectuée via une analyse statique du système de règles. Notre heuristique de réduction repose sur des justifications physiques solides. Cependant, tout comme pour d'autres techniques de réduction de modèles exploitant la séparation des échelles, on note la manque de méthodes précises pour quantifier l'erreur d'approximation, tout en évitant de résoudre le modèle original.

C'est pourquoi nous proposons ensuite une méthode d'approximation dans laquelle les garanties de réduction représentent l'exigence majeure. Cette seconde méthode combine abstraction et approximation numérique, et vise á fournir une meilleure compréhension des méthodes de réduction de modèles basées sur une séparation des échelles de temps et de concentration.

Dans la deuxième partie du manuscrit, nous proposons une nouvelle technique de reparamétrisation pour les modèles d'équations différentielles des réseaux biochimiques, afin d'étudier l'effet des stratégies de stockage de ressources intracellulaires sur la croissance, dans des modèles mécanistiques d'auto-réplication cellulaire. Enfin, nous posons des bases pour la caractérisation de la croissance cellulaire en tant que propriété émergeante d'une nouvelle sémantique des réseaux de Petri modélisant des réseaux de réactions biochimiques.

MOTS CLÉS

réseaux de réactions biochimiques, séparation des échelles temps et concentration, réduction de modèles, croissance cellulaire, stockage de resources cellulaires, modèles mécanistiques de réplication cellulaire

ABSTRACT

This thesis aims at studying two aspects related to the modelling of Biochemical Reaction Networks, in the context of Systems Biology.

In the first part, we analyse how scale-separation in biological systems can be exploited for model reduction. We first argue for the use of rule-based models for prototyping genetic circuits, and then show how the inherent multi-scaleness of such systems can be used to devise a general model approximation method for rule-based models of genetic regulatory networks. The reduction proceeds via static analysis of the rule system.

Our method relies on solid physical justifications, however not unlike other scale-separation reduction techniques, it lacks precise methods for quantifying the approximation error, while avoiding to solve the original model. Consequently, we next propose an approximation method for deterministic models of biochemical networks, in which reduction guarantees represent the major requirement. This second method combines abstraction and numerical approximation, and aims at providing a better understanding of model reduction methods that are based on time- and concentration- scale separation.

In the second part of the thesis, we introduce a new re-parametrisation technique for differential equation models of biochemical networks, in order to study the effect of intracellular resource storage strategies on growth, in self-replicating mechanistic models. Finally, we aim towards the characterisation of cellular growth as an emergent property of a novel Petri Net model semantics of Biochemical Reaction Networks.

KEYWORDS

Biochemical Reaction Networks, multiscaleness, model reduction, cellular growth, cellular resource storage, mechanistic self-replicating cellular models