



HAL
open science

Model Building by Temporal Logic Constraint Solving: Investigation of the Coupling between the Cell Cycle and the Circadian Clock

Pauline Traynard

► **To cite this version:**

Pauline Traynard. Model Building by Temporal Logic Constraint Solving: Investigation of the Coupling between the Cell Cycle and the Circadian Clock . Quantitative Methods [q-bio.QM]. Université Paris Diderot, 2016. English. NNT: . tel-01404060

HAL Id: tel-01404060

<https://theses.hal.science/tel-01404060v1>

Submitted on 28 Nov 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE

présentée à

**L'UNIVERSITÉ PARIS DIDEROT (PARIS 7)
SORBONNE PARIS CITÉ**

pour obtenir le titre de

Docteur

spécialité

Informatique

Model Building by Temporal Logic Constraint Solving: Investigation of the Coupling between the Cell Cycle and the Circadian Clock

soutenue par

PAULINE TRAYNARD

le 10 mai 2016

devant le jury composé de :

Président	M. Ahmed BOUAJJANI
Rapporteurs	M. Attila CSIKÁSZ-NAGY M. Nir PITERMAN
Examineurs	Mme. Laurence CALZONE M. Franck DELAUNAY Mme. Anne SIEGEL
Directeurs	M. François FAGES M. Denis THIEFFRY

Abstract

In this dissertation, we explore the use of temporal logic and model checking in systems biology. Our thesis is that temporal logic provides a powerful language to formalize complex yet imprecise dynamical properties of biological systems and to partly automate model building as a constraint satisfaction problem. We take advantage of this logical paradigm for systems biology to capture properties emerging from complex regulatory networks.

First, we investigate the ability of Computation Tree Logic to verify dynamical properties in asynchronous state transition graphs derived from logical models of the mammalian cell cycle. Logical modeling provides a qualitative and potentially non-deterministic description of a biological system. This feature is useful to account for a variety of dynamical properties, observed in different conditions within a generic model. We develop an approach of iterative property verification to assist the building and updating of logical models.

Then we consider quantitative deterministic models. For such models, oscillatory properties such as pseudo-periods and pseudo-phases are formalized by quantitative constraints in First-Order Linear Time Logic. A continuous model can provide a precise description of the mechanisms governing a complex regulatory network, and a quantitative prediction of its dynamics. However, the classical difficulties associated with this approach are brought by numerous and often poorly characterized kinetic parameters. We address this challenge by implementing two complementary strategies to obtain efficient solving of dynamical constraints over a finite time horizon: the design of useful temporal logic formula patterns associated with dedicated constraint solvers, and some trace simplification rules to safely reduce the size of the traces to analyze.

We show that this approach enables the calibration of high-dimensional models on quantitative single-cell data with an application to model coupling for the mammalian cell cycle and the circadian clock. Understanding the relationships between these two molecular oscillators is an important problem in the field of chronobiology. We draw several coupling hypotheses and investigate their consequences.

Résumé

Cette dissertation explore l'utilisation de la logique temporelle et de la vérification de modèle en biologie des systèmes. Nous soutenons que la logique temporelle constitue un outil puissant pour formaliser des propriétés dynamiques à la fois complexes et imprécises permettant de caractériser un système biologique. Cet outil peut être utilisé pour partiellement automatiser la construction de modèle, comme une résolution d'un problème de satisfaction de contraintes.

Tout d'abord, nous étudions l'emploi de la logique arborescente (Computation Tree Logic) pour vérifier des propriétés dynamiques dans des graphes de transitions d'états discrets asynchrones, dérivés de modèles logiques du cycle cellulaire mammifère. La modélisation logique fournit une description qualitative et potentiellement non-déterministe d'un système biologique. Elle offre un cadre utile pour rendre compte des diverses propriétés dynamiques observées dans différentes conditions au sein d'un modèle générique. Nous développons une approche de vérification itérative de propriétés pour assister la construction et la mise à jour de modèles logiques.

Puis, nous considérons des modèles quantitatifs déterministes. Pour de tels modèles, des propriétés sur les oscillations, telles que pseudo-période ou pseudo-phase, peuvent être formalisées par des contraintes quantitatives en logique du premier ordre à temps linéaire (First-Order Linear Time Logic, FO-LTL). Un modèle continu peut fournir une description précise des mécanismes impliqués dans un réseau de régulations complexe, ainsi qu'une prédiction quantitative de ses dynamiques. Cependant, une difficulté bien connue associée à cette approche provient du grand nombre de paramètres cinétiques, qui sont souvent mal caractérisés. Afin de répondre à ce problème, nous implémentons deux stratégies complémentaires pour améliorer l'efficacité de la résolution de contraintes dynamiques sur un horizon de temps fini. Une première approche exploite la définition d'une liste de motifs de logique temporelle fréquemment utilisés, associés à des solveurs de contraintes dédiés. Dans la deuxième approche, nous définissons et appliquons des règles de simplification de trace permettant de réduire la trace à analyser sans perte d'information.

Nous montrons que cette approche permet de calibrer des modèles de haute dimension sur des données quantitatives sur cellule individuelle. Elle est appliquée au couplage de modèles du cycle cellulaire et de l'horloge circadienne mammifères. Comprendre la relation entre ces deux rythmes moléculaires est un problème important dans le domaine de la chronobiologie. Nous déduisons différentes hypothèses de couplage et étudions leurs conséquences.

Remerciements

La thèse n'est pas un travail solitaire. Je dédie la mienne à celles et ceux qui par leur générosité, leur bonne humeur, par l'intérêt manifesté à l'égard de mon projet et leur implication, m'ont permis de progresser et d'aboutir dans cette phase de « l'apprenti-chercheur ».

Je tiens tout d'abord et spécifiquement à remercier Attila Csikász-Nagy et Nir Piterman pour avoir accepté d'être rapporteurs de ce mémoire, dont la version finale a grandement bénéficié de leur lecture attentive et de leurs précieuses observations. Je suis également reconnaissante à tous les membres du jury d'avoir accepté de se déplacer pour assister à la présentation de ce travail.

A François Fages et Denis Thieffry, mes directeurs de thèse, pour la confiance qu'il m'ont accordée en acceptant d'encadrer ce travail doctoral, pour m'avoir fait profiter de leurs connaissances et bénéficier de leurs multiples conseils, pour le temps qu'ils ont consacré à diriger cette recherche, à relire et corriger tant nos articles que ce présent document, je veux exprimer mes plus chaleureux remerciements. Je leur suis également redevable des conditions de travail privilégiées qu'ils m'ont apportées : entre l'IBENS et Inria, j'ai profité de deux environnements aussi riches que variés - partage qui m'a permis d'élargir mes horizons aussi bien en biologie qu'en informatique. Ils m'ont permis, à travers de multiples déplacements en conférences et en écoles d'été, de découvrir que le domaine de la recherche est un monde riche de personnes brillantes et accessibles, mues par la passion de la science, dont le dynamisme et le talent intellectuel forcent au respect et à l'humilité. Elles incarnent à merveille la devise affectionnée de François : « le plaisir de comprendre comme seule contrainte ».

Je ne veux pas oublier l'ensemble des deux équipes de recherche dont j'ai eu la chance de faire partie et mes collègues dans chaque institut : par les précieux moments partagés autour d'un thé, café, gâteau et autres pintes de bières, sans oublier les barbecues, cours de salsa et balades en raquettes, ils ont contribué à apporter une ambiance chaleureuse. Je remercie particulièrement la joyeuse équipe de sportifs d'Inria, notamment Victorien, Jonathan, Renaud, Sarah et Thierry. J'étais loin d'imaginer que cette préparation au doctorat ferait également de moi une fière candidate au marathon. Nos entraînements assidus et bavards dans la riante forêt de Bailly seront de mes meilleurs souvenirs.

Au terme de ce parcours, mes pensées vont enfin à mes proches et amis qui - bien qu'un peu délaissés ces derniers mois - m'ont toujours adressé leurs encouragements. A mes parents va ma gratitude pour m'avoir offert le goût de la science, poussée à la curiosité intellectuelle, et donné toutes les chances pour réussir. Enfin une pensée pour Charles, pour son accompagnement et son réconfort, notamment durant les derniers mois de rédaction quand il partageait mes insomnies. A toi dont j'ai égoïstement saboté pas mal de soirées, week-ends et autres vacances, merci pour ton soutien indéfectible.

Contents

Contents	6
1 Introduction	8
1.1 Computational Systems Biology	8
1.2 Abstraction of biological systems	9
1.3 Model validation	13
1.4 Outline	17
2 Temporal constraints in the logical framework	18
2.1 Cell cycle	18
2.2 Logical modeling framework	19
2.3 Model analysis	21
2.4 Application: studying the structural properties of the cell cycle	24
2.5 Stochastic simulations of the logical model	32
2.6 Conclusions and prospects	34
3 Formalizing complex behaviors with temporal constraints	36
3.1 First Order Linear Time Logic for continuous trajectories	36
3.2 Formalizing common properties for oscillatory systems with temporal logic	43
3.3 Illustration: parameter analysis of a toy oscillator model	50
3.4 Conclusion	57
4 Efficient solving of FO-LTL(\mathbb{R}_{lin}) constraints	58
4.1 Solving complexity	58
4.2 Dedicated solvers for complex specifications	59
4.3 Trace simplification preserving temporal specifications	73
4.4 Trace simplification for selecting main peaks	80
4.5 Conclusion	84
5 Application to the coupling between the cell cycle and the circadian clock	86
5.1 Introduction	86
5.2 Experimental data and their formal specification in temporal logic	89
5.3 Cell Cycle and Circadian Clock Models	92
5.4 Coupled Model	96
5.5 Alternative hypotheses	103
5.6 Bidirectional coupling	108
5.7 Conclusion	110
6 Conclusion, prospects	112
6.1 Summary	112
6.2 Contributions	113
6.3 Prospects	114

Bibliography	116
A Appendix	127
A.1 Perturbation properties and CTL formulae for the logical model of the cell cycle . .	127
A.2 Toy oscillator model	130
A.3 Models studied in Chapter 5	130

1. Introduction

“ Every object that biology studies is a system of systems.

”
François Jacob, 1974

1.1 Computational Systems Biology

Systems approach for biology

Molecular systems biology: from molecules to processes

Describing biological processes with mathematics is a challenge of increasing importance in modern biology. On the molecular level, biochemical reactions form intra- and intercellular relationships between molecular components and delineate networks of interconnected and mutually dependent molecules.

The study of regulatory networks relies on a systems approach, formalized with computational models. This approach explains the paradoxical tolerance of huge variability of behaviors observed among similar organisms, organs and cells: inherently stochastic individual trajectories are constrained in robust biological systems. Feedback-based constraints cause high-level behaviors to emerge and are responsible for maintaining the stability of global properties.

The approach of building explanatory and predictive computational models from the biochemical elements of living organisms is central in the field of systems biology. The goal of this extensively multidisciplinary research field, fed with techniques borrowed from other disciplines such as non-linear dynamics, statistics, control theory, graph theory, was described by Kitano in 1999: "Gain system-level understanding of multi-scale biological processes in terms of their elementary interactions at the molecular level." Proposing a model accounting for the observed cell responses, or better, predicting novel behaviors, is now regarded as an essential step to validate a proposed mechanism in systems biology.

Cells as complex biochemical integrative circuits

The development of systems biology is fueled with the constant improvement of experimental tools, resulting in informative data on biological systems responses in various environmental and stimulation conditions and at different scales, from single cell to whole organism. At the molecular level in particular, the increasingly precise knowledge of interactions composing molecular networks and their mechanisms derives from the huge amount of biological data generated by high-throughput methods and analyzed by functional genomics. A high variety characterizes these mechanisms: chemical reaction, complex formation, binding, transcriptional regulation, etc.

Abstracting molecules as functional elements that execute operations on a signal helps reducing the complexity.

From this point of view, cells can be seen as biochemical integrative circuits in which complex regulatory networks integrate and process information from multiple external signals, and compute an adequate answer. Accumulation and optimization driven by evolution make them efficient, robust and flexible. The approach of describing biological systems as executable models is detailed in particular in [Fisher and Henzinger, 2007]. The difficulties of understanding such circuits lie in part in the number of molecular species and their interactions, but most of all in the nature of the biochemical computation, which is high dimensional, concurrent, distributed and stochastic, and relies on tight controls to enforce specific cellular responses. In particular, biological processes operate at widely disparate time and spatial scales and commonly involve nonlinearities. Hence, processing biochemical information requires an understanding of control notions such as structural stability, resilience, and robustness.

It creates a strong push for the development of methods that analyze large interaction networks and produce quantitative predictions that match single cell data precision and robustness, and multiscale models that take into account biological variability and population effects. Dedicated mathematical and computational approaches are needed to calibrate models to experimental data.

Formal methods for systems biology

Computational Science, as the field of study concerned with the theoretical foundations of information and computation, brings useful techniques to the modeling and understanding of molecular biological networks. Beyond the algorithm and computing power, able to deal with the high number of behaviors produced by combinatorial composition and stochastic trajectories, and the definition of common standards and methods that structure the systems biology community (such as repositories for data, ontology or biological models), the most important contribution is a wealth of formal methods borrowed from programming theory and designed to deal with complex networks. These methods are increasingly used for the formal analysis of natural processes such as biological behaviors.

Formal methods rely on abstraction to infer properties from complex systems. In particular, methods for the systematic study of the structure, correctness and efficiency of computational systems are adaptable to biological systems.

1.2 Abstraction of biological systems

Model building

Understanding the temporal behavior of biological regulatory networks requires the integration of regulatory data into a formal dynamical model. This abstract representation of a biological system fulfills several essential roles:

- to formalize existing knowledge,
- to bring understanding of dynamical features of biological networks,
- to make predictions on the system,
- to assist the rational design of novel informative experiments.

These roles fall within contradictory perspectives. On the one hand, knowledge representation implies to take into account all known interactions with the maximum of details. However, some relations between molecules sometimes need to be abstracted for lack of knowledge of the precise mechanism. On the other hand, complete, deep, and scalable analysis sometimes

necessitates heavy abstractions. In general, models for making predictions should represent the minimum information sufficient for answering particular questions, in order to maximize the efficiency of available tools. This classic difficulty faced by modeling is highlighted by Olaf Wolkenhauer [Wolkenhauer, 2014]: "The model should be as simple as possible and as detailed as necessary. [...] Thus modeling is the art of making reasonable assumptions and appropriate choices." The same trade-off appears for program analysis and is at the heart of the theory of abstract interpretation [Cousot and Cousot, 1977, Fages and Soliman, 2008]. Therefore each decision has to be made with the objective of a good balance between computational cost and precision of the dynamics. For example, considering gene regulation, the activation of a gene may be modeled by taking into account the different processes from transcription to translation, or by simply considering that the gene is activated. Most of all, a model has to be built adequately to address specific questions.

These objectives are best achieved through an iterative modeling process, ideally forming a cycle of data-driven modeling and model-driven experimentation, as shown in Figure 1.1. Phases of model building, refinement or extension, are interwoven with phases of model validation, where experiments performed on the model allow to verify relevant properties and assess whether the model achieves a useful approximation of the system. A negative outcome reveals the need of changes in the model and give new insights into the underlying mechanisms. Many choices have to be made along this process, starting with the level of precision of the initial model, to the formalism and abstraction used to describe the dynamics, the properties to check, and the successive refinements needed to reach a satisfactory description of the system.

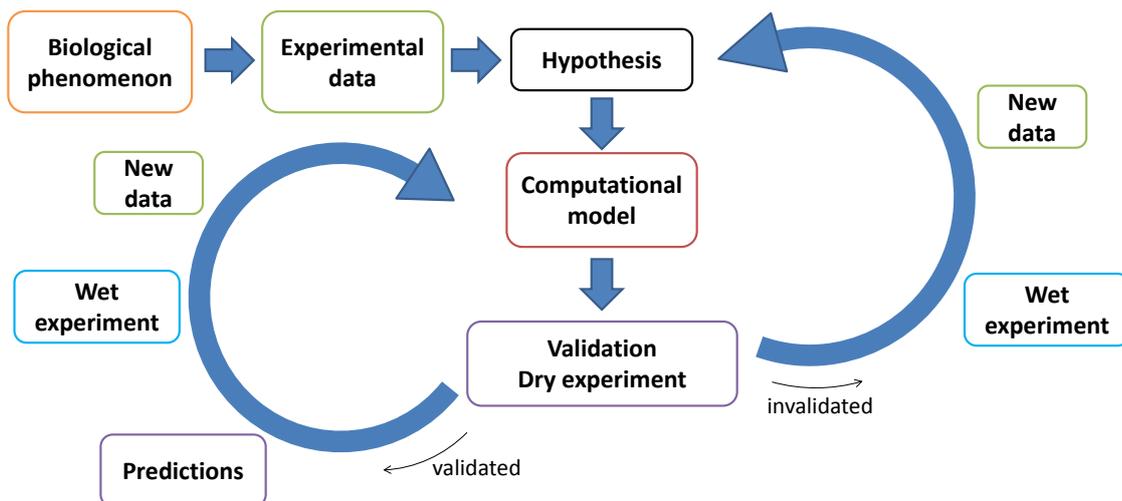


Figure 1.1: The cycle of model building: a computational model is built on experimental data on the basis of hypotheses accounting for missing data or unobserved conditions. These hypotheses have to be validated with well-chosen experiments that generate new data, which are integrated in the model and lead to modifications. If the model is invalidated, the model has to be revised with different hypotheses, potentially with the help of new data.

Hierarchy of semantics

Whatever the chosen level of precision, abstracting a biological system in a formalized model is done in several steps. It starts with a regulatory map that integrates and organizes the knowledge on the system. Tools like CellDesigner [Funahashi et al., 2008] facilitate the construction and maintenance of large or detailed maps. The map is then abstracted into an influence graph constituting the model structure with chosen components and describing the relationships between

modeled quantities. Finally, the influence graph is formalized in a model with a chosen semantics, depending on the amount of dynamical parameters that can be estimated, the available data and the expected result [Franke et al., 2010].

Different semantics have been developed to formalize biological systems, and can be classified in two main categories: quantitative and qualitative. The most frequently used semantics are introduced below.

- **Logical models** (qualitative)

If precise mechanisms for interactions are unknown and the data about the system's behavior is mostly qualitative, the influence graph can be directly used to build a logical model. This qualitative approach based on Boolean algebra or generalization thereof [Glass and Kauffman, 1973, Thomas, 1991] offers a flexible framework to delineate the main dynamical properties of complex biological regulatory networks. In this formalism, the activity levels of regulatory components are abstracted into discrete variables and their dynamics are represented by transitions between discrete states of the system (see [Chaouiya and Remy, 2013] for more detail). The time is implicitly represented through the discrete transitions. The logical framework has been used successfully to construct and analyze large networks [Albert and Thakar, 2014, Naldi et al., 2015, Le Novere, 2015].

- **Continuous models** (quantitative)

Quantitative models are essentially based on systems of ordinary differential equations (ODEs), representing the continuous evolution of each molecule concentration, with a continuous time. This formalism provides quantitative predictions, and dynamical systems analyses techniques such as phase portrait and bifurcation analysis can be used to study the possible dynamics (used for example in [Leloup and Goldbeter, 2003, Csikász-Nagy et al., 2006, Novák and Tyson, 2008]). Since kinetic parameters have to be calibrated with experimental data, this formalism suffers from the lack of reliable kinetic data for the regulatory mechanisms and the high levels of noise in experimental data. Analyzing a continuous model is based on deterministic numerical simulations, which offers a partial view of the system dynamics properties, restricted to a particular set of parameter values. Moreover, detailed simulations are computationally expensive and often result in time consuming analyses. In Figure 1.2, a simple reaction of repressed mRNA transcription is represented as a regulatory map, and modeled with logical and continuous semantics.

- **Stochastic models**

Attempts at modeling intrinsic and extrinsic noise have increased the interest in stochastic methods. Logical and continuous semantics can be extended to include stochastic rates, in order to take into account the high variation exhibited by some processes that involve a small number of molecules, such as transcription. For example, stochastic equations are a refinement from differential models that includes additional stochastic variables representing uncertainty in parameters, reaction rates, or network structure. When biological variability as well as uncertainty in measurements may significantly affect the interpretation of data, stochastic models relying on Markov processes and simulated with Continuous Time Markov Chains (CTMCs) are better suited, at the cost of a heavier computational effort [Wilkinson, 2009]. The compositionality offered by process algebra, originally developed for describing computer processes, is also a successful approach to model biological systems as concurrent systems [Regev et al., 2001].

In this dissertation we focus on non-stochastic logical and continuous modeling. While this choice impacts greatly on the dynamical results, theorems allow to draw conclusions from one semantic to another one. Although it overtakes the scope of this thesis, it is important to have in mind that links can be drawn between different modeling semantics, and that most analysis techniques can be adapted from one semantic to another.

Some formalisms support various semantics with extensions. For example, Petri nets are a graphical and mathematical formalism suitable for the modeling and the analysis of concurrent, asynchronous, distributed systems. They were initially used in systems biology as equivalent of

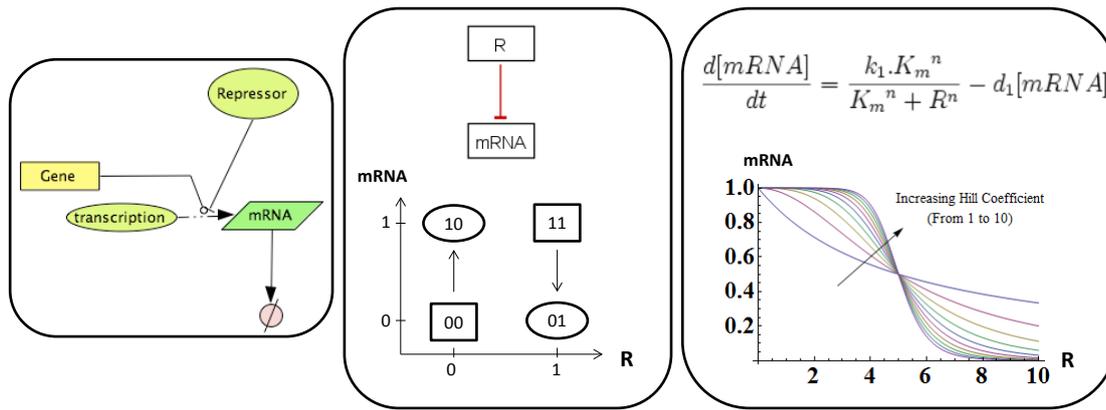


Figure 1.2: **Left:** CellDesigner-style regulatory map of a repressed mRNA transcription. **Middle:** Logical modeling of a repressed mRNA transcription with two Boolean components R (repressor) and mRNA. The dynamics on the bottom exhibits two stable states (round nodes): mRNA is active if R is inactive, and mRNA is inactive if R is active. **Right:** ODE modeling a repressed mRNA transcription with a Hill function. R is the repressor concentration, k_1 is the maximal expression rate when there is no repressor, K_m is the repression coefficient, equal to the concentration of repressor needed to repress by 50% the overall expression, and n is the Hill Coefficient. It controls the steepness of the switch between no-repression to full-repression. On the bottom, the level of transcribed mRNA versus R is displayed.

logical systems, based on the number of molecules for each species and simulated with multiset rewriting [Reddy et al., 1996, Chaouiya, 2007]. However, extensions of the formalism allow the consideration of stochastic or continuous models.

Based on the theory of abstract interpretation, links can be drawn between different semantics. In [Fages and Soliman, 2008], Galois connections have been shown to link in the logical formalism the syntactical, stochastic, discrete Petri Net and Boolean semantics. Moreover, for ample conditions the ODE semantics approximates the mean stochastic behavior [Gillespie, 1977]. These abstraction relationships are summarized in Figure 1.3.

Developing common formalisms for different semantics, and relationships between semantics is useful to draw links between models. Techniques of model reduction [Naldi et al., 2011, Naldi et al., 2012], modular modelling (see for example [Fauré et al., 2009]) or hybrid [Lincoln and Tiwari, 2004, Chiang et al., 2015] and multiscale modeling can then take advantage of existing models to broaden the analysis of already studied biological systems at multiple levels of organization (gene, protein/enzyme, cell, tissue, organ, organism). Population models can be approximated by stochastic models that use ODEs modelling higher order moments to estimate the mean and variance. Proper population models, based on PDEs and agent-based simulations (for example in [Billy et al., 2014]), usually capture less details on the molecular machinery of the cell, but account for cell-to-cell variability. Multiscale models attempt to describe both molecular interactions and population characteristics such as growth rate [El Cheikh et al., 2014].

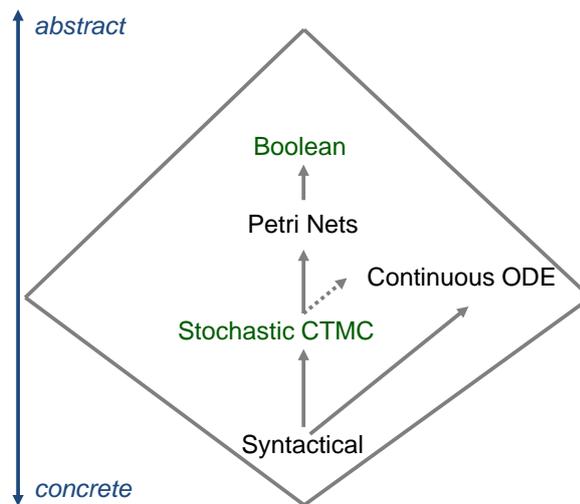


Figure 1.3: Hierarchy of semantics described in [Fages and Soliman, 2008]: Boolean semantics can be abstracted from logical semantics, which can be abstracted from stochastic semantics based on Continuous-Time Markov Chains. Continuous semantics based on ODEs abstracts the mean behavior of stochastic semantics in some conditions.

1.3 Model validation

Formal specifications of dynamical behaviors

A model is validated relatively to a considered behavior by checking the coherence of its dynamics with available data on this behavior. Different *in silico* experiments can be used to infer dynamic properties that will be most relevant to confront with the data, which can be quantitative or qualitative, incomplete or uncertain, reflect different conditions of perturbations of the system, and may need a phase of preprocessing.

Asymptotic properties can usually be determined with static analysis methods based on structural conditions rather than simulations. For example, steady-state analysis finds equilibrium states of the systems on topology grounds. On the other hand, formalizing properties on the trajectories of the system is more complex, because of the wide possibility of possible behaviors. Of crucial importance is the selection and correct specification of relevant properties. For example, comparing a simulation trajectory to an experimental trace with a general comparison method such as curve-fitting poses great risks of over-specification in case of imprecise or noisy data.

This problem is addressed by the formalization of complex properties with temporal logic, a formal system for representing and reasoning about propositions qualified by the orders of events in time. Properties expressed in temporal logic are automatically verified against a model by model checkers. In this dissertation, we explore the use of temporal logic to formalize complex yet imprecise dynamical properties of biological systems for different semantics, and to partly automate model building as a constraint satisfaction problem.

Model checking

Model checking is a computer science technique that aims at determining whether a dynamical system, formalized as a finite-state automaton, satisfies a set of properties, themselves formalized as logical specifications. It has been initially proposed in the 1980s to reason about very large hardware and software systems, typically to check for the absence of deadlock and critical state that can cause the system to crash. As it faces state explosion, this technique is often manually

intractable. This motivates the use of automated formal verification techniques that explore exhaustively and automatically finite models of programs so as to determine whether undesirable error states are accessible.

The system dynamics is formalized as a Kripke structure, which is an oriented graph representing the transitions between the different states of the system. Each path in the graph, starting from some initial states, represents a sequence of states separated by permissible transitions, i.e. a possible computation of the system. Model-checkers use specific algorithms to verify properties in the graph.

The size of the model is critical for the checking phase to be practical. Traditional model checking is based on Binary Decision Diagrams and is implemented in symbolic model checkers such as NuSMV [Cavada et al., 2002]. This approach has been complemented by various successful model checking algorithms based on SAT-solving, such as IC3 [Bradley, 2011]. In the frame of biological systems viewed as programs, it is natural to use model checking tools to verify the validity of biological properties in a system. In this respect, in mid 2000s, model checking started to be applied in the broad field of Systems Biology, mainly for the verification of qualitative systems dynamics. Successful applications include [Chabrier-Rivier et al., 2004, Bernot et al., 2004, Batt et al., 2005, Gong et al., 2011].

This approach makes a bridge between theoretical models and biological experiments, through the following identifications:

biological model = transition system,
biological property = temporal logic formula,
model validation = model checking,
model inference = constraint solving.

Initially developed for discrete systems, model checking methodologies have been improved, as well as their range of applicability, and the tools have been extended to deal with real-time and limited forms of hybrid systems. Temporal logic specifications are applied differently on logical models and continuous models, although the challenge is the same: automatically decide if the dynamics of the model satisfies a given temporal property. On the one hand, the simplicity of the logical framework allows the logical symbolic verification of qualitative temporal logic specifications for all possible dynamics of the model with model checking methods. On the other hand, extensions of temporal model checking techniques enable quantitative verifications with continuous semantics, such as robustness. But the verification relies on numerical simulation and is restricted to a given parameterization.

Formalizing dynamical behaviors with temporal logic

Different temporal logics exist, each with specific operators to reason about time. The most widely used are the Linear Time Logic (LTL) and the Computation Tree Logic (CTL), which have been developed successively and have become standards in the model checking community. All properties expressible in either logic can also be expressed in the superset CTL*.

Computation Tree Logic (CTL)

CTL is a branching-time logic that can reason about multiple time lines [Clarke et al., 1999] organized in a tree-like structure, and is thus adapted to non-deterministic automata derived from discrete models. Symbolic methods avoid enumerating each state of an automaton, which can be costly, by representing sets of states and transitions. A widely adopted method uses binary decision diagrams (BDDs). Another method called Bounded Model Checking (BMC) [Clarke et al., 2001] uses a propositional SAT solver rather than BDD manipulation techniques and looks for solutions of size smaller than a user-provided bound. Extensions of CTL exist, for example the Action Restricted Computation Tree Logic (ARCTL) [Monteiro and Chaouiya, 2012] is used to account for a distinct semantics of inputs and internal (regulated) components. It imposes an additional path restriction on a subset of inputs while letting others inputs to freely vary.

Linear Temporal Logic (LTL)

LTL is restricted to reasoning about one time line, and can thus be successfully used to characterize deterministic dynamics. Otherwise, it checks the satisfaction of a property successively on all possible sequences of states. Similarly to CTL, solving LTL specifications can be done with BDDs or SAT solvers in the BMC framework. Initially applied to discrete systems, LTL can be generalized to quantitative models in two ways: either by discretizing the different regimes of the dynamics in piece-wise linear or affine models [de Jong et al., 2004, Batt et al., 2010], or by relying on numerical simulations instead of symbolic solving and taking a first-order version of temporal logic (FO-LTL) with constraints on concentrations, as query language for the numerical traces [Fages and Rizk, 2008]. Such language can be used not only to extract information from numerical traces coming from either experimental data or model simulations, but also to specify the expected behaviors as constraints for model calibration and robustness measure [Rizk et al., 2009, Rizk et al., 2011, Donzé et al., 2013].

Other extensions of temporal logics have been developed for quantitative model checking, like the continuous interpretation of Signal Temporal Logic [Donzé and Maler, 2010], which has been successfully used in [Stoma et al., 2013] to build a model of TRAIL-induced apoptosis and revisit the classification of T-cells, or for oscillation constraints in [Banks et al., 2015]. However, STL is limited by the inability to distinguish signal values occurring in time points in which some local property is satisfied. In [Brim et al., 2014], an extension of STL logic denoted STL* based on a signal-value freeze quantification allows referencing a signal value in a time point. In this formalism, suited for nontrivial real-time processes, a property satisfaction problem is solved with an approximate monitoring procedure. The grammar uses bounded operators constrained by time intervals.

FO-LTL, in contrast, is suited for smooth signals supporting derivatives and uses unbounded operators, which allows the definition of complex specifications on a whole trace.

Thesis

The formalization of biological properties as a specification in temporal logic remains a delicate task and a bottleneck of this approach.

In this dissertation, we explore the use of both CTL and FO-LTL for constraint solving in systems biology. Although both have already been used in various studies, we investigate their application to analyze oscillatory systems, which necessitate to formalize more complex behaviors than previously done.

For example, in [Batt et al., 2005] CTL specifications have been used with model-checking methods to test conditions leading to a given state in a model, imposing restrictions on sequences of events along the path.

Moreover, the method based on FO-LTL and described in Chapter 3 has been used in cell signaling to elucidate the complex dynamics of GPCR signaling in [Heitzler et al., 2012]. However, the specifications defined in this study were mostly based on curve-fitting, a technique that requires precise quantitative and temporal data. In that case, some response curves in a model of GPCR signaling were available, and the failure to fit them was the key to revisit the structure of GPCR signaling interactions, and propose a different mechanism that has been verified experimentally. As another example, a temporal logic specification of timing constraints in a model of a genetic switch was successfully used in [Rizk et al., 2009] to compute global sensitivity indices and improve the design of a synthetic switch.

We investigate the definition of temporal logic specifications that are more complex than curve-fitting or simple reachability. Our work highlights the ability of this formal framework to specify a wide variety of dynamical behaviors, and to deal with imprecise data. For example, as will be detailed in Chapter 3, FO-LTL allows us to combine qualitative properties of oscillations and quantitative properties on the shapes of the traces such as distances between peaks or peak amplitudes. This is useful to capture the periods on both experimental and simulated traces, even when the traces are irregular and noisy.

Constraints for model synthesis: parameter inference

Calibrating dynamical models on experimental data is a central task in computational systems biology. Beyond property verification, the logical paradigm exposed earlier allows to develop methods of model synthesis as constraint solving problems. Given a regulatory network, the goal is to infer a parameterization (logical or continuous rules) that satisfies a list of properties specified as constraints.

When logical or numerical values for model parameters can be found to fit the data, the model can be used to make predictions, whereas the absence of any good fit may suggest to revisit the structure of the model and gain new insights in the biology of the system, see for instance [Stoma et al., 2013, Heitzler et al., 2012]. The validation of the model is thus a phase of intelligent assistance for the design of new experiments.

In particular, several optimization methods have been developed to infer kinetic parameters for continuous models by fitting timed quantitative data such as time series. They are increasingly used as technical innovations in experimental devices now produce precise temporal data with single-molecule measurements and single-cell time series.

The difficulty for model calibration is to identify the most relevant characteristics from the quantitative data and avoid over-fitting. Temporal logic provides a generic method to specify dynamical behaviors while combining expressiveness and flexibility.

Complexity

The general idea of model-checking a single finite trace has been well known for years, notably in the framework of Runtime Verification [Markey and Schnoebelen, 2003]. It usually relies on the classical bottom-up algorithm, which is bilinear [Roşu and Havelund, 2005]. This extends even to quantitative model checking like the continuous interpretation of Signal Temporal Logic [Donzé and Maler, 2010] since the combination of two booleans or two reals by min/max is cheap. However, when using the full power of First-Order Linear Time Logic (FO-LTL) to compute validity domains, the dependency of the complexity on the size of the trace is no longer linear but exponential in the number of variables [Fages and Rizk, 2008], reflecting the computational cost of combining complex domains. In this dissertation we address this issue with two approaches to improve the performance of FO-LTL constraint solving, and with it of the corresponding calibration methods: a trace simplification procedure prior to the constraint solving [Traynard et al., 2014] and the definition of dedicated solvers for frequently used properties [Fages and Traynard, 2014].

Modelling platforms

Two modelling softwares have been mainly used for the work presented in this thesis: GINsim for logical modelling and Biocham for continuous modelling.

- GINsim - <http://ginsim.org>

The logical model presented in Chapter 2 has been built and partly analyzed with GINsim. This software suite has been developed to facilitate the definition, the simulation, and the dynamical analysis of logical models [Naldi et al., 2009, Chaouiya et al., 2012]. In particular, this software supports STG construction for different updating assumptions. In addition, it provides a function to compress state transition graphs by regrouping the states into components leading to the same attractors or cyclic components, thereby easing the identification of dynamical attractors (stable states, simple or complex cycles) along with their basins of attractions [Bérenguier et al., 2013]. Finally, GINsim enables the definition and storage of different initial states and perturbations (e.g. gain- or loss-of-function mutants).

- Biocham - <http://lifeware.inria.fr/biocham> The logical paradigm for systems biology is at the heart of the development of the biochemical abstract machine (BIOCHAM) modeling

platform [Fages, 2005, Calzone et al., 2006]. Over the last decade, this software environment has been developed to model cell biology molecular reaction systems, reason about them at different levels of abstraction, and formalize biological behaviors in temporal logic with numerical constraints. Biocham allows us to analyze dynamical properties, infer non-measurable kinetic parameter values, and evaluate robustness.

Several methods for efficient solving of temporal specifications, in particular trace simplification [Traynard et al., 2014] and dedicated solvers [Fages and Traynard, 2014], have been developed as part of this work and implemented in Biocham.

The continuous model presented in Chapter 5 has been built using the BIOCHAM modeling software for

1. importing and exporting models in SBML, and modeling the molecular interactions of the coupling of the models,
2. specifying experimentally observed behaviors in quantitative temporal logic using relations for periods and phases [Fages and Traynard, 2014, Traynard et al., 2014],
3. searching parameter values [Rizk et al., 2011] and measuring robustness and parameter sensitivity indices [Rizk et al., 2009] with respect to the temporal logic specification of the dynamical behavior,
4. making biological hypotheses on the coupling between the cell cycle and the circadian clock.

1.4 Outline

- **Chapter 2** provides a formal introduction to CTL and an illustration of its use for analyzing the structural properties of a logical model of the cell cycle. An updating of its regulatory network and logical rules is proposed based on this analysis.
- **Chapter 3** introduces FO-LTL logic and recalls some results used in this thesis. Then common dynamical properties, in particular for oscillatory systems, are formalized as temporal logic specifications and illustrated on a toy oscillator model.
- **Chapter 4** extends FO-LTL-based tools with trace simplification and built-in functions, in order to improve the solving efficiency of common oscillatory constraints. An illustration on oscillatory data analysis is also provided.
- **Chapter 5** puts these tools to full application to investigate coupling mechanisms between the mammalian cell cycle and circadian clock. Temporal logic is used to specify period and phase constraints extracted from experimental studies and synthesize satisfactory models.
- **Chapter 6** concludes, discusses limits and envisions future directions.

2. Temporal constraints in the logical framework

2.1 Cell cycle

The cell cycle involves a succession of phases governing genome replication (S phase) and cell division (mitosis or M phase), separated by regulated irreversible transitions (checkpoints), as schematized in Figure 2.1. Each phase is closely regulated, and the resulting cycle exhibits regular oscillations in some conditions, for example in developing tissues, where cells divide synchronously with a fixed period. In other conditions, cells can exit the cycle and remain in a quiescent state.

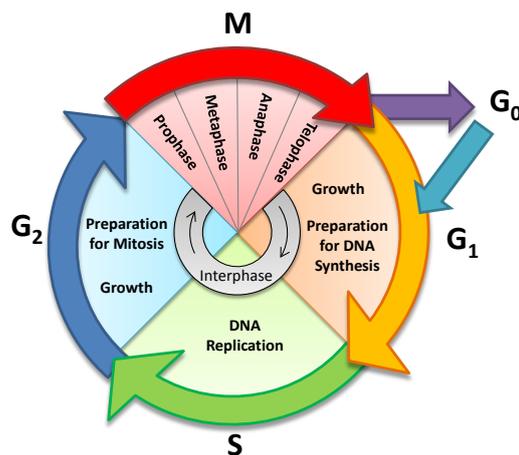


Figure 2.1: Schema of the phases composing the cell cycle.

Widely conserved among eukaryotes, the underlying core network has been modelled using differential equations for several species (Yeast, *Xenopus*, mammals), leading to novel insights into its organization and dynamical properties (see e.g. [Novák and Tyson, 2004, Csikász-Nagy et al., 2006, Gérard and Goldbeter, 2009, Ferrell et al., 2011, Tyson and Novák, 2015] and references

therein). However, extension and analysis of such differential models becomes really difficult as the number of experimentally identified components and interactions increases. This led to the consideration of simpler, qualitative but nevertheless rigorous formal approaches, using discrete formalisms (see e.g. [Li et al., 2004, Davidich and Bornholdt, 2008, Irons, 2009, Fauré et al., 2009, Mombach et al., 2014]), that have proven to be able to correctly represent the succession of phases and transitions in the cell cycle.

Notably, [Fauré et al., 2006] defined the first Boolean model for the core network driving the mammalian cell cycle and demonstrated that the logical framework enables the reproduction of important properties of the highly complex and coordinated system regulating the maintenance and preservation of distinct phases in the cell cycle. This model accounts for the existence of a quiescent stable state, as well as for an asymptotic behavior characterized by the periodic activities of the four main cyclins, that drive the cell cycle through key transitions by enabling the phosphorylation of a number of substrates by their catalytic partners, the cyclin-dependent kinases (cdks). However, detailed conclusions on the order of activity switching (off or on) and its match to the available data are restricted to the kinetic approximation induced by the assumption that all possible changes in the component activities occur synchronously. Since asynchronous transitions cause a combinatorial explosion, Fauré et al. propose a compromise by defining priority classes for the transitions between states. This approach produces a reduced non-deterministic transition graph and results in some unrealistic pathways. [Fauré et al., 2006] further considered a list of documented perturbations to validate their model. The simulations of several perturbations that did not match experimental observations already pointed toward some limitations of the model.

Based on this previous work, we now focus on the fully asynchronous updating strategy, which allows the consideration of alternative dynamics in the absence of kinetic data. The resulting complex dynamical trajectories can be analyzed with formal analysis tools. More precisely, the existence of specific sequences of states or set of states in the asynchronous trajectories is verified with model-checking techniques. These sequences formalize dynamical properties such as irreversible transitions between phases of the cell cycle. The novel application of model-checking techniques is combined with two validation methods applied in the previous study - the type of asymptotic behavior and the impact of perturbations - to characterize the dynamics of the logical model and its consistency with biological observations. Combining these methods thereby achieves a better understanding of the dynamics of the model and highlights its limitations.

Then, the model is refined in a data-driven approach through a careful review of the literature to identify relevant novel information. It leads to the introduction of new interactions and components. Although we only comment the dynamical properties directly affected by each modification, we also check at each step that other properties in Table A.1 (effect of perturbations) and Tables A.2 and A.3 (sequential aspects) are still satisfied.

The main features of the logical framework are summarized in Section 2.2. The use of model checking is then described more precisely in Section 2.3. Section 2.4 details the composition of the network driving the mammalian cell cycle, and the results of the analysis of Fauré's model are presented in Section 2.4. The rest of Section 2.4 is then devoted to assessing the improvement brought by refining and extending the model. Finally, Section 2.5 explores a stochastic extension of the model.

2.2 Logical modeling framework

Logical modeling aims at representing regulatory or signaling networks as molecular automata with symbolic logic. Gene expression changes are encoded as transitions between logical states of the automata. This formalism was first introduced by Mitoyosi Sugita [Sugita, 1963], and then detailed with a synchronous updating by Stuart Kauffman [Kauffman, 1969] and an asynchronous updating by René Thomas [Thomas, 1973].

A logical model is defined by a regulatory graph, where each node represents a regulatory component, and is associated with a discrete variable abstracting its level of activity (0, 1, and

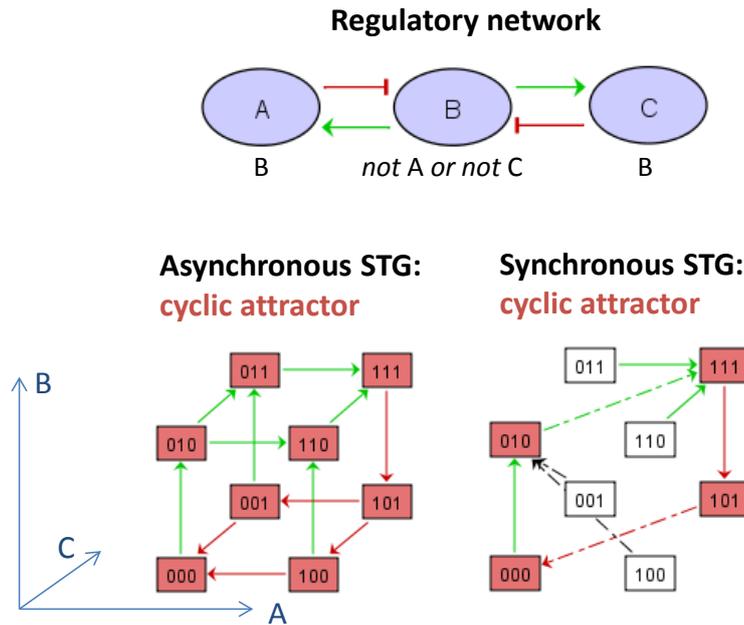


Figure 2.2: Top: a toy regulatory network with three nodes A, B and C and the corresponding logical rule below each node. Green arrows depict positive interactions (activation) while red arrows represent negative interactions (inhibition). Bottom: the corresponding state transition graphs (STG) with asynchronous transitions (bottom left) or synchronous transitions (bottom right).

further integers when justified). Each arc represents a regulatory interaction between the source and target nodes, and is labelled by a threshold and a sign (positive or negative). The dynamical behavior of each node is then defined by logical functions or rules, which associate a target value for each combination of regulators. The dynamics of the system can be represented in terms of a state transition graph (STG), where the nodes denote the states of the system (i.e. vectors giving the levels of activity of all the variables), and the arcs represent state transitions (i.e. changes in the value of one or several variables, according to the corresponding logical functions) (for more details, see [Thomas and D’Ari, 1990, Thomas et al., 1995, Chaouiya et al., 2003]).

When concurrent variable changes are enabled at a given state, the resulting state transitions depend on the chosen updating assumption. With a fully synchronous strategy, all concurrent variables are updated through a unique transition. This assumption leads to relatively simple transition graphs and deterministic dynamics. However, this assumption notably causes spurious cyclic attractors. On the other hand, the asynchronous updating assumption considers the updating of each variable separately. The resulting dynamics is more difficult to evaluate. For instance, for the very simple toy model represented in Figure 2.2, synchronous transitions result in a cyclic attractor with four states, while the asynchronous attractor involves twice more states with multiple trajectories. Although methods have been developed to infer the asynchronous attractors [Garg et al., 2008], refined analysis of the dynamics inside an attractor remain hard. Nonetheless, the asynchronous updating strategy is seen as more realistic because it allows the consideration of alternative dynamics in the absence of kinetic data.

In this chapter, we focus on the asynchronous updating strategy and we rely on model-checking techniques to analyze the resulting complex transition graph.

2.3 Model analysis

We use three types of properties to characterize the dynamics of a logical model and its consistency with biological observations: attractors, sequences of states in the simulated trajectories, and impacts of perturbations.

Attractors, perturbations, model checking

The attractors of a model, defined as the terminal strongly connected components of the STG, are the final states of all possible trajectories and characterize the asymptotic behavior of the model. An attractor composed of several states linked in cyclic trajectories is a complex attractor, while an attractor composed of a single state is a stable state. The existence of a cyclic attractor is a first requirement for the logical model to reproduce the oscillatory expressions of proteins in the cell cycle, and its composition point out which components of the model take part in periodic activities.

Moreover, reproducing the results of experimental perturbations is a method of choice to validate a dynamical model. Comparing the asymptotic behavior of a model with or without perturbation provides interesting insights into the structural properties of the system. If the resulting attractors clearly contradict the expected behavior, it provides a rejection of the model.

Finally, the existence of specific sequences of states or set of states can be checked by model checking among the possible trajectories simulated with a given updating strategy (from deterministic trajectories with the synchronous updating to branching trajectories with the asynchronous updating). This technique is used originally in this study to look for sophisticated sequences characterizing the transitions between cell cycle phases, rather than mere reachability properties. We describe more precisely its use and application in the following paragraphs.

Model checking with CTL

Model checking allows the formal verification of specific dynamical properties and thereby the validation or refutation of a model [Clarke et al., 1999]. It can also be used to formulate predictions or hypotheses on the studied system.

As introduced in Chapter 1, behavioral properties are expressed in the logical framework in Computation Tree Logic (CTL), described in more details in the next subsection.

CTL specifications are then evaluated on discrete models with a model-checker. In particular we use NuSMV, a symbolic model-checker based on Binary Decision Diagrams that provides a description language to specify generic finite state machines [Cimatti et al., 2002]. It has been widely used to check properties on discrete regulatory networks.

Model checking with CTL has already been applied to address biological problems, although usually focusing on simple properties. Symbolic model checking for systems biology using CTL formulae as introduced by [Chabrier and Fages, 2003] to assess state reachability. Later, [Batt et al., 2005] tested conditions leading to a given state, imposing restrictions on sequences of events along paths. In [Batt et al., 2010], model checking was used to solve a parameter search problem for piecewise-affine differential equation models of regulatory networks in order to reproduce observed expression profiles.

Approaches using extensions of standard CTL have been explored, such as Action Restricted CTL (ARCTL), to discriminate between variants of a logical model of T-helper cell differentiation, or to investigate reachability properties between logical expression patterns as input conditions representing T-helper cell types [Monteiro et al., 2014, Abou-Jaoudé et al., 2015]. Another approach implemented in ANTELOPE (Analysis of Networks through TEmporal-LOGic sPEcifications) [Arellano et al., 2011] supports Hybrid CTL, an extension of standard CTL with a special binder temporal operator, capable of selecting partially characterized states.

CTL

The Computation Tree Logic CTL [Clarke et al., 1999] is a branching-time logic that allows reasoning about an infinite tree of state transitions.

A CTL formula is built upon atomic propositions potentially combined with the usual operators from propositional logic, such as negation (\neg), conjunction (\wedge), disjunction (\vee), and implication (\rightarrow).

In addition, it uses operators about branches (non-deterministic choices) and time (state transitions).

CTL temporal operators are:

- **F** (finally): $\mathbf{F} \phi$ means that ϕ is eventually true (at some time point in the future).
- **G** (globally): $\mathbf{G} \phi$ means that ϕ is always true (at all time points in the future).
- **X** (next): $\mathbf{X} \phi$ means that ϕ is true at the next transition.
- **U** (until): $\phi \mathbf{U} \psi$ means that ϕ is true until ψ becomes true.
- **W** (weak until): $\phi \mathbf{W} \psi$ means that ϕ is true until ψ becomes true, or ϕ is always true.

Two path quantifiers allow to handle non-determinism, and each temporal operator must be paired with a path quantifier:

- **A** (for all): $\mathbf{A} \phi$ means that ϕ is true on all branches
- **E** (exists): $\mathbf{E} \phi$ means that it is true on at least one branch.

These operators enjoy some simple duality properties:

- $\neg(\mathbf{EF}(\phi)) = \mathbf{AG}(\neg\phi)$,
- $\neg(\mathbf{E}\phi\mathbf{U}\psi) = \mathbf{A}(\neg\psi\mathbf{W}\neg\phi)$,
- $\neg(\mathbf{E}\phi\mathbf{W}\psi) = \mathbf{A}(\neg\psi\mathbf{U}\neg\phi)$.

In the following we provide examples of statements containing CTL formulae that can be used with the model checker NuSMV to assess the belonging of set of states to attractors or basins of attractions. In the following statements, SPEC and INIT are the SMV statements defining, respectively, a CTL specification to check and the initial states where the specification will be tested. The terms *Attractor*, *Attractors*, *PointInAttractor*, *initStates* and *BasinOfAttraction* all represent user-defined set of states or single states.

Attractors

- **Existence of an attractor**
INIT *Attractor*; SPEC $\mathbf{AG} \text{ } \mathbf{Attractor}$; - equivalent to $\mathbf{AX} \text{ } \mathbf{Attractor}$
If the answer to this specification is True, *Attractor* contains one or several attractors (stable state or cycle).
- **Finding all attractors**
SPEC $\mathbf{AG} \mathbf{EF}(\mathbf{Attractors} \wedge \mathbf{AG} \text{ } \mathbf{Attractors})$;
If the answer to this specification is True, *Attractors* contains all attractors. Note that the same formula with $\mathbf{AG} \mathbf{AF}$ instead of $\mathbf{AG} \mathbf{EF}$ would fail in the presence of transient cycles in the transition graph, because infinite paths that never reach the attractors could exist inside transient cycles.

Basins of attraction

- **Exact basin of attraction**

SPEC ($initStates \rightarrow \mathbf{AG} \mathbf{EF} PointInAttractor$) \wedge ($\neg initStates \rightarrow \mathbf{AG} \neg PointInAttractor$);

If the answer to this specification is True, the attractor is the only one reachable from *initStates* and is not reachable from any other state: *initStates* is the exact basin of attraction of the attractor containing *PointInAttractor*. Otherwise, *initStates* is not the exact basin of attraction of the attractor containing *PointInAttractor*: there is either a missing state or too many states.

- **Transient cycle in a basin of attraction**

INIT *BasinOfAttraction*; SPEC $\mathbf{AG} \mathbf{EF} Attractor \wedge \neg \mathbf{AF} Attractor$;

If the answer to this specification is True, there is a transient cycle in the basin of attraction.

Sequences of states

So far most uses of model checking for logical modeling correspond to reachability properties verifying the existence of a path between sets of states, with possible restrictions on the paths. Here, we use the model-checker NuSMV to verify the existence or the absence of specific state transition paths corresponding to sophisticated dynamical properties. We use the following generic CTL temporal logical formula to verify the existence of a trajectory complying a sequence of properties $S_1, S_2, S_3, \dots, S_{n-1}, S_n$, each denoting a set of states defined by constraints on some of the model components:

$$\neg \mathbf{E}[(S_1) \mathbf{U}(S_2 \wedge \mathbf{E}[(S_2) \mathbf{U}(S_3 \wedge \dots \mathbf{E}[(S_{n-1}) \mathbf{U}(S_n)])))]$$

The negation is used here for two reasons. First, since a CTL temporal logic property ϕ holds if all initial states satisfy ϕ , testing whether $\neg\phi$ holds verifies the absence of the specified sequence. Second, a contradiction of $\neg\phi$ returns an example of transition path matching the prescribed sequence.

Importantly, we do not expect all the sequential activities observed experimentally to be satisfied by our model. Indeed in the logical framework, the asynchronous assumption relies on a branching definition of time, potentially resulting in different dynamics compatible with the same model. On the one hand, experimentally-observed sequences of states are expected to exist in the asynchronous trajectories. Otherwise the model can be safely rejected. On the other hand, *safety properties* (verifying that incorrect sequences do not exist) do not invalidate the model if they are unsatisfied. Indeed, incorrect sequences of states that can be exhibited in the asynchronous trajectories do not necessarily indicate a flaw of the model, but might rather point to specific kinetic constraints, not taken into account by the asynchronous dynamics, that could oppose these spurious sequential activities.

Nonetheless, satisfied safety properties represent interesting features of the model. Indeed, a property satisfied in the asynchronous transition graph is intrinsically rooted in the structure of the model, and as such could always be exhibited in some conditions. The cell cycle represents a particularly interesting case, because the articulation of distinct phases is a highly complex and coordinated process. It is regulated by protein synthesis, phosphorylation (through the activity of cyclin-dependent kinases or CDKs) and protein degradation processes (involving ubiquitin ligases). We expect that characteristic dynamical properties, such as checkpoints and irreversible transitions, are robustly encoded in the structure of the corresponding regulatory network. Analysis of a logical model should reveal such properties, despite the absence of detailed kinetic information.

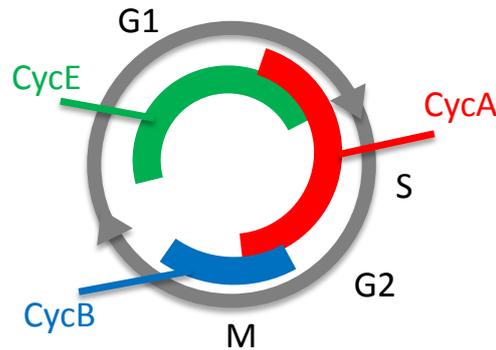


Figure 2.3: Schema displaying the sequential activities of the three main cyclins in the mammalian cell cycle: cyclin E, cyclin A and cyclin B.

2.4 Application: studying the structural properties of the cell cycle

Logical modeling of the mammalian cell cycle

The present study is based on the first Boolean model of the mammalian cell cycle [Fauré et al., 2006], which demonstrated that the logical framework enables the reproduction of important properties of the cell cycle. In order to update and extend this model, we further rely on a recent differential model emphasizing the role of Skp2 [Gérard and Goldbeter, 2009].

[Fauré et al., 2006] defined a Boolean model for the core network driving the entry of mammalian cells into cell cycle, based on the differential model proposed by [Novák and Tyson, 2004]. For proper logical rules, this model accounts for the existence of a quiescent stable state, as well as for a cyclic attractor characterized by the periodic activities of the four main cyclins, which drive the cell cycle through key transitions by enabling the phosphorylation of a number of substrates by their catalytic partners, the cyclin-dependent kinases (cdks). Cyclin D (called CycD and corresponding to an input in the model) is the main target of the growth factors that push a cell out of its quiescent state to enter the cell cycle. Cyclin E (CycE) regulates the transition between the G1 and S phases. Cyclin A (CycA) controls S phase and its progression into G2 phase. Finally, cyclin B (CycB) is in charge of the transition from G2 phase to the mitosis, and thus triggers the division of the cell before its return to the quiescent state or to G1 phase. The successive activities of these three cyclins in the cell cycle are schematized on Figure 2.3.

Fauré's model further includes the three main inhibitors of the cell cycle: the retinoblastoma protein Rb, the Cdk inhibitor p27/Kip1 (p27 in the sequel) and the proteasome complex represented by its two co-activators Cdh1 and Cdc20. However, this seminal model considered CycD sufficient to completely inhibit p27 and Rb. Hence, these inhibitors were kept inactive along the whole cyclic attractor, whereas, according to experimental data, these factors should be active in G1 phase, and inactive in the rest of the cell cycle ([Rivard et al., 1996], Nelson et al. 1997).

Finally, Fauré's model accounts for the role of the E2 ubiquitin conjugating enzyme UbcH10, which participates in Cdh1 dependent degradation of cyclin A. This extension of the original differential model explains how the auto-ubiquitination of UbcH10 likely prevents cyclin A from degradation by the APC in G1 phase. The logical rules governing the dynamics of the model species are listed in Table 2.2.

Reproducing the results of experimental perturbations is a method of choice to validate a dynamical model. Comparing the asymptotical behavior of a model with or without a perturbation provides interesting insights into the properties of the system. [Fauré et al., 2006] considered a

Node	Level	Initial rules
Rb	1	!CycB & !CycD & (p27 (!p27 & !CycA & !CycE))
E2F	1	!Rb & !CycB & (p27 (!p27 & !CycA))
CycE	1	E2F & !Rb
CycA	1	(E2F CycA) & (!UbcH10 !Cdh1) & !Rb & !Cdc20
CycB	1	!Cdh1 & !Cdc20
p27	1	!CycB & !CycD & ((p27 & (!CycA !CycE)) (!p27 & !CycA & !CycE))
Cdc20	1	CycB
Cdh1	1	(!CycB & (!CycA p27)) Cdc20
UbcH10	1	!Cdh1 (Cdh1 & UbcH10 & (CycA CycB Cdc20))

Table 2.1: Logical rules governing transitions in the initial model of the mammalian cell cycle [Fauré et al., 2006].

list of documented perturbations to validate their model. However, the simulations of several perturbations did not match experimental observations. In particular, the effect of a loss-of-function of cyclin E does not seem to be adequately predicted.

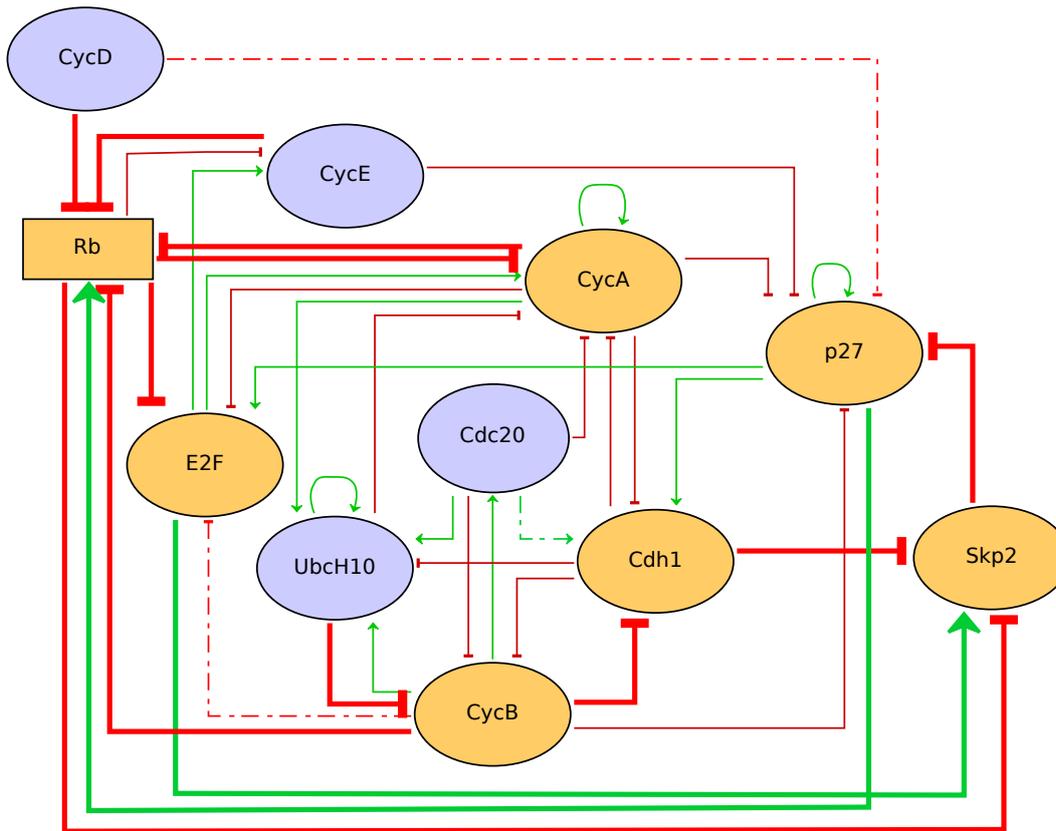


Figure 2.4: Updated model of the mammalian cell cycle. Red arrows depict negative interactions (inhibitions), green arrows are positive interactions (activations). Nodes are either Boolean (ellipses) or multilevel (rectangle). Logical rules of yellow nodes have been modified compared to the initial model, and thick arrows correspond to modified regulations. Dashed arrows correspond to interactions that have been removed from the model.

Results

We have investigated the asynchronous dynamics of Fauré’s model using model checking and perturbation analysis, and we systematically assessed the results to refine and extend this model so that it better matches reported data. In this respect, we first carefully reviewed the literature to identify relevant novel information.

The potential roles of different phosphorylation states of the protein Rb (Lundberg & Weinber, 1998) was considered through the use of a ternary node (i.e. taking the value 0, 1 and 2). We further assessed an extension of the model considering the role of Skp2, which links three key inhibitors of the cell cycle and constitutes an additional pathway by which Rb can arrest the progression of the cell cycle [Binné et al., 2007, Liu et al., 2008]. The regulatory graph of the updated model is depicted on Figure 2.4, while the corresponding logical rules are listed in Table 2.2.

Updating of regulatory rules

The cdk inhibitor p27 plays a critical role in several phases of the cell cycle and in the regulation of the quiescent state. It inhibits the activities of cyclin E/cdk2 and cyclin A/cdk2. This inhibitory activity is modeled by opposite regulations on the targets of CycE and CycA. Moreover, p27 and cyclin D also bind in a complex, but this complex retains the activity of cyclin D. Since the cyclins are in competition for complexation with p27, the initial model considered a direct inhibition of

Node	Level	Logical rule
Rb	1	(CycD & !CycE & !CycA & !CycB) (!CycD & p27 & CycE & CycA & CycB) (CycD & (CycE CycA CycB) & p27)
	2	(!CycD & !CycE & (!CycA & !CycB) p27)) (!CycD & p27 & CycE & (!CycA !CycB))
E2F	1	(!CycA p27) & !Rb
CycE	1	E2F & !Rb
CycA	1	(E2F & !Rb CycA) & !Cdc20 & (!UbcH10 !Cdh1)
CycB	1	(!UbcH10 !Cdc20) & !Cdh1
p27	1	(!CycB & (!CycA p27) & !CycE) !Skp2
Cdc20	1	CycB
Cdh1	1	(!CycA p27) & !CycB
UbcH10	1	!Cdh1 (Cdh1 & UbcH10 & (CycA Cdc20 CycB))
Skp2	1	!Cdh1 (!Rb & E2F)

Table 2.2: Logical rules governing transitions in the updated model of the mammalian cell cycle.

p27 by cyclin D to reflect the sequestration of the inhibitor by the cyclin D during the cell cycle. This causes p27 to be completely inactive in presence of the input cyclin D, while it is released and active in absence of the input (i.e. in the quiescent state). This approximation overlooks the role of p27 in the transition from G1 to S. Indeed, the complete activation of cyclin E is a progressive process, slowed down by both Rb and p27, which are both negative regulators of cyclin E: Rb binds to the transcription factor E2F, thereby inhibiting its ability to activate the synthesis of cyclin E, whereas p27 directly binds to cyclin E/Cdk2 and thereby blocks its activity. Rb and p27 are both phosphorylated by cyclin E, inducing inactivity of Rb and proteasome-dependent degradation of p27. These factors are thus involved in a positive circuit enabling the full activation of the kinase and ultimately entry into S phase [Kotoshibai et al., 2005].

In order to account for this mechanism, we removed the inhibition of p27 by CycD. Although it is clear that cyclin D plays a role in p27 phosphorylation, the observation that the activity of p27 varies in the cell cycle while the level of cyclin D stays consistently high suggests that this role is weak relatively to those of cyclin A, cyclin E and cyclin B [Rivard et al., 1996]. Alternatively, a ternary node could have been associated with p27 to distinguish two activation levels, in presence versus absence of cyclin D. We have further modified the rule for p27 (see Table 1) to account for the inhibitory effect of CycE [Montagnoli et al., 1999]. Simulation of the model with this new rule results in a correct asynchronous attractor with varying p27 activity (in the presence of CycD).

In order to verify the correct role of p27 in the cycle, we check for the existence of the three state transition sequences defined in Table 2.3 and formalized with CTL formulae. For example, the sequence of states [001, 101, 100, 110, 010] taken by the vector (CycE, CycA, p27) is translated into the CTL formula:

$$\begin{aligned} & \neg E[(CycE=0 \wedge CycA=0 \wedge p27=1) \cup (CycE=1 \wedge CycA=0 \wedge p27=1) \\ & \wedge E[(CycE=1 \wedge CycA=0 \wedge p27=1) \cup (CycE=1 \wedge CycA=0 \wedge p27=0) \\ & \wedge E[(CycE=1 \wedge CycA=0 \wedge p27=0) \cup (CycE=1 \wedge CycA=1 \wedge p27=0) \\ & \wedge E[(CycE=1 \wedge CycA=1 \wedge p27=0) \cup (CycE=0 \wedge CycA=1 \wedge p27=0)]]]] \end{aligned}$$

This first sequence represents the expected sequence for the G1/S transition, while the other two correspond to incorrect situations where p27 is either degraded before the activation of its inhibitor CycE, or stays active after the activation of CycA. We further impose the constraint CycB=0 & Cdh1=1, characterizing G0, at the initial states.

The requirement of cyclin E for cell cycle viability might depend on the cell context. Cyclin E is known to participate in the phosphorylation of Rb together with cyclin D [Weinberg, 1995]. However, observations on cyclin E-deficient cells highlight a crucial role of cyclin E in some situations. [Geng et al., 2003] reported that cyclin E-deficient cells can maintain active proliferation but are unable to reenter the cell cycle from the quiescent G0 state. [Ohtsubo et al., 1995] further

Phenotype	The phosphorylation and subsequent degradation of p27 by cyclin E is necessary to drive the feedforward loop that ensures that cells advance to S-phase irreversibly.								
	CycE	CycA	p27	CycE	CycA	p27	CycE	CycA	p27
Sequence	0	0	1	0	0	1	0	0	1
	1	0	1	0	0	0	1	0	1
	1	0	0	1	0	0	1	1	1
	1	1	0	1	1	0	1	1	0
	0	1	0	0	1	0	0	1	0
Expected result	True			False			False		
Updated model	True			False			False		

Table 2.3: Sequence properties: correct or alternative properties for the G1/S transition

report that inhibition of cyclin E blocks the entry in S phase in human cells and arrests the cell cycle.

For CycE knock-out, the simulation of the original model did not result into cell cycle arrest. In contrast, the simulation of our modified model results in an arrest of the cell cycle in phase S, where CycA is inactivated by p27. This simulation emphasizes the role of the Rb-CycE-p27 circuit in driving the cell cycle through the S phase.

A similar perturbation was introduced in a recent study where cancer cells are treated with a specific cyclinE/cyclinA-cdk2 inhibitor, resulting in a cell cycle arrest with an increased accumulation of p27 and hypophosphorylation of Rb [Dai et al., 2013]. The simulation of a double perturbation of CycE and CycA for the revised model is consistent with the reported results, while the same perturbation in the initial model results in a cell cycle arrest but with phosphorylated Rb ($Rb=0$) and no p27 accumulation ($p27=0$).

Careful reconsideration of each regulation rule further led us to remove the inhibition of E2F by CycB. Although E2F has been shown to be inhibited by cyclin A through direct binding, this is not the case for cyclin B [Krek, 1994], contradicting the inclusion of this interaction in previous models [Novák and Tyson, 2004]. Interestingly, the delay between the degradation of cyclin B after the mitosis and the activation of cyclin E during G1-phase was ensured by this regulation. In the updated model, this dynamical property is no longer satisfied, due to the activation of E2F before the degradation of cyclin B as seen in the example sequence provided by the model checker (cf. sequence 16 in Table A.3). As this hypothetical mechanism reveals to be unsupported by experiments, it leads us to question the existence of another mechanism or a kinetic cause.

We further modified the regulatory rule defining an activation of Cdh1 by Cdc20. These two substrate adaptor proteins of the anaphase-promoting complex (APC) are activated in turn during the G2 and M phases to activate the degradation of mitotic cyclins A and B. Since Cdc20 participates in degrading CycA, which inhibits Cdh1 by phosphorylation, there is an indirect activation of Cdc20 by Cdh1. As Cdh1 has a broader spectrum than Cdc20, it completes CycA and CycB inactivation, and inactivates Cdc20 [Meyer and Rape, 2011]. We have thus verified that the direct activation of Cdh1 by Cdc20 could be eliminated without impacting on relevant model properties (cf. Tables 2.3 to 2.6 and A.1 to A.3).

Multiple forms and roles of Rb

The protein Rb is a major cell cycle inhibitor and tumor suppressor. It is regulated by numerous stimuli that are channeled through Cdk regulation of Rb phosphorylation. The unphosphorylated protein binds to the transcription factor E2F, thereby acting as a growth suppressor and preventing progression through the cell cycle. When phosphorylated, Rb releases E2F, which activates the synthesis of CycE and CycA. Recent findings show that the phosphorylation of Rb is a progressive process [Henley and Dick, 2012]. The phosphorylation of Rb begins in early G1-phase with cyclin D/cdk4 or 6 [Narasimha et al., 2014]. This allows some E2Fs to be released and initiates the

transcription and subsequent production of cyclin E. The cyclin E/cdk2 kinase activity is able to hyper-phosphorylate Rb. As the majority of phosphorylation sites on Rb need to be modified to abrogate E2F binding, it defines differential activities for the different phosphorylation levels of Rb and a necessary role for cyclin E to drive the cell cycle into S-phase. Rb is maintained hyper-phosphorylated by cyclin A/cdk2 and cyclin B/cdk1 in S, G2, and M phases.

Refining the description of Rb with a multivalued node was already suggested by [Fauré et al., 2006]. We thus introduced a ternary node for Rb. The levels 0, 1 and 2 represent the hyper-phosphorylated, partially-phosphorylated and unphosphorylated states respectively. The logical rules for each level are defined in Table 2.2. The cyclins phosphorylate and inhibit Rb. In absence of cyclins, Rb is unphosphorylated (level 2). CycD alone initiates the partial phosphorylation at the beginning of the cell cycle, and thus leads Rb down to the level 1. It requires the assistance of another cyclin to complete the phosphorylation, leading Rb down to the level 0 in absence of p27. CycA and CycB have symmetrical roles and both of them keep Rb hyperphosphorylated during S, G2 and M phases. p27 binds to the cyclin-cdk complexes and inhibits the kinase activities, playing the role of an indirect activator of Rb. However, although it binds to CycD/Cdk4, the activity of this complex is not impaired. We take into account the saturation of p27 by the different cyclins through additive and saturation effects.

Interestingly, this model extension provides a mechanism explaining the sequential synthesis of cyclin E and cyclin A. The synthesis of both cyclins is activated by E2F, but their expressions peak in G1 phase and in S phase respectively [Lees et al., 1992, Wong et al., 2014]. We propose to model this difference of activation by different effects of the phosphorylated states of Rb. The complexation of Rb and E2F is modeled with a direct inhibition of E2F by Rb but also a negative regulation of Rb on the E2F targets CycE and CycA (as in [Fauré et al., 2006]). Setting different thresholds of activity for these interactions, namely threshold 2 for E2F and CycE, and threshold 1 for CycA, ensures that the synthesis of CycA can only be activated once Rb is hyperphosphorylated and E2F is completely released.

Phenotype	Cyclin A expression is delayed relatively to E2F and cyclin E			
	CycE	CycA	CycE	CycA
Sequence	0	0	0	0
	1	0	0	1
	1	1	1	1
Expected result	True		False	
Initial model	True		True	
Updated model	True		False	

Table 2.4: Sequence properties: alternative dynamics for the sequential activation of CycE and CycA.

The delayed synthesis of CycA relatively to the synthesis of CycE can be verified by model checking. A sequence with the reverse order is checked in Table 2.4, and the result is compared with the initial model. Initially both sequences of the table existed in the asynchronous STG, showing the lack of preferential order of activation between CycE and CycA. After the modification of the model described above, the incorrect sequence with CycA activated before CycE no longer exists in the asynchronous STG. Hence, our model provides an explanation for the observed sequential activation of cyclin E and cyclin A in the cell cycle, which relies on a robust mechanism encoded in the logical structure of the model. This mechanism is based on the positive loop that drives the complete phosphorylation of Rb and the progression into S phase.

Role of UbcH10 in mitosis

The anaphase-promoting complex (APC) coordinates mitosis and G1 by sequentially promoting the degradation of key cell-cycle regulators. APC is represented in the model by the two

Phenotype	UbcH10 is required for mitotic cyclin destruction.			
	CycA	CycB	Cdc20	UbcH10
Sequence	1	0	0	0
	1	1	0	0
	1	1	1	0
	0	1	1	0
	0	0	1	0
	0	0	0	0
Expected result	False			
Initial model	True			
Updated model	False			

Table 2.5: Sequence property showing the role of UbcH10 in mitosis

co-activators Cdh1 and Cdc20. The degradation of several targets of either Cdh1 or Cdc20 is assisted by the ubiquitin-conjugating enzyme 2C (UbcH10, also called UBE2C), a component of the ubiquitin proteasome system.

While the initial model encoded the fact that UbcH10 is necessary for Cdh1-dependent degradation of Cyclin A [Rape and Kirschner, 2004], UbcH10 might also be required for the destruction of mitotic cyclins and other mitosis-related substrates, including cyclin B [Rape et al., 2006]. We encoded this putative mechanism by updating the logical rule associated with the node CycB (cf. Table 2.2). Adding the intervention of UbcH10 for the degradation of CycB enables a consistent simulation of UbcH10 KO. Indeed, using model checking, one can verify that there is no asynchronous sequence going through G2 and M phase (driven by cyclin A and cyclin B) in absence of UbcH10 (starting from initial states with Cdh1=1 and Rb=0, corresponding to S phase) (Table 2.5).

UbcH10 is probably involved also in Cdh1-dependent degradation of CycB. In the absence of definitive experimental data, we can test the impact of this mechanism on cell cycle dynamics using model checking on model variants with alternative rules for CycB. As shown in Table 2.6, for the second rule considered, CycB can be activated at the beginning of the cell cycle in presence of Cdh1 because UbcH10 has been degraded at the end of the previous cycle.

Phenotype: CycB should not be able to increase during G1-phase		
	CycE	CycB
Sequence	0	0
	0	1
Expected result	False	
Rule for CycB: (!UbcH10 !Cdc20) & !Cdh1	False	
Rule for CycB: !UbcH10 (!Cdc20 & !Cdh1)	True	

Table 2.6: Sequence properties for the irreversibility of mitosis and result for different rules for CycB.

Role of Skp2

Within the logical framework, it is relatively easy to extend a model to consider novel regulatory components and interactions. Here, we set to include an additional cell cycle inhibitory pathway. Indeed, recent evidence point to the existence of an E2F-independent proliferative control mechanism, which involves the F-box protein Skp2, a substrate recognition subunit of the SCP ubiquitin ligase complex that targets p27 for degradation [Dick and Rubin, 2013]. Briefly, Skp2 promotes the degradation of phosphorylated p27, thus enabling its degradation induced by CycE

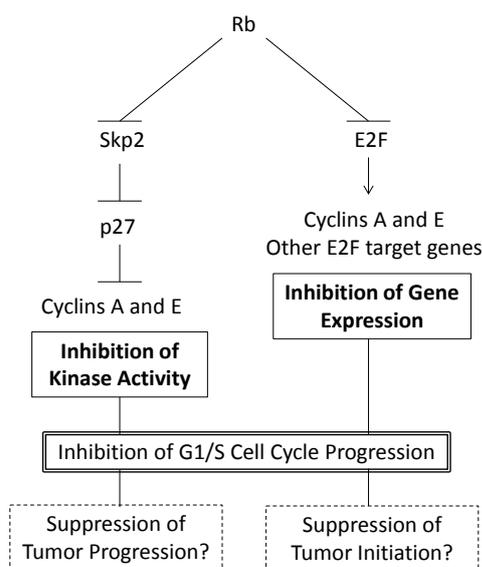


Figure 2.5: Schema from [Ji and Zhu, 2005] showing the additional cell cycle arresting pathway provided by the effect of Rb on Skp2.

and CycA. Rb binds to Cdh1 to participate in the ubiquitin-mediated degradation of Skp2 [Binné et al., 2007, Liu et al., 2008]. This mechanism links the two cell cycle repressors Rb and p27, and provides an additional mechanism by which Rb can arrest the cell cycle, as shown in Figure 2.5.

Consequently, we added a novel node representing Skp2 in the model, negatively regulated by Cdh1 and by non-phosphorylated Rb. The observation that E2F directly activates transcription of *skp2* gene [Assoian and Yung, 2008] led us to include a positive regulation by E2F opposing the inhibition by Rb. The performances of this revised model have been validated by checking consistency between simulated perturbations and experimental data.

[Ji et al., 2004] described the partial penetrance Rb mutation RbR661W, which impedes E2F repression, and effectively shuts down the Rb-E2F pathway. [Ji et al., 2004] further found that RbR661W retains the ability to arrest the cell cycle at the G1/S transition, with a p27 accumulation. They verified that Rb's ability to interact with Skp2-p27 was preserved in this mutant. Using the software GINsim, it is possible to model this subtle perturbation by specifically suppressing the regulation of E2F, CycE and CycA by Rb.

The authors further performed kinetic studies and showed that the Rb-induced G1 cell cycle arrest is initiated by the up-regulation of p27, which inhibits the kinase activity associated with cyclins E and A before the decline of protein levels. The cell cycle is still arrested through the (slower) repression of E2F target genes. Using the perturbation RbR661W, we could assess the early effect of the induction of Rb (Table 2.7), leading to simulation results consistent with published experimental results [Ji and Zhu, 2005]. In another experiment, the p27 antisense treatment prevents G1 arrest by Rb, showing that p27 is required for Rb-mediated G1 arrest. This can be modeled by considering another constraint (knock-down of p27) in the perturbation simulation, leading to a result consistent with experimental data.

Moreover, Skp2 specific perturbations have been reported in the literature, which can also be qualitatively reproduced with our model. In particular Skp2 KO has been shown to lead to severe proliferation defects with accumulation of both cyclin E and p27 [Nakayama et al., 2000, Kotoshiba et al., 2014]. Consistently, the simulation of this perturbation leads to a steady state with CycE and p27 both active (level 1).

Furthermore, the accumulation of cyclin E in this mutant has been interpreted by the suppression of cyclin E degradation. However, the negative regulation of cyclin E by Skp2 is not represented in our model because CycE represents the complex cyclin E/Cdk2. Skp2 is known

Experiment	Phenotype	Initial	Updated
RbR661W	Viable cell cycle in presence of growth factors, cell cycle arrest in absence with p27 accumulation	No	Ok
Rb induction (early kinetics)	Cell cycle arrest with present CycE and CycA	No	Ok
p27 KO (early kinetics)	Cell cycle in absence of growth factors	No	Ok
Skp2 KO	Cell cycle arrest or endoreplication with accumulation of cyclin E and p27	-	Ok
Skp2 KO p27 KO	Cell cycle in presence of growth factors, cell cycle arrest in absence	-	Ok

Table 2.7: Selected perturbations characterizing the Rb-Skp2-p27 pathway.

to degrade free cyclin E, while cyclin E complexed with Cdk2 is protected from Skp2-dependent degradation. This suggests an alternative mechanism by which CycE is accumulated in Skp2 mutants, presumably involving p27 binding to CycE. This complexation could inhibit CycE activity and arrest the cell cycle before the transition toward S phase. The rescue of Skp2 loss-of-function by the deletion of p27 [Kotoshiba et al., 2014] is also correctly reproduced by the model.

Finally, we design a sequence example out of the set of possible asynchronous sequences compatible with the experimentally observed sequence of variations of all the proteins encompassed by the model. Using model checking, we could verify that this sequence exists in the asynchronous graph, showing that the asynchronous dynamics of the updated model is compatible with experimental observations (Figure 2.6).

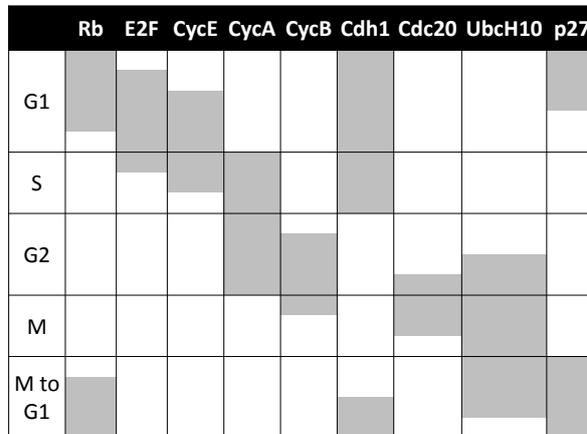


Figure 2.6: Correct asynchronous sequence for a complete cell cycle. The rows denote successive states along the cell cycle. Gray cells denote active nodes (level 1), while white cells represent inactive nodes (level 0).

2.5 Stochastic simulations of the logical model

More quantitative characterizations of asymptotic behaviors can be provided by stochastic simulations using MaBoSS (<http://maboss.curie.fr>). It is a command-line tool simulating continuous/discrete time Markov processes applied on Boolean models [Stoll et al., 2012]. Transitions

are associated with rates and a Gillespie algorithm produces time trajectories. Time evolution of probabilities are estimated. Global and semi-global characterizations of the whole system dynamics are further provided.

Without kinetic information, equal transition rates can be used to estimate a simple stochastic measure of trajectories in the asynchronous STG. Since MaBoSS is restricted to Boolean models, the logical model of the cell cycle is derived into a Boolean model by splitting the ternary node Rb into two Boolean nodes Rb1 and Rb2, associated with the first and second Rb thresholds, respectively. A population of 500,000 stochastic trajectories is simulated with this model, with equal transition rates. The mean level for each node over all trajectories is plotted in Figure 2.7, and reflect the kinetics of the cell cycle progression driven by the input cyclin D. Transient oscillations can be observed as the trajectories all start in G0 (with Rb1, Rb2, p27 and Cdh1 the only active nodes) and progressively desynchronize.

It is particularly interesting to compare the trajectories obtained for wild type (WT) versus perturbed conditions. The trajectories of five perturbations illustrate the role of Rb and of the pathway Rb-Skp2-p27 in the model (Figure 2.7).

Two perturbations were considered for Rb: the full loss-of-function (Rb KO), and a partial loss-of-function where Rb loses its ability to repress E2F but conserves its repressing activity on Skp2 (Rb R661W). The resulting stochastic trajectories highlight the role of Rb in the sequential activation of cyclin E and cyclin A, ensured by the repressing activity of the two underphosphorylated forms of Rb on E2F: in the WT case, the activation of cyclin A is clearly delayed relatively to the activation of cyclin E. In contrast, in the absence of the repressing effect of Rb on E2F, cyclin E and cyclin A are activated at the same time. The similarity between the trajectories of Rb R661W and Rb KO suggests that the repression of Skp2 by Rb has no major impact in the cell cycle. However, this interaction is necessary to ensure the quiescent state in the absence of cyclin D.

Skp2 loss-of-function (Skp2 KO) arrests the cell cycle (Figure 2.7). The cell cycle arrest is prompted by the stabilized p27, hence the restored oscillations in the double mutant Skp2 KO p27 KO.

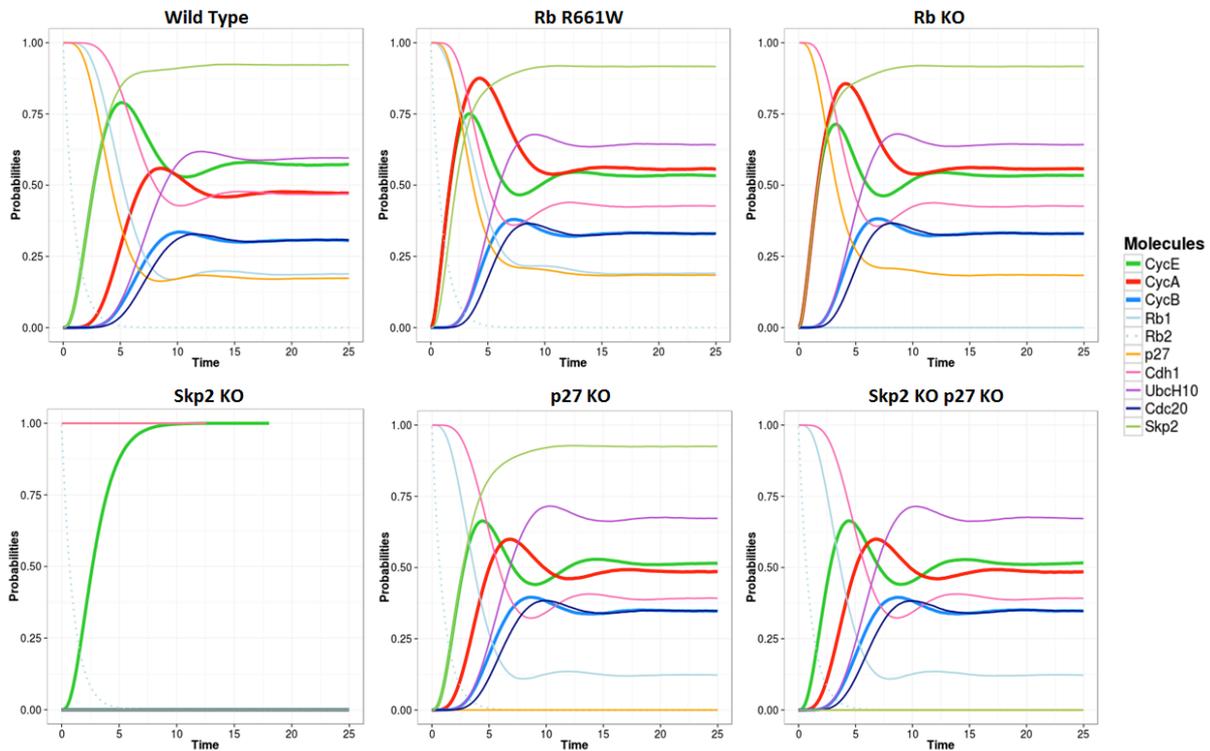


Figure 2.7: Stochastic trajectories simulated with MaBoSS for each component of the model of Figure 1.1, with equal transition rates for all transitions. From top right to bottom left: simulations without perturbation (wild-type); with a perturbation corresponding to the partial mutation RbR661W annihilating the repressing activity of Rb on E2F; Rb loss-of-function; ectopic expression of Skp2; p27 loss-of-function; combination of Skp2 and p27 loss-of-functions.

2.6 Conclusions and prospects

Model refinement

We have refined a logical model of the mammalian cell cycle to include recent data pointing to novel regulatory components and interactions. Each model update was evaluated using model checking to assess the conservation of documented dynamical properties, for the wild type cycle and for various genetic perturbations. Refining a model is an iterative process, and while we have focused here on the satisfaction of novel dynamical properties, it is as important to check at each step that all relevant properties satisfied by the original model are not altered. In appendix, we provide a complete list of the perturbations (Table A.1) and dynamical properties (Tables A.2 and A.3) evaluated on the initial and final models, along with their assessment for both models. As can be seen in these tables, the updated model complies with several additional properties, thereby constituting a clear improvement over the original one.

Prospects

A careful model checking analysis of our current model already points to some limitations. In particular, the order of activation or degradation of CycB relatively to the other cyclins is not properly determined (see appendix). This raises the question of what refinement could ensure the correct sequential order between cyclin B and the other cyclins. Relevant perturbation studies could bring an insight into possible mechanisms. For examples, the use of stabilized cyclins in

proliferating cells could help to conclude whether the degradation of cyclin E, cyclin A and cyclin B are necessary for the progression into the next phase. Alternatively, more subtle kinetic aspects might be responsible for the temporal regulation of cyclin B, which would then require the use of more quantitative approaches.

Our current model takes into account the most notable components of the mammalian cell cycle network. However, this model could be further updated in the light of novel experimental data. For example, several proteins have been involved in the regulation of the G2 phase, such as Aurora, Plk1, Emi1, but have not been yet included in our model because their role or regulation are still uncertain. These factors could be incorporated in the model once more precise mechanistic information becomes available.

Our model could also be refined to fit data concerning specific cell types, or to study the effect of multiple perturbations associated with cancer (e.g. [Grieco et al., 2013, Mombach et al., 2014, Flobak et al., 2015]). It can be used as a starting point for subsequent studies, using the model checking approach delineated here in order to increment and assess successive model versions. In this respect, we provided our current model in two computer readable formats (an archive that can be opened using GINsim, along with an SBML export) as supplementary material (<http://ginsim.org/node/189>). These files include extensive annotations and links to external databases, thereby documenting all components and interactions.

Finally, although the consideration of multivalued nodes is useful to represent multifunctional proteins such as Rb, the analysis of logical models with multivalued nodes can be difficult and is supported by a limited number of tools. To enable the import of our model in purely Boolean modelling tools, we further provide a Boolean version of our multivalued model, where the node Rb is split into two Boolean nodes Rb1 and Rb2. In particular, this Boolean model can be used to perform stochastic simulations using the software MaBoSS (<http://maboss.curie.fr>), provided that the user defines transition rates for component (up or down) updates. Such stochastic simulations have the potential to provide more quantitative characterizations of asymptotic behaviors, for wild type versus perturbed conditions, as seen in Section 2.5. While we have focused on mean trajectories in the asynchronous attractor, further analysis could be done with the change in probabilities of reaching this attractor in response to perturbations.

3. Formalizing complex behaviors with temporal constraints

In this chapter we describe how temporal constraints can be used to specify dynamical behaviors of continuous models, with a particular focus on oscillatory systems. After detailing the syntax of linear time logic (LTL), its first-order extension (FO-LTL) handling continuous variables and its use for continuous model checking and model analysis, we give examples of frequent dynamical behaviors characterizing biological systems, in particular oscillatory behaviors. We show how combining them allow to define precise specifications that can contain both qualitative and quantitative constraints.

3.1 First Order Linear Time Logic for continuous trajectories

Linear Time Logic (LTL) is adapted to analyze the behavior of deterministic automata formalized as linear sequences of states.

LTL has initially been used in systems biology to check the satisfaction of Boolean properties representing the behavior of biochemical reaction systems or gene regulatory networks. LTL was successfully used to analyze deterministic boolean models, for example with a synchronous updating, similarly to the way CTL is used to analyze asynchronous dynamics (see Chapter 2). Generalizing these techniques to quantitative models has been done in two ways: either by discretizing the different regimes of the dynamics in piece-wise linear or affine models [Batt et al., 2005], or by relying on numerical simulations and taking a first-order version of temporal logic (FO-LTL) with constraints on concentrations, as query language for numerical traces [Fages et al., 2004, Calzone et al., 2006]. This last approach, implemented in Biocham, is adopted in this work.

Linear Time Logic

Infinite trace

Like in the logical framework, LTL formulae are interpreted on a Kripke structure, i.e. a transition relation over a set of states such that each state has at least one successor. In this case each state

has exactly one successor.

Finite traces are composed of a finite number of temporal points associated with values, either obtained by biological experiments or by numerical integration. They can be directly translated to sequences of states, noted (s_0, \dots, s_n) with $n \in \mathbb{N}$ the number of states. Each state defines values for all state variables, which are in this case the concentrations of all molecular species in the model. Moreover, the first derivative $\frac{dA}{dt}$ of each concentration A is computed as the slope between each concentration point and the next one, and added to the state variables. Although in the temporal logic framework treatment of time is implicit, and considered only through the order of states in the sequence, in this case each state derived from a temporal point is also associated with a real time value. Consequently, the time is also included in the state variables and denoted by *Time*, allowing the introduction of precise time constraints in the properties that can be verified.

Finite traces are then complemented to infinite traces by adding a loop on the last state, $(s_0, \dots, s_n, s_n, \dots)$. The practical assumption behind this classical convention for interpreting temporal logic on finite traces is that the time horizon considered is sufficiently long for evaluating the formulae of interest and not interfering with the last state. The value of $\frac{dA}{dt}$ in the last state is defined as the 0 in order to maintain the coherence between the concentrations and their derivatives.

Syntax

The syntax of LTL is similar to the syntax of CTL already detailed in Section 2.3, minus the path quantifiers and with linear constraints over \mathbb{R} as atomic constraints. Hence the grammar of $\text{LTL}(\mathbb{R}_{\text{lin}})$ formulae:

$$\phi ::= c \mid \neg\phi \mid \phi \Rightarrow \psi \mid \phi \wedge \phi \mid \phi \vee \phi \mid \mathbf{X}\phi \mid \mathbf{F}\phi \mid \mathbf{G}\phi \mid \phi \mathbf{U}\phi \mid \phi \mathbf{W}\phi,$$

where c denotes linear constraints between state variables (molecular concentrations, their first derivatives, or the state time variable), and real numbers. $A > 1$, $\frac{dA}{dt} = 0$ and $A \leq B$ are examples of such constraints, where A and B are concentrations of molecular species.

In this interpretation over finite traces, the formula $\mathbf{G}\phi$ is true in the last state if ϕ is true in the last state. The dual properties seen in Section 2.3 without path quantifiers become:

- $\neg\mathbf{X}\phi = \mathbf{X}\neg\phi$,
- $\neg\mathbf{F}\phi = \mathbf{G}\neg\phi$,
- $\neg\mathbf{G}\phi = \mathbf{F}\neg\phi$,
- $\neg(\phi \mathbf{U} \psi) = \neg\psi \mathbf{W} \neg\phi$,
- $\neg(\phi \mathbf{W} \psi) = \neg\psi \mathbf{U} \neg\phi$.

For instance, the formula $\mathbf{F}(A \geq 0.2)$ expresses that the concentration of molecule A gets greater than 0.2 at some time point in the future (\mathbf{F}), and $\mathbf{F}(A \geq 0.2 \wedge \text{Time} < 1)$ expresses that this property should be satisfied before the variable *Time* reaches 1.

The syntax used in Biocham to write temporal logic formulae is shown in Table 3.1.

$\mathbf{F}, \mathbf{X}, \mathbf{G}, \mathbf{U}, \mathbf{W}$	temporal operators
$\&$	conjunction (\wedge)
$ $	disjunction (\vee)
$!$	negation (\neg)
$[A]$	concentration of molecule A
$d([A]) / dt$	derivative of $[A]$
$<, \leq$	less than, less than or equal to
$>, \geq$	greater than, greater than or equal to
\Rightarrow	implication ($\phi \Rightarrow \psi$ is equivalent to $\psi \vee \neg\phi$)

Table 3.1: Syntax for temporal logic formulae in Biocham.

Linear Time Logic with free variables

Although LTL formulae include quantitative information, the classic boolean evaluation of LTL formulae by true/false is not adapted to many problems.

To address this issue, a first-order extension of LTL with real free variables, denoted FO-LTL(\mathbb{R}_{lin}), can be used to store quantitative results [Fages and Rizk, 2008]. The grammar of LTL is extended with the quantifier operators $\exists x \phi$ | $\forall x \phi$ to project free variables. In Biocham, the syntax for each operator is, respectively, `Exists` and `Forall`.

For instance in the previous example, if needed, the precise time values where the concentration of A gets greater than the threshold value can be expressed by introducing a free variable t with an equality constraint to the real time variable, with the formula

$$\mathbf{F}(A \geq 0.2 \wedge t = \text{Time}).$$

Constraints between time variables can also relate the times of different events. For instance, the formula

$$\mathbf{G}(\text{Time} \leq t_1 \Rightarrow A < 1 \wedge \text{Time} \geq t_2 \Rightarrow A > 10) \wedge (t_2 - t_1 < 60)$$

expresses that the concentration A is always less than 1 up to some time t_1 , always greater than 10 after time t_2 , and the switching time between t_1 and t_2 is less than 60 units of time.

A local maximum for molecule concentration A can be defined with the formula

$$\mathbf{F}(A \leq x \wedge \mathbf{X}(A = x \wedge \mathbf{X}(A \leq x)))$$

specifying the existence of three successive states, where the value of A in the second state is greater than in the two other states. x is a free variable which takes this maximum value. Alternatively, local maxima can also be defined using the derivatives with the following formula, which specifies a change of the sign of the derivative of A , and extracts maxima values in the variable x with $x = A$:

$$\mathbf{F}\left(\frac{dA}{dt} \geq 0 \wedge \mathbf{X}\left(\frac{dA}{dt} < 0 \wedge x = A\right)\right).$$

We will see in Section 3.2 how such formulae can be combined and extended to define complex oscillation properties, with period and phase constraints defined as time separation constraints between local extrema of one or several molecular species.

Validity Domains of Free Variables

The satisfaction of a FO-LTL formula containing free variables is given by the validity domains of the variables.

Definition 3.1 Validity domain

The validity domain of an FO-LTL(\mathbb{R}_{lin}) formula $\phi(\mathbf{y})$ involving a vector of q real free variables \mathbf{y} , on a finite trace $T = (s_0, \dots, s_n)$, is the set of values of \mathbf{y} for which ϕ is true:

$$\mathcal{D}_{T,\phi} = \{\mathbf{y} \in \mathbb{R}^q \mid T \models \phi(\mathbf{y})\}$$

It is computed with the algorithm 1.

A linear temporal logic formula is false if $\mathcal{D}_{T,\phi} = \emptyset$, valid if $\mathcal{D}_{T,\phi} = \mathbb{R}^q$, and satisfiable otherwise, for some values given by $\mathcal{D}_{T,\phi}$.

It has been shown in [Fages and Rizk, 2008] that the validity domain computed by the algorithm is a finite union of orthotopes, with an orthotope being the cartesian product of v intervals over \mathbb{R} .

The recursive definition of validity domains (Definition 3.1) can be directly implemented to compute the validity domains of the free variables in each point of the numerical trace, starting from the last to the first. In this computation the LTL subformulae are considered in the bottom-up order, i.e. first from the linear constraints at the leaves, up to the root of the syntactic tree.

Algorithm 1 Validity domain

The validity domain of an FO-LTL(\mathbb{R}_{lin}) formula $\phi(\mathbf{y})$ involving a vector of q real free variables \mathbf{y} , on a finite trace $T = (s_0, \dots, s_n)$, is computed by induction on T and the subformulae of $\phi(\mathbf{y})$ by using the following equations:

- $\mathcal{D}_{T,\phi} = \mathcal{D}_{s_0,\phi}$,
- $\mathcal{D}_{s_i,c(\mathbf{x})} = \{\mathbf{v} \in \mathbb{R}^k \mid s_i \models c[\mathbf{v}/\mathbf{x}]\}$ for a constraint $c(\mathbf{x})$ depending on a k -variables vector \mathbf{x} evaluated with \mathbf{v} ,
- $\mathcal{D}_{s_i,\phi \wedge \psi} = \mathcal{D}_{s_i,\phi} \cap \mathcal{D}_{s_i,\psi}$,
- $\mathcal{D}_{s_i,\phi \vee \psi} = \mathcal{D}_{s_i,\phi} \cup \mathcal{D}_{s_i,\psi}$,
- $\mathcal{D}_{s_i,\neg\phi} = \mathbb{C} \mathcal{D}_{s_i,\phi}$,
- $\mathcal{D}_{s_i,\exists x\phi} = \Pi_x \mathcal{D}_{s_i,\phi}$,
- $\mathcal{D}_{s_i,\forall x\phi} = \mathcal{D}_{s_i,\neg\exists x\neg\phi}$,
- $\mathcal{D}_{s_i,\mathbf{X}\phi} = \mathcal{D}_{s_{i+1},\phi}$ if $i < n$,
- $\mathcal{D}_{s_n,\mathbf{X}\phi} = \mathcal{D}_{s_n,\phi}$,
- $\mathcal{D}_{s_i,\mathbf{F}\phi} = \bigcup_{j=i}^n \mathcal{D}_{s_j,\phi}$,
- $\mathcal{D}_{s_i,\mathbf{G}\phi} = \bigcap_{j=i}^n \mathcal{D}_{s_j,\phi}$,
- $\mathcal{D}_{s_i,\phi \mathbf{U} \psi} = \bigcup_{j=i}^n (\mathcal{D}_{s_j,\psi} \cap \bigcap_{k=i}^{j-1} \mathcal{D}_{s_k,\phi})$,
- $\mathcal{D}_{s_i,\phi \mathbf{W} \psi} = \mathcal{D}_{s_i,\phi \mathbf{U} \psi} \cup \mathcal{D}_{s_i,\mathbf{G}\phi}$.

where \mathbb{C} is the set complement operator over domains, and Π_x is the domain projection operator out of x , restoring domain \mathbb{R} for x .

For instance, to evaluate the formula $\mathbf{F}(A \geq y_1 \wedge \mathbf{F}(A \leq y_2))$ of Example 3.3, the validity domains of y_1, y_2 for the subformulae $A \geq y_1$ and $A \leq y_2$ are first computed in each time point of the trace, then the validity domains for $\mathbf{F}(A \leq y_2)$ in each time point from the last to the first, and last for the complete formula by composing the results for the subformulae from the last time point to the first.

Geometrical operations on linear constraints

Linear constraints over the reals have a simple geometrical interpretation: a conjunctive state constraint represents a (possibly non-closed) convex polyhedron in the state variables' space while a disjunctive state constraint represents a (non convex) finite union of convex polyhedra. Domain operations with linear constraints can thus be implemented quite straightforwardly with a polyhedra manipulation library. The implementation in BIOCHAM [Calzone et al., 2006] uses the Parma Polyhedra Library (PPL) [Bagnara et al., 2008] that performs set operations for dealing with finite unions of polyhedra [Rizk, 2011].

- The conjunction of linear constraints is represented by the intersection polyhedron of the polyhedra associated with the constraints.
- The existential quantification of a variable is directly implemented by the projection operation of a polyhedron on the subspace of the space without that variable.
- A disjunction of polyhedra is represented by a finite set of polyhedra.
- The complementary of a polyhedron p is computed by the unions of the negated constraints describing p .
- The universal quantification is implemented by double complementation of the existential quantification.

Crucial to the efficiency of the set-based implementation of disjunctive constraints is the elimination of redundant constraints. The subsumption check between finite sets of linear state constraints is co-NP hard, since checking whether one convex polyhedron is contained in a finite union of convex polyhedra is co-NP complete [Srivastava, 1993]. On the other hand, the local subsumption check, i.e. checking whether one convex polyhedron is contained in another one, can be done by linear programming in polynomial time.

Example 3.2 Reachability

The following formula tests whether a concentration threshold is reached by a molecule.

- FO-LTL(\mathbb{R}_{lin}) formula: $\phi_1(v) = F(A \geq v)$
- Validity domain $\mathcal{D}_{T,\phi_1} =]-\infty, \max A]$

In the formula ϕ_1 in example 3.2, A denotes the concentration of a molecule, and v is a free variable. The figure 3.1 schematizes the generic solving procedure described in Definition 3.1 for ϕ_1 .

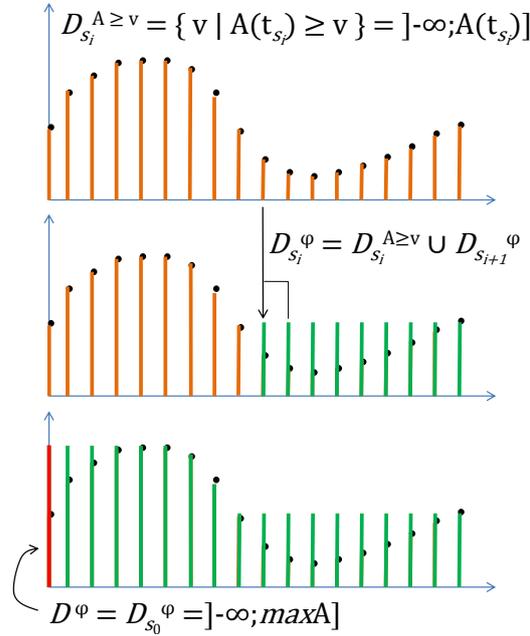


Figure 3.1: Schema describing the computing of the validity domain for $F(A \geq v)$. On the top panel, for each point s_i of the trace, the validity domain $\mathcal{D}_{s_i}^{A \geq v}$ corresponding to the satisfied values of v for which the formula $A \geq v$ is satisfied in s_i is computed (intervals $]-\infty, A(t_{s_i})]$ in orange). In the middle panel, for each point s_i starting from the last point of the trace, the validity domain $\mathcal{D}_{s_i}^{\phi_1}$ for which ϕ_1 is satisfied in s_i is obtained recursively with the union $\mathcal{D}_{s_i}^{\phi_1} = \mathcal{D}_{s_i}^{A \geq v} \cup \mathcal{D}_{s_{i+1}}^{\phi_1}$ (intervals in green). On the bottom panel, the domain for the first point is finally taken as the domain for the whole trace (interval in red): $\mathcal{D}^{\phi_1} = \mathcal{D}_{s_0}^{\phi_1}$.

Example 3.3 Extrema

The following FO-LTL(\mathbb{R}_{lin}) pattern expresses two global comparisons for the concentration of the molecule on the whole trace. This extracts the greatest value reached by the concentration as the boundary of the validity domain projected on the variable y_1 , and the smallest value reached as the boundary of the validity domain projected on y_2 .

- FO-LTL(\mathbb{R}_{lin}) formula: $\phi_2(y_1, y_2) = F(A \geq y_1) \wedge F(A \leq y_2)$
- Validity domain $\mathcal{D}_{T,\phi_2} =]-\infty, \max A] \times [\min A, +\infty[$

On the numerical trace of Figure 3.2, the validity domain of $\phi_2(y_1, y_2)$ is the domain $y_1 \leq 10 \wedge y_2 \geq 2$.

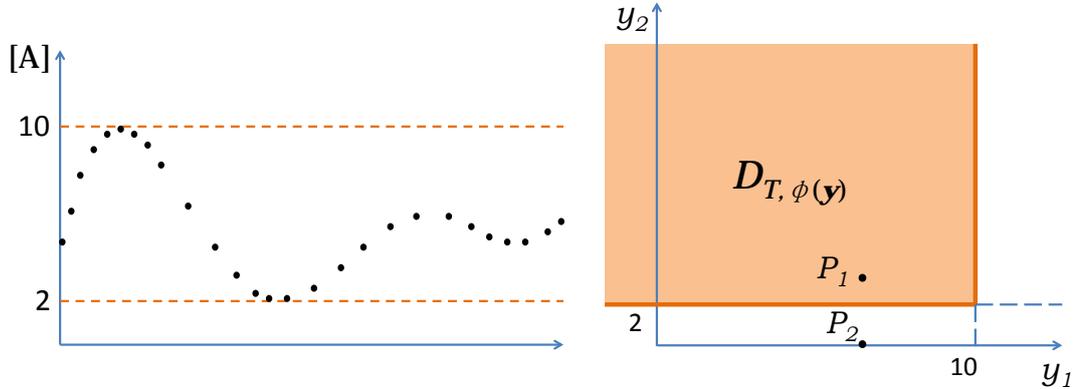


Figure 3.2: Left: numerical trace for the evolution of the concentration of molecule A over time. Right: validity domain $]-\infty, 10] \times [2, +\infty[$ for ϕ_2 on the trace.

The two points $P_1 = (7, 3)$ and $P_2 = (7, 0)$ in this space correspond to the closed formulae $\psi_1 = \mathbf{F}(A \geq 7 \wedge \mathbf{F}(A \leq 3))$ and $\psi_2 = \mathbf{F}(A \geq 7 \wedge \mathbf{F}(A \leq 0))$ respectively. In the first case, the formula is true since the point is inside the validity domain, and the second formula is false since the point is outside the validity domain.

Example 3.4 Sequence of comparisons

The more complex following FO-LTL(\mathbb{R}_{lin}) pattern expresses a sequence of events: it is satisfied when the concentration of the molecule is first above some value denoted by the free variable y_1 and then below some other denoted by the free variable y_2 .

- FO-LTL(\mathbb{R}_{lin}) formula: $\phi_2(y_1, y_2) = \mathbf{F}(A \geq y_1 \wedge \mathbf{F}(A \leq y_2))$
- Validity domain $\mathcal{D}_{T, \phi_3} = \bigcup_{i=0}^n]-\infty, s_i] \times ([\max(s_i, \min_{j \geq i} s_j), +\infty[$

The corresponding validity domain is more complex than for ϕ_2 , and can be computed formally according to the algorithm 1:

$$\begin{aligned} \mathcal{D}_{T, \phi_3} &= \mathcal{D}_{T, \mathbf{F}(A \geq y_1 \wedge \mathbf{F}(A \leq y_2))} \\ &= \bigcup_{i=0}^n (\mathcal{D}_{s_i, A \geq y_1} \cap (\bigcup_{j=i}^n \mathcal{D}_{s_j, A \leq y_2})) \\ &= \bigcup_{i=0}^n (]-\infty, s_i] \times ([\max(s_i, \min_{j \geq i} s_j), +\infty[) \end{aligned}$$

We describe in the next subsection how the distance to the validity domain is used to define a continuous satisfaction degree for FO-LTL(\mathbb{R}_{lin}) formulae.

Continuous Satisfaction Degree in $[0, 1]$

The true/false valuation of FO-LTL(\mathbb{R}_{lin}) formulae makes it possible to scan the parameter space and check for each parameter set whether the temporal specification is satisfied. However, such generate-and-test methods have an exponential complexity in the number of parameters. They are thus limited to two or three parameters and do not scale up.

To provide a quantitative evaluation of an FO-LTL(\mathbb{R}_{lin}) specification on a trace, a continuous satisfaction degree is derived from distance between the the validity domain and an objective value for one or several variables [Rizk et al., 2008]. This allows to score a simulation or experimental trace with respect to an objective behavior.

Calculating a continuous satisfaction degree for an FO-LTL(\mathbb{R}_{lin}) specification is particularly useful for parameter inference, since comparing satisfaction degrees obtained with different parameter sets provides a direction to follow in the parameter space. A continuous satisfaction degree able to measure progress towards satisfaction makes it possible to move from generate-and-test procedures for parameter scanning, to powerful continuous optimization methods for computing dozens of parameter values in one run, for instance by using evolutionary algorithms with the satisfaction degree as fitness function.

This is the strategy implemented in BIOCHAM [Calzone et al., 2006] for parameter search, where the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [Hansen and Ostermeier, 2001] is updated with the satisfaction degree of the temporal specification as fitness function [Rizk et al., 2011]. Its iterative procedure is schematized in Figure 3.3. It is used in the application part of this dissertation (Chapter 5).

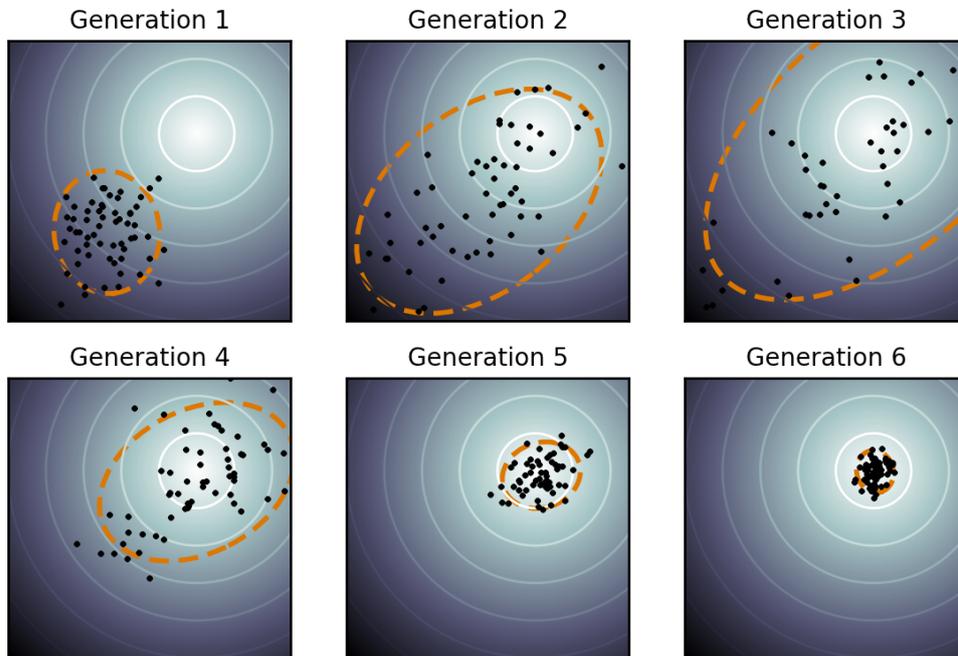


Figure 3.3: Illustration of an optimization run with covariance matrix adaptation on a simple two-dimensional problem. The spherical optimization landscape is depicted with solid lines of equal fitness function values. The darker the background color, the greater the fitness value on this point. At each iteration, a new population of parameter sets (dots) is generated and its distribution (dotted line) is updated. On this simple problem, the population (dots) is larger than necessary and concentrates over the global optimum within a few generations. Figure from <http://en.wikipedia.org/wiki/CMA-ES>

The continuous satisfaction degree can be used for parameter analysis: the evolution of the satisfaction degree is used to visualize how the satisfaction of a property evolves when some parameter values change. However, the effect of two parameters at most can be assessed with this method, since such representations are limited to one-dimensional response curves or two-dimensional landscapes (examples can be seen in Section 3.3).

Definition 3.5 Let π be a numerical trace.

- The violation degree of a formula ϕ with free variables \mathbf{x} with respect to objective values \mathbf{v} is the euclidean distance between the point of coordinates \mathbf{v} and the validity domain ($D_{\pi,\psi}$) if it is not empty:

$$vd(\phi, \mathbf{v}) = d(\mathbf{v}, D_{\pi,\psi})$$

If $D_{\pi,\psi}$ is empty, and $vd(\phi, \mathbf{v}) = +\infty$.

- The satisfaction degree of ϕ w.r.t. \mathbf{v} is

$$sd(\phi, \mathbf{v}) = \frac{1}{1 + vd(\phi, \mathbf{v})}$$

- The robustness degree is the violation degree of the negation of ϕ , i.e. the distance to the complement of the validity domain:

$$rb(\phi, \mathbf{v}) = vd(\neg\phi, \mathbf{x})$$

Example 3.6 Going back to example 3.3, we have

$$vd(\psi_1) = 0, \quad sd(\psi_1) = 1, \quad ro(\psi_1) = 1,$$

and

$$vd(\psi_2) = 2, \quad sd(\psi_2) = 1/3, \quad ro(\psi_2) = 0.$$

The absolute values of the satisfaction degree are not meaningful but the relative values make it possible to compare different parameter sets, and improve them in the most promising direction.

The robustness degree is defined here as the distance between the objective and the complement of the validity domain. Indeed, this distance indicates a degree of robustness of the temporal properties under the assumption that if the validity domain boundaries are far from the objective, the validity domain obtained after slightly changing some parameters are close to the first one, and are thus likely to satisfy the specification. However, in non-linear systems this assumption may be violated, and the robustness of the system must be more accurately estimated by sampling the parameter space around the parameter set [Rizk et al., 2009].

3.2 Formalizing common properties for oscillatory systems with temporal logic

Analyzing models of biochemical reaction systems that exhibit complex dynamics require to formalize sophisticated behaviors. We investigate here the expressivity of temporal logic to formalize various behaviors for such dynamics.

This is particularly useful in the case of oscillating systems. Different properties characterize an oscillator: the amplitude, the period, the phase relationships if several components are involved, the regularity and shape of the oscillations.

Specifying such properties with temporal logic allows much more flexibility than simple curve fitting or specific evaluation methods such as period estimation. Indeed, it is possible to evaluate only the quantitatively known properties, while overlooking the unknown aspects of the oscillations and instead relying on the more general qualitative description of the existence of oscillations. Moreover, several properties of interest can be easily combined in a longer formula and evaluated together. For instance, a trace can be evaluated with respect to expected periods and phase relationships, while keeping the amplitude free of constraint.

Furthermore, accounting for imprecise quantitative information is also possible. Boundaries can be set to constraint the amplitude only inside an estimated interval.

Here we present a selection of common properties characterizing an oscillatory system, such as pseudo-period, pseudo-phase, and measures of the regularity of the oscillations. Validity

domains of each FO-LTL formula are then computed with the generic algorithm 1. In the next chapter, dedicated solvers for the same common properties will be implemented for a more efficient evaluation.

Example 3.7 Existence of a local maximum at a time t

The formula $\phi_4(t)$ specifies that at some point in the trace the derivative of a concentration A should have a positive or null value, followed by a negative value at the next point. The time value of this next point is stored in the variable t . Hence, $\phi_4(t')$ is satisfied when there exists a local maximum at the time point t' .

- FO-LTL(\mathbb{R}_{lin}) formula:

$$\phi_4(t) = \mathbf{F}(\frac{dA}{dt} \geq 0 \wedge \mathbf{X}(\frac{dA}{dt} < 0 \wedge Time = t))$$
- Validity domain $D = \{T_i\}$: the set of local maxima times of the trace.

Note that if the formula is used with $\frac{dA}{dt} \leq 0$ instead of $\frac{dA}{dt} < 0$, the last point of the trace will appear in the result as a spurious peak because it satisfies $\frac{dA}{dt} = 0$.

This simple temporal logic pattern capturing the peaks of a trace is the basic brick with which more complex formulae are built to characterize oscillations in a trace.

The validity domain of $\phi_4(t)$ is the set of all local maxima in the trace. It is possible to deduce the maximum and the minimum of these values with the following operations:

$$\max_x D_{\phi(x)} = D_{\phi(x) \wedge \forall x', (\neg \phi(x') \vee x' \geq x)} = D_{\phi(x) \wedge \neg \exists x' | (\phi(x') \wedge x' > x)} \quad (3.1)$$

$$\min_x D_{\phi(x)} = D_{\phi(x) \wedge \forall x', (\neg \phi(x') \vee x' \leq x)} = D_{\phi(x) \wedge \neg \exists x' | (\phi(x') \wedge x' < x)} \quad (3.2)$$

The schema in Figure 3.4 represents such a projection of a validity domain composed of disjointed single values in order to obtain the highest value.

An example of FO-LTL(\mathbb{R}_{lin}) formula adopting this pattern is given in Example 3.8.

Example 3.8 Smallest local maximum at a time t

$\phi_5(t)$ captures the smallest local maximum time in the trace.

- FO-LTL(\mathbb{R}_{lin}) formula:

$$\phi_5(t) = \mathbf{F}(\frac{dA}{dt} \geq 0 \wedge \mathbf{X}(\frac{dA}{dt} < 0 \wedge Time = t)) \wedge \neg \exists t' | (\mathbf{F}(\frac{dA}{dt} \geq 0 \wedge \mathbf{X}(\frac{dA}{dt} < 0 \wedge Time = t')) \wedge t' < t)$$
- Validity domain $\mathcal{D} = \{\min_i(T_i)\}$.

```
biocham: validity_domain(F((d([X])/dt>=0) & X(d([X])/dt<0 & Time=t))
& !(Exists([t2],F((d([X])/dt>=0) & X(d([X])/dt<0 & Time=t2))&t2<t))
).
t = 5.29325
Time elapsed : 156 ms
```

Listing 3.1: Evaluation of ϕ_5 on the oscillatory trace in Figure 3.6 with Biocham

ϕ_4 can be extended with the operator **U** (until) to capture whole intervals between the single points where the sign of the derivative changes. In Example 3.9, it is used to measure what we call the *left amplitude* of a peak, defined as the difference between the value at the local maximum and the value at the previous local minimum, as shown in Figure 3.5. Left amplitudes can be used as a relevant measure to assess the variation in peak shapes. Obviously, its counterpart the *right amplitude* could be used similarly, or more complex measures such as the maximum, minimum or mean value of the left and right amplitudes could be defined if needed with more complex FO-LTL formulae.

Example 3.9 Left peak amplitude

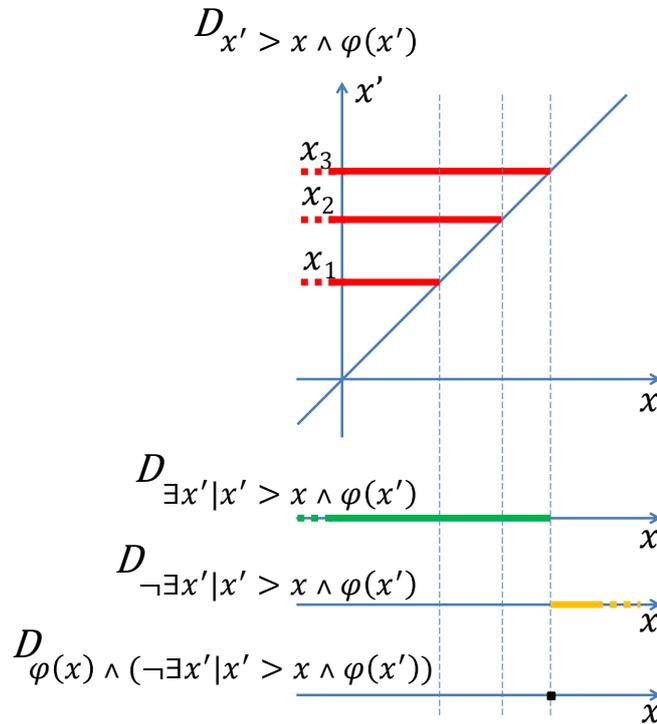


Figure 3.4: Schema describing the projection of a validity domain composed of distinct single values in order to obtain the highest value. Say that in this case, the validity domain of ϕ is a set of values $\{x_i\}$. The conjunction $\phi(x') \wedge x' > x$ turns each value into a open half-line in the space (x, x') (in red). Then the operator \exists projects the domain on the x dimension. The result is the interval $]-\infty, \max_i(x_i)[$ (in green). The negation takes the complementary interval (in yellow), then the conjunction with $\phi(x)$ intersects this interval with the set $\{x_i\}$ and restores the single value domain $\{\max_i(x_i)\}$ (in black).

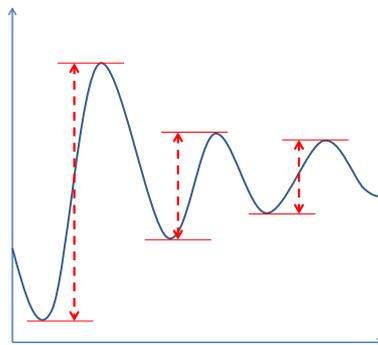


Figure 3.5: Left amplitudes of peaks in a trace: difference between the value at the local maximum and the value at the previous local minimum.

- *FO-LTL(\mathbb{R}_{lin}) formula:*

$$\phi_6(a) = \exists(m_1) | (F(\frac{dA}{dt} < 0 \wedge X(\frac{dA}{dt} \geq 0 \wedge A = m_1 \wedge (\frac{dA}{dt} \geq 0) \mathbf{U}(\frac{dA}{dt} < 0 \wedge A = m_1 + a))))$$
- *Validity domain $\mathcal{D} = \{a_i\}$: the set of left amplitudes of local maxima for the trace of A , defined as the difference between the local maximum value and the preceding local minimum value.*

The validity domain of $\phi_6(a)$ is the set of left amplitudes of all peaks in the trace. As explained before, the maximum of these values can be found with the operator $\exists(a)$. The formula $\phi_7(a)$ in Example 3.10 uses a similar approach to capture the greatest left amplitude of all peaks in the trace, as the upper bound of the validity domain.

Example 3.10 Greatest left peak amplitude

- *FO-LTL(\mathbb{R}_{lin}) formula:*

$$\phi_7 = \exists(m_1) | (F(\frac{dA}{dt} < 0 \wedge X(\frac{dA}{dt} \geq 0 \wedge A = m_1 \wedge (\frac{dA}{dt} \geq 0) \mathbf{U}(\frac{dA}{dt} < 0 \wedge A - m_1 \geq a))))$$
- *Validity domain $\mathcal{D} =]-\infty, a_j]$, where $a_j = \max_i a_i$.*

Similarly to Example 3.8, if a single-value domain is needed rather than a half-line (for example to compare the greatest left peak amplitude to a higher value), a variant of ϕ_7 which repeats the same sub-formula with an equality constraint on a can be used.

Among the features of an oscillator, the period of the oscillations is one of the most important. In a finite trace, measuring the time distances between successive peaks gives a first approximation of the period. The formula $\phi_8(d)$ in Example 3.11 extends the formula $\phi_6(t)$ to capture the times of two successive local maxima t_1 and t_2 and extract the difference d between the two times.

Example 3.11 Distance between successive peaks

- *FO-LTL(\mathbb{R}_{lin}) formula:*

$$\begin{aligned} \phi_8(d) = & \exists(t_1, t_2) | (t_2 - t_1 = d \\ & \wedge F((\frac{dA}{dt} > 0) \wedge X((\frac{dA}{dt} \leq 0) \wedge (Time = t_1) \\ & \wedge ((\frac{dA}{dt} \leq 0) \mathbf{U}((\frac{dA}{dt} > 0) \wedge ((\frac{dA}{dt} > 0) \mathbf{U}((\frac{dA}{dt} \leq 0) \wedge (Time = t_2)))))))) \end{aligned}$$
- *Validity domain $\mathcal{D} = \{d_i = T_{i+1} - T_i, i \in [1; n - 1]\}$*

For instance, computing the validity domain corresponding to the formula $\phi_8(d)$ on the oscillatory trace in Figure 3.6 (time horizon of 180h) with Biocham is shown in Listing 3.2. Biocham prints the resulting validity domain, which is a disjunction of orthotopes, with one orthotope per line.

```
biocham: validity_domain(
Exists([t1,t2],t2-t1=d
& F((d([X])/dt>=0) & X(d([X])/dt<0 & Time=t1
& (d([X])/dt<0) U (d([X])/dt>=0
& (d([X])/dt>=0) U (d([X])/dt<0 & Time=t2
))))).
d = 30.0234
| bb
d = 24.7015
|
```

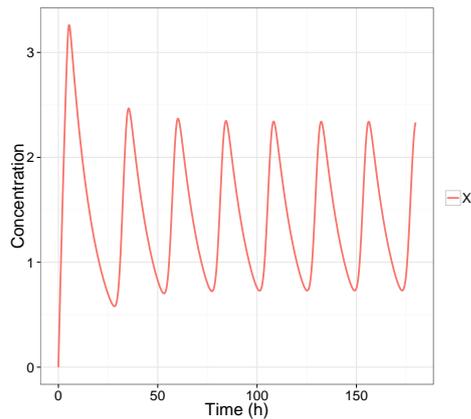


Figure 3.6: Simulated traces of a toy oscillator model described in Section 4.2 and detailed in Appendix A.3. The trace obtained with a Rosenbrock’s implicit simulation method in Biocham is discrete but plotted as a continuous line because of the high number of points.

```
d = 24.4486
|
d = 23.9441
|
d = 23.9091
|
d = 23.9032
Time elapsed : 192 ms
```

Listing 3.2: Evaluation of ϕ_8 on the oscillatory trace in Figure 3.6 with Biocham.

Oscillations can exhibit irregularities during a transitory time at the beginning of the simulation, in particular when initial concentrations are not calibrated on the oscillatory regime, as in the case shown in Figure 3.6. As a consequence, $\phi_8(d)$ captures unwanted values corresponding to the first peak-to-peak intervals ($d = 30.0234$).

This can be corrected by specifying a lower bound for the time t_1 of the first peak. In Listing 3.3, the validity domain is restricted to peak-to-peak intervals measured after a time of 80h, and is more relevant for the pseudo-period:

```
biocham: validity_domain(Exists([t1,t2],t2-t1=d
& F((d([X])/dt>=0) & X(d([X])/dt<0 & Time=t1 & t1>80
& (d([X])/dt<0) U (d([X])/dt>=0
& (d([X])/dt>=0) U (d([X])/dt<0 & Time=t2
))))).
d = 23.9441
|
d = 23.9091
|
d = 23.9032
Time elapsed : 200 ms
```

Listing 3.3: Evaluation of a variant of ϕ_8 specifying peak-to-peak intervals after 80h on the oscillatory trace in Figure 3.6 with Biocham.

It is also possible to extract specifically the last peak-to-peak interval, which is useful if the transitory duration is long or unknown (e.g. when the model is repeatedly simulated in a calibration procedure).

To this end, the operator \exists can be used in a variant of $\phi_4(t)$ to capture the maximal peak time:

$$\neg\exists(t)|(\mathbf{F}((\frac{dA}{dt} \geq 0) \wedge \mathbf{X}(\frac{dA}{dt} < 0 \wedge \text{Time} = t)) \wedge t > \text{tmax})$$

This temporal logic pattern is used on the last line of the example 3.12. The penultimate line of $\phi_9(\text{period})$ turns the validity domain of the variable tmax from the interval $[\max_i(T_i), +\infty, [$ to the single value $\max_i(T_i)$, which is the time of the last peak of the trace. This time is then used to extract the last two peak-to-peak intervals p_1 and p_2 . The mean of their values is returned with the variable period (cf Example 3.12) and gives a more precise value of the pseudo-period.

Example 3.12 Mean of the last two peak-to-peak intervals

- FO-LTL(\mathbb{R}_{lin}) formula:

$$\begin{aligned} \phi_9(\text{period}) = & \exists(p_1, p_2, t_1, t_2, \text{tmax})| (\\ & p_1 = t_2 - t_1 \wedge p_2 = \text{tmax} - t_2 \wedge 2 * \text{period} = p_1 + p_2 \\ & \wedge \mathbf{F}((\frac{dA}{dt} \geq 0) \wedge \mathbf{X}(\frac{dA}{dt} < 0 \wedge \text{Time} = t_1 \\ & \wedge (\frac{dA}{dt} < 0) \mathbf{U}(\frac{dA}{dt} \geq 0 \wedge (\frac{dA}{dt} \geq 0) \mathbf{U}(\frac{dA}{dt} < 0 \wedge \text{Time} = t_2 \\ & \wedge (\frac{dA}{dt} < 0) \mathbf{U}(\frac{dA}{dt} \geq 0 \wedge (\frac{dA}{dt} \geq 0) \mathbf{U}(\frac{dA}{dt} < 0 \wedge \text{Time} = \text{tmax})))))) \\ & \neg\exists(t)|(\mathbf{F}((\frac{dA}{dt} > 0) \wedge \mathbf{X}(\frac{dA}{dt} \leq 0 \wedge \text{Time} = t)) \wedge t > \text{tmax}) \end{aligned}$$

- Validity domain $\mathcal{D} = \frac{T_n + T_{n-1}}{2}$ where $T_n = \max_i(T_i)$.

Computing the validity domain corresponding to the formula $\phi_9(\text{period})$ on the trace in Figure 3.6 gives the following result:

```
biocham: validity_domain(Exists([p1,p2,t1,t2,tmax],
p1 = t2-t1 & p2 = tmax - t2 & 2*period=p1+p2
& F((d([X])/dt>=0) & X(d([X])/dt<0 & Time=t1
& (d([X])/dt<0) U (d([X])/dt>=0
& (d([X])/dt>=0) U (d([X])/dt<0 & Time=t2
& (d([X])/dt<0) U (d([X])/dt>=0
& (d([X])/dt>=0) U (d([X])/dt<0 & Time=tmax))))))
& !(Exists([t],F((d([X])/dt>=0) & X(d([X])/dt<0 & Time=t)) & t>tmax))).
period = 23.9042
Time elapsed : 1424 ms
```

Listing 3.4: Computing the mean of the last two peak-to-peak intervals on the oscillatory trace in Figure 3.6 with Biocham.

The formula $\phi_{10}(d)$ detailed in Example 3.13 can be used to determine the phase-shift between two oscillating molecules. This variant of the formula $\phi_8(d)$ captures a peak of a concentration A followed by a peak of concentration B . It takes advantage of the fact that $\text{False} \mathbf{U} \psi$ is true if ψ is true. In that case, the next peak of B is captured whether $\frac{dB}{dt} \geq 0$ or $\frac{dB}{dt} < 0$ when A reaches its peak.

Example 3.13 Phase shift between two molecules

- FO-LTL(\mathbb{R}_{lin}) formula:

$$\begin{aligned} \phi_{10}(d) = & \exists(t_1, t_2)| (t_2 - t_1 = d) \\ & \wedge \mathbf{F}((\frac{dA}{dt} \geq 0) \wedge \mathbf{X}((\frac{dA}{dt} < 0) \wedge \text{Time} = t_1 \\ & \wedge ((\frac{dB}{dt} < 0) \mathbf{U}((\frac{dB}{dt} \geq 0) \wedge ((\frac{dB}{dt} \geq 0) \mathbf{U}((\frac{dB}{dt} < 0) \wedge (\text{Time} = t_2)))))))) \end{aligned}$$

- *Validity domain* $\mathcal{D} = \{d_{i,j}\}$ with $d_{i,j} = T_{A_j} - T_{B_i}, i \in [1; n], j \in [i; n]$ where $\{T_{A_i}\}$ is the set of local maxima times for the trace of A and $\{T_{B_i}\}$ is the set of local maxima times for the trace of B.

Because they only depend on the existence of at least three peaks in the trace, formulae such as ϕ_8 to ϕ_{10} , specifying pseudo-periods and pseudo-phases are flexible, and can be evaluated even for irregular or damped oscillations.

The regularity of oscillations need to be assessed with other means: additional specifications can be used to measure dissimilarities between peaks, such as differences between peak amplitudes or peak-to-peak intervals. Such measures can then be constrained with thresholds to filter out irregular oscillations.

For instance, the formula $\phi_{11}(\text{diff})$ in Example 3.14 measures the maximal difference between two peak-to-peak intervals in the trace, and the formula $\phi_{12}(\text{diff})$ in Example 3.15 measures the maximal difference between two peak amplitudes.

Example 3.14 Maximal difference between peak-to-peak intervals

First let us define a subformula $\phi'(\text{diff})$ extracting in diff the difference between two successive peak-to-peak intervals:

$$\begin{aligned} \phi'(\text{diff}) = & \exists(t1, t2, t3, t4, p1, p2) | (t3 \geq t2 \wedge t2 - t1 = p1 \wedge t4 - t3 = p2 \\ & \wedge (p2 - p1 = \text{diff} | p1 - p2 = \text{diff}) \wedge \text{diff} \geq 0 \\ & \wedge \mathbf{F}(\frac{dA}{dt} \geq 0 \wedge \mathbf{X}(\frac{dA}{dt} < 0 \wedge \text{Time} = t1 \\ & \wedge (\frac{dA}{dt} < 0) \mathbf{U}(\frac{dA}{dt} \geq 0 \wedge (\frac{dA}{dt} \geq 0) \mathbf{U}(\frac{dA}{dt} < 0 \wedge \text{Time} = t2)))) \\ & \wedge \mathbf{F}(\frac{dA}{dt} \geq 0 \wedge \mathbf{X}(\frac{dA}{dt} < 0 \wedge \text{Time} = t3 \\ & \wedge (\frac{dA}{dt} < 0) \mathbf{U}(\frac{dA}{dt} \geq 0 \wedge (\frac{dA}{dt} \geq 0) \mathbf{U}(\frac{dA}{dt} < 0 \wedge \text{Time} = t4)))) \end{aligned}$$

- FO-LTL(\mathbb{R}_{lin}) formula: $\phi_{11}(\text{diff}) = \phi'(\text{diff}) \wedge \neg \exists \text{diff2} | (\text{diff2} > \text{diff} \wedge \phi'(\text{diff2}))$
- *Validity domain* $\mathcal{D} = \max_{i,j} |(T_{j+1} - T_j) - (T_{i+1} - T_i)|$

Example 3.15 Maximum difference between peak amplitudes

Similarly to the previous example, we define a subformula $\psi'(\text{diff})$ extracting in diff the difference between two left amplitudes of peaks in the trace:

$$\begin{aligned} \psi'(\text{diff}) = & \exists(t, v1, v2, v3, v4, a1, a2) | (v2 - v1 = a1 \wedge v4 - v3 = a2 \\ & \wedge (a2 - a1 = \text{diff} | a1 - a2 = \text{diff}) \wedge \text{diff} > 0 \\ & \wedge \mathbf{F}((\frac{dA}{dt} < 0) \wedge \mathbf{X}(\frac{dA}{dt} \geq 0 \wedge A = v1 \wedge \text{Time} = t \\ & \wedge (\frac{dA}{dt} \geq 0) \mathbf{U}(\frac{dA}{dt} < 0 \wedge A = v2))) \\ & \wedge \mathbf{F}((\frac{dA}{dt} < 0) \wedge \mathbf{X}(\frac{dA}{dt} \geq 0 \wedge A = v3 \wedge \text{Time} > t \\ & \wedge (\frac{dA}{dt} \geq 0) \mathbf{U}(\frac{dA}{dt} < 0 \wedge A = v4)))) \end{aligned}$$

- FO-LTL(\mathbb{R}_{lin}) formula: $\phi_{12}(\text{diff}) = \psi'(\text{diff}) \wedge \neg \exists \text{diff2} | (\text{diff2} > \text{diff} \wedge \psi'(\text{diff2}))$
- *Validity domain* $\mathcal{D} = \left[\max_{(i,j) \in [1,n]^2} |a_i - a_j|, +\infty \right]$ where $\{a_i\}$ is the set of left amplitudes of the local maxima for the trace of A, defined as the difference between the local maximum value and the preceding local minimum value.

Below, a variant of $\phi_{12}(\text{diff})$ that considers only peaks after 80h and sets a maximal value of 0.02 for the difference in peak amplitudes is applied to the trace in Figure 3.6.

```

biocham: validity_domain(diff<0.02
& Exists([t,m1,m2,a1,a2],(a2-a1=diff | a1-a2=diff) & diff>0
& F((d([X])/dt=<0) & X(d([X])/dt>0 & [X]=m1 & Time=t
& (d([X])/dt>0) U (d([X])/dt=<0 & [X]=m1+a1 & Time>80
& F((d([X])/dt=<0) & X(d([X])/dt>0 & [X]=m2 & Time>t
& (d([X])/dt>0) U (d([X])/dt=<0 & [X]=m2+a2))))))
& !(Exists([t,m1,m2,a1,a2,diff2], diff2>diff
& (a2-a1=diff2 | a1-a2=diff2) & diff2>0
& F((d([X])/dt=<0) & X(d([X])/dt>0 & [X]=m1 & Time=t
& (d([X])/dt>0) U (d([X])/dt=<0 & [X]=m1+a1 & Time>80
& F((d([X])/dt=<0) & X(d([X])/dt>0 & [X]=m2 & Time>t
& (d([X])/dt>0) U (d([X])/dt=<0 & [X]=m2+a2)))))))).
diff = 0.0122592
Time elapsed : 1260 ms

```

Listing 3.5: Computing the maximum difference between peak amplitudes after 80h on the oscillatory trace in Figure 3.6 with Biocham.

3.3 Illustration: parameter analysis of a toy oscillator model

We illustrate the use of complex linear temporal logic formulae on a parameter analysis of a toy oscillator model reproducing the dynamics of a self-inhibited gene. This model is a simple multi-component differential oscillator described in [Novák and Tyson, 2008] and schematized in Figure 3.7. Based on a delayed negative-feedback loop, it is composed of only three molecular species represented with three ODEs with nine kinetic parameters. The molecules X , Y_{cyto} and Y_{nucl} represent the gene mRNA, protein in the cytoplasm and protein in the nucleus, respectively. Y_{nucl} downregulates the expression of X (allosteric binding with Michaelis-Menten kinetics). X activates Y_{cyto} and Y_{cyto} is translocated into Y_{nucl} . All these reactions, along with the degradations, follow mass-action kinetics, except the degradation of Y_{cyto} which follows a Hill function. The transport of the protein from the cytoplasm to the nucleus provides the necessary delay enabling the rise of oscillations. The ODEs and parameter values of this model are given in Appendix A.3.

We aim at investigating the effect of the parameter kdx , the kinetic rate for the degradation of the mRNA X , on the dynamics of the model.

As shown on Figure 3.8, increasing the value of the parameter kdx from the initial value 0.05 first decreases the period of the oscillations while decreasing the amplitude ($kdx = 0.2$), then dampen the oscillations ($kdx = 0.25$, $kdx = 0.4$), and finally destabilizes the oscillations which disappear for $kdx = 0.5$.

We use four temporal logic formulae to compute validity domains corresponding to relevant measures characteristic of the oscillations of the molecule X :

- The amplitude of the oscillations of X , computed with $\phi_3(\text{ampl})$. The result is a single value for the variable *ampl*.
- All peak-to-peak intervals in the trace of X , computed with $\phi_8(\text{dist})$. The result is a set of values for the variable *dist*.
- A better estimation for the period as the mean between the last two peak-to-peak intervals. It is computed with $\phi_9(\text{period})$ and stored in the variable *period*.
- As a measure of the regularity of the oscillations, the greatest difference between two peak amplitudes, computed with the formula $\phi_{12}(\text{diffampl})$ and stored in the variable *diffampl*.

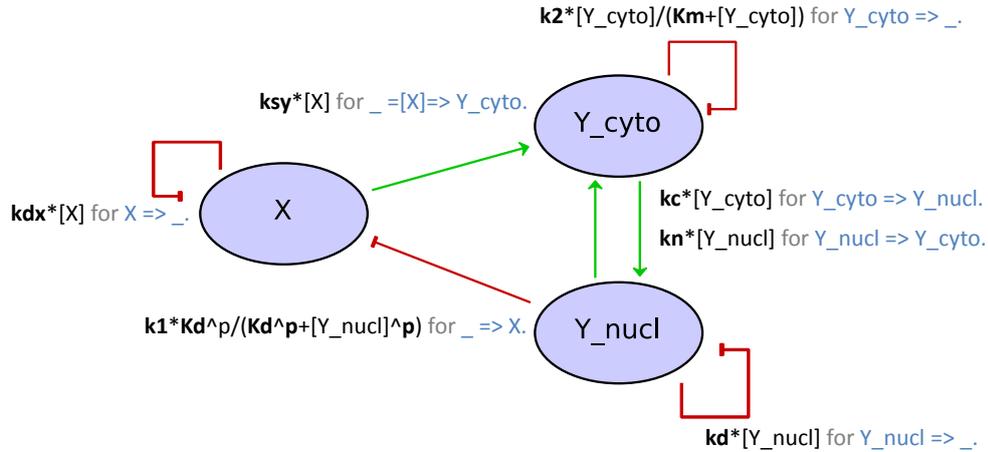


Figure 3.7: Schema of the toy oscillator model taken from [Novák and Tyson, 2008]. Green arrows denote positive interactions, red arrows represent negative interactions. All regulatory rules are displayed close to the corresponding reactions, written in Biocham syntax: in black is the kinetic rate and in blue the corresponding reaction. The corresponding ODEs and parameter values are detailed in Appendix A.3.

We implemented a tool in Biocham that computes in parallel the validity domains of one or several temporal logic specifications for a set of values for a given parameter (see the command `domains_parameter` in Biocham’s reference manual [Fages et al., 2014]). We use it to compute the validity domains of $\phi_3(\text{ampl})$, $\phi_8(\text{dist})$, $\phi_9(\text{period})$ and $\phi_{12}(\text{diffampl})$ for 100 simulations with a time horizon of 200h, for values of kdx taken evenly in the interval $[0, 0.5]$. The result is a response curve for each characteristic measure, shown in Figure 3.9.

The response curves show that increasing the value of the parameter kdx in the interval $[0.05, 0.3]$ decreases the period of the oscillator and decreases its amplitude. The variation intervals between values of dist for the simulations where $kdx > 0.3$, as well as the increasing value of diffampl indicate the destabilization of the oscillations.

How best to evaluate these traces, in order to score their satisfaction to the objective: regular 24h-periodic oscillations?

In order to address this question, we compare four FO-LTL formulae ψ_1, ψ_2, ψ_3 and ψ_4 , obtained by combinations of the previously-defined temporal logic patterns and detailed below with objective values. Each formula computes an approximation of the period on the trace, that we call pseudo-period, with potential additional measures characterizing the regularity of the oscillations.

The evaluated violation degrees of these formulae on the traces in Figure 3.8 with respect to the objective values are summarized in Table 3.2. Moreover, we use the satisfaction degree of these formulae to visualize the joint effect of kdx and another important parameter for the dynamics of X : its synthesis kinetic rate $k1$, as shown in Figure 3.10.

- $\psi_1(\text{period}) = \phi_8(\text{period})$
Objective: 24

ψ_1 merely finds all the values of the peak-to-peak intervals in the trace. The violation degree of this formula is the smallest distance between the objective value (24h) and each peak-to-peak interval in the trace. One can notice in Table 3.2 that the violation degrees of the three first traces ($kdx = 0.05$, $kdx = 0.2$, $kdx=0.25$) correctly reflect the decreasing period and the decreasing satisfaction of the 24h-objective. However, this formula is clearly simplistic as it estimates the last three traces to be closer to satisfaction, without taking notice of their damped oscillations.

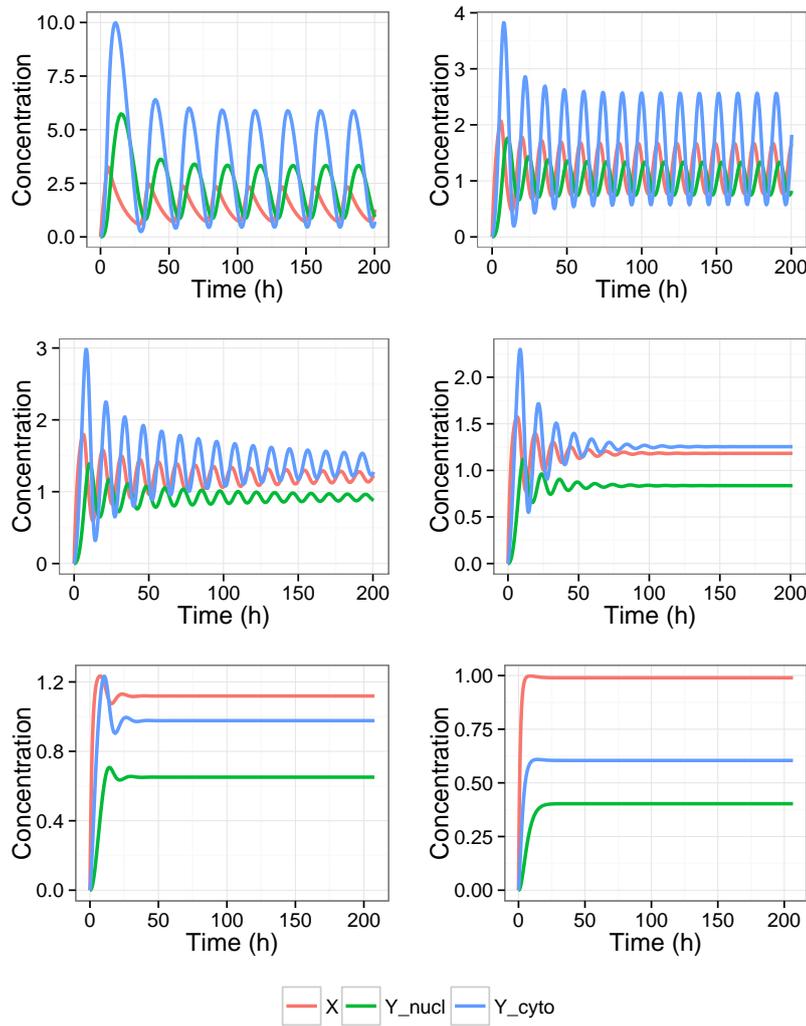


Figure 3.8: Traces of the toy oscillator model defined in Appendix A.3, simulated with Biocham for different values for the parameter kdx (from top left to bottom right: 0.05,0.2,0.25,0.3,0.4 and 0.5). Each simulation starts from null initial values.

The first panel (top left) in Figure 3.10 represents the landscape of satisfaction degrees computed with ψ_1 . Several domains of satisfaction, where the satisfaction degree is close to 1, appear in yellow on the landscape. Some of them correspond to incorrect measurement of the pseudo-period. For instance, the simulation with $k1 = 1.1$ and $kdx = 0.1$ is plotted in Figure 3.11.

The validity domain of $\psi_1(\text{period})$, as well as its satisfaction degree, both computed with Biocham (Listing 3.6), reveal that the objective 24 is close to satisfaction because the first peak-to-peak interval, quite different from the other intervals in the trace, is close to 24h. The same reason causes the small violation degree (0.72) in Table 3.2 for $kdx = 0.4$.

```

biocham: validity_domain(Exists([tmax1,tmax2],(tmax2-tmax1=period)&F((d([X])/
dt>=0)&X((d([X])/dt<0)&(Time=tmax1)&((d([X])/dt<0) U ((d([X])/dt>=0)&((d([X]
)/dt>=0) U ((d([X])/dt<0)&(Time=tmax2)))))).
period = 24.0879
|
period = 19.0456
|
period = 18.662
    
```

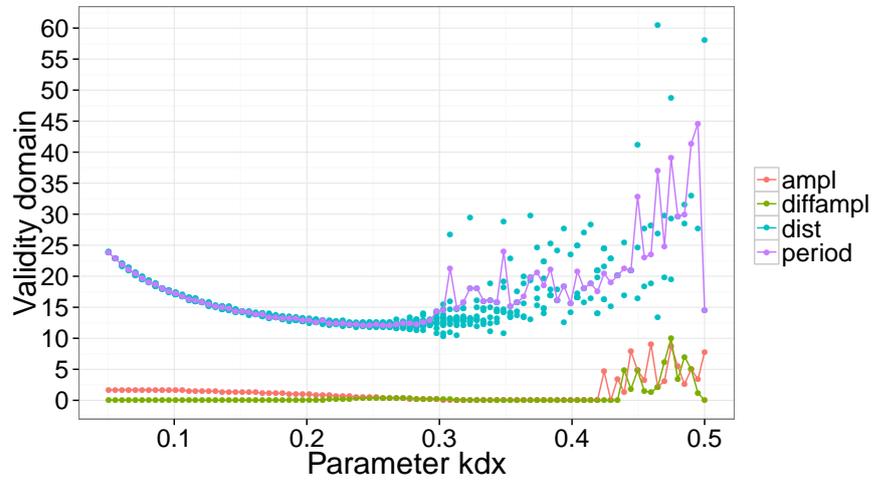


Figure 3.9: Values of the validity domains for the variables *ampl*, *diffampl*, *dist* and *period* computed in 100 simulations with varying values of the parameter *kdx*.

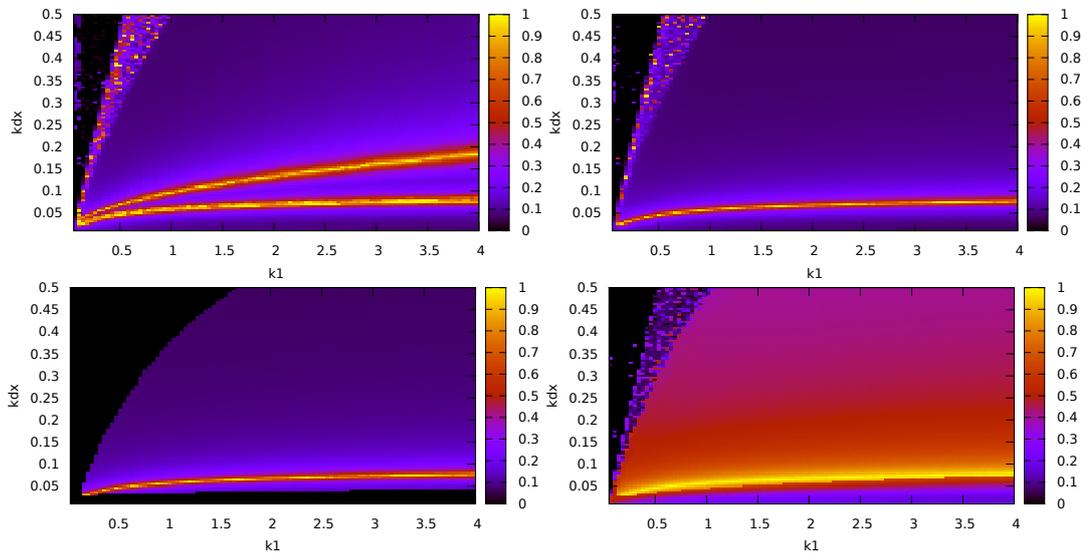


Figure 3.10: Landscapes displaying the satisfaction degree for the formulae ψ_1 (top left), ψ_2 (top right), ψ_3 (bottom left) and ψ_4 (bottom right), evaluated on 100x100 simulations of the toy oscillator model with varying values of the parameters k_1 and kdx and a 200h-time horizon.

kdx	0.05	0.2	0.25	0.3	0.4	0.5
ψ_1	0.056	9.59	11.21	6.56	0.72	9.47
ψ_2	0.096	10.99	11.66	9.52	8.48	9.47
ψ_3	0.096	10.99	Inf	Inf	Inf	Inf
ψ_4	2.00	2.28	4.06	8.97	15.67	159.83

Table 3.2: Violation degree for the formulae ψ_1 , ψ_2 , ψ_3 and ψ_4 on simulated traces of the toy oscillator model with different values for the parameter kdx .

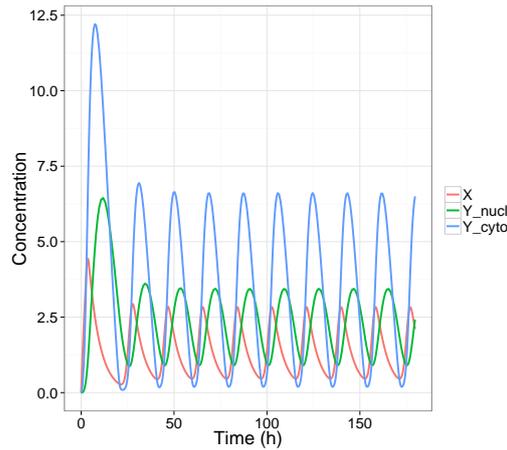


Figure 3.11: Simulation of the toy oscillator model with $k1 = 1.1$ and $kdx = 0.1$.

```

|
period = 18.6566
|
...
|
period = 18.7143
Time elapsed : 544 ms

biocham: satisfaction_degree(Exists([tmax1,tmax2], (tmax2-tmax1=period)&F((d([X
    ])/dt>=0)&X((d([X])/dt<0)&(Time=tmax1)&((d([X])/dt<0) U ((d([X])/dt>=0)&((d
    ([X])/dt>=0) U ((d([X])/dt<0)&(Time=tmax2)))))), [period], [24], 200).
Simulation time: 0.66s
Violation degree 0.0878529
Satisfaction degree 0.919242
Time elapsed 1.13 s
    
```

Listing 3.6: Validity domain and satisfaction degree for ψ_1 computed with Biocham on the toy oscillator model defined in Appendix A.3 simulated with $k1 = 1.1$ and $kdx = 0.1$.

The rest of the landscape is divided in purple and black regions. The region in purple correspond to low satisfaction degrees: for these parameter values the simulation traces exhibit oscillations but their period is different from 24h. In the region in black, corresponding to small values of $k1$ and high values of kdx , the satisfaction degree is null. It indicates that simulated traces contain less than three peaks, such that oscillations can not be detected.

- $\psi_2(\text{period}) = \phi_9(\text{period})$
Objective: 24

ψ_2 uses the mean between the last two peak-to-peak intervals to estimate the period. Hence, any incorrect value due to the transitory irregularities after the initial values are avoided. For this reason, the violation degree in Table 3.2 for $kdx = 0.4$ is greater than for ψ_1 . In the same manner, the spurious satisfaction region on the landscape for ψ_1 landscape in Figure 3.10 is filtered out for ψ_2 . For instance, the satisfaction degree for the trace in Figure 3.11 becomes 0.15 for ψ_2 .

Although the landscape is more precise with ψ_2 , some simulations remain incorrectly scored. For example, the evaluated satisfaction degree for the simulation with $k1 = 0.52$ and $kdx = 0.46$ is 0.97. The corresponding trace is plotted in Figure 3.12. The trace appears to contain only two peaks at most, however more peaks are detectable with very small amplitudes, as seen in Listing 3.7. In this listing the validity domains of three variables are computed: *dist* takes the values of peak-to-peak intervals (encoded by ϕ_8), *period* is the mean of the last two intervals (encoded by ϕ_9), and *ampl* extracts the left amplitudes of all peaks (encoded by ϕ_6). In particular the value of *period* is the result of ψ_2 and is close to 24h, hence the high satisfaction degree.

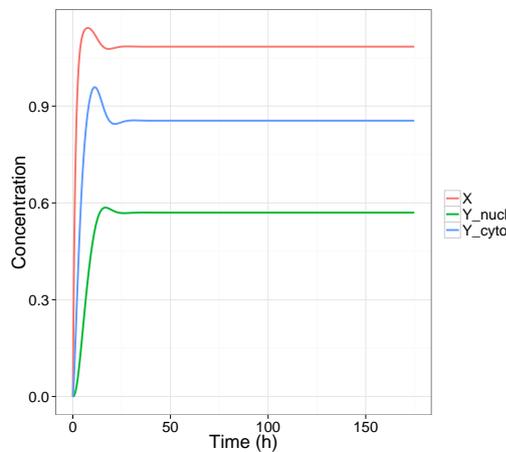


Figure 3.12: Simulation of the toy oscillator model with $k1 = 0.52$ and $kdx = 0.46$.

```

biocham: validity_domain(Exists([tmax1,tmax2],(tmax2-tmax1=dist)&F((d([X])/dt
  >=0)&X((d([X])/dt<0)&(Time=tmax1)&((d([X])/dt<0) U ((d([X])/dt>=0)&((d([X])
  /dt>=0) U ((d([X])/dt<0)&(Time=tmax2))))))).
dist = 27.4605
|
dist = 20.5953
|
dist = 20.5953
|
dist = 22.9764
|
dist = 21.7027
Time elapsed : 40 ms

biocham: validity_domain(Exists([tmax],Exists([p1,p2,t1,t2],p1 = t2-t1 & p2 =
  tmax - t2 & 2*period=p1+p2& F((d([X])/dt>=0) & X(d([X])/dt<0 & Time=t1 & (d
  ([X])/dt<0) U (d([X])/dt>=0 & (d([X])/dt>=0) U (d([X])/dt<0 & Time=t2 & (d
  ([X])/dt<0) U (d([X])/dt>=0 & (d([X])/dt>=0) U (d([X])/dt<0 & Time=tmax)))
  ))) & !(Exists([t],F((d([X])/dt>=0) & X(d([X])/dt<0 & Time=t) & t>tmax)))
.
period = 24.0279
Time elapsed : 188 ms

biocham: validity_domain(Exists([m1],F(d([X])/dt< 0 & X( d([X])/dt >= 0 & [X]=
  m1 & (d([X])/dt >= 0) U (d([X])/dt < 0 & [X]=m1+ampl))))).

```

```

ampl = 1.04361e-14
|
...
|
ampl = 2.50334e-06
|
ampl = 0.000137424
|
ampl = 0.00846336
|
ampl = 1.13563
Time elapsed : 28 ms
    
```

Listing 3.7: Validity domains of several temporal logic patterns computed with Biocham on the toy oscillator model defined in Appendix A.3 simulated with $k_1 = 0.52$ and $k_{dx} = 0.46$

- $\psi_3(\text{period}) = \exists(\text{maxpeak}, \text{maxdiff}, \text{diffd})|$
 $\phi_9(\text{period}) \wedge \phi_7(\text{maxpeak}, 40) \wedge \phi_{11}(\text{maxdiff}, 40) \wedge \phi_{12}(\text{diffd}, 40)$
 $\wedge 4 * \text{diffd} < \text{period}$
 $\wedge 5 * \text{maxdiff} < \text{maxpeak}$
 $\wedge \text{maxpeak} > 0.05$

Objective: 24

The formula ψ_3 combines the formula ϕ_9 , which estimates the period with the variable period , with a variant of the formulae ϕ_7 which stores in the variable maxpeak the greatest left amplitude of the peaks occurring after the first 40h in the trace, and variants of the formulae ϕ_{11} and ϕ_{12} which compute in the variables maxdiff and diffd the greatest difference between, respectively, the peak amplitudes and the peak-to-peak intervals in the variable, for peaks occurring after the first 40h.

These additional variables are not associated with objective values. Instead fixed constraints are defined for these variables with comparison terms. For instance, the maximum difference between two peak-to-peak intervals (diffd) have to remain lower than a quarter of the pseudo-period (period). If any of these terms fails to be satisfied, the violation degree becomes $+\infty$. This way, irregular oscillations (for examples for $k_{dx} > 0.2$ in Table 3.2) are assigned a null satisfaction degree and are filtered out. It is easily visible on the corresponding landscape in Figure 3.10 (bottom left): some areas in the landscape become black, indicating irregular oscillations.

- $\psi_4(\text{dist}, \text{diff1}, \text{diff2}, \text{ampl}) = \exists(\text{maxpeak}, \text{maxdiff}, \text{diffd}, \text{period})|$
 $\phi_9(\text{period}) \wedge \phi_7(\text{maxpeak}, 40) \wedge \phi_{11}(\text{maxdiff}, 40) \wedge \phi_{12}(\text{diffd}, 40)$
 $\wedge \text{dist} = (24 - \text{period})/10$
 $\wedge 4 * \text{diffd} - \text{period} < \text{diff1}$
 $\wedge 10 * \text{maxdiff} - \text{maxpeak} < \text{diff2}$
 $\wedge 20 * (0.1 - \text{maxpeak}) < \text{ampl}$

Objectives: 0, 0, 0, 0

The formula $\psi_4(\text{dist}, \text{diff1}, \text{diff2}, \text{ampl})$ is a variant of $\psi_3(\text{period})$ where, instead of setting fixed constraints for diffd , maxpeak and maxdiff , additional variables are defined to measure quantitative errors associated with these variables. These errors, stored in diff1 , diff2 and ampl , are null if the constraints on diffd , maxpeak and maxdiff are satisfied, and positive otherwise. The coefficients defining diff1 , diff2 and ampl are chosen in order to keep error values comparable.

Similarly, the variable period is replaced by dist , which measures a smaller error on the period than the distance between period and the objective of 24h. The violation degree is the euclidean distance between the evaluated errors, a single point domain in a four-dimensional space, and the null error point $(0, 0, 0, 0)$. The satisfaction degrees displayed on the landscape in Figure 3.10 result in a greater contrast and allow to distinguish several domains.

The region at the bottom, in shades of yellow to pink, characterize regular oscillations. The color gradient indicates a period increasingly far from the objective 24h when the value of kdx increases. The region above, in purple, characterizes irregular oscillations that are assigned a positive but smaller satisfaction degree.

While ψ_3 , by filtering out irregular oscillations, provides a more precise visualization of regular oscillation domains in the parameter space, ψ_4 ensures a continuous and gradual violation degree which is a more adequate for parameter calibration. Indeed, as one can see in Table 3.2, regular oscillations with the wrong period have a better score than damped oscillations (see $kdx = 0.2$ with contrast with $kdx = 0.25$), and damped oscillations have a better score than simulations without oscillations (see $kdx = 0.25$ compared to $kdx = 0.5$). This score gradient is useful to guide the parameter search toward the regions of regular oscillations.

3.4 Conclusion

Temporal logic provides a powerful language for describing the important properties of the dynamical behavior of a biological systems. While curve fitting methods assume a precise specification of the dynamics given by a complete curve, it is possible with temporal logic formulae to specify imprecise behaviors in a semi-qualitative semi-quantitative fashion.

The full first-order setting presented in this chapter makes it possible to extract interesting information from numerical traces, e.g. on periods and phases of irregular oscillations, through the computation of validity domains for the free variables of the formulae. In addition, these validity domains can be used to define a continuous satisfaction degree in the interval $[0, 1]$ for temporal logic formulae, a score which can support parameter analysis of which can be combined with powerful continuous optimization algorithms to search parameter values in high dimension.

In the next chapter we present dedicated solvers which speed-up the computation of validity domains of the formula patterns presented here.

4. Efficient solving of FO-LTL(\mathbb{R}_{lin}) constraints

In the previous chapter, we have shown how First-Order Linear Time Logic (FO-LTL) can be used for verifying complex dynamical properties on one or several simulated or experimental traces. This chapter tackles the efficient use of temporal logic specifications.

Two aspects have to be addressed. First, FO-LTL formulae describing complex behaviors are very long and writing them is error-prone. Moreover, the computation cost of solving such verification problem has to be as small as possible, since each verification possibly has to be evaluated hundreds or thousands of times in a calibration procedure.

Therefore we explore ways of improving the syntax and the solving efficiency of temporal logic specifications.

4.1 Solving complexity

Constraints used in FO-LTL(\mathbb{R}_{lin}) formulae are often bound constraints, i.e. constraints of the form $x \leq c$ or $x \geq c$ where x is a variable and c a constant. They define Cartesian products of intervals, which are a particular kind of polyhedra named boxes. In that case, the validity domains are finite union domains of boxes, since they are obtained by intersection, union, complementation and projection of boxes.

However, it is worth noticing that even in the case of bound constraints, the validity domain of a temporal formula can contain an exponential number of polyhedra [Fages and Rizk, 2008]. Let us define the *size* of a finite union of boxes \mathcal{D} , as the greatest integer k such that $\mathcal{D} = \bigcup_{i=1}^k \mathcal{R}_i$, where the \mathcal{R}_i 's are boxes.

Proposition 4.1 [Fages and Rizk, 2008] *On a trace of length n , the validity domain of a FO-LTL(\mathbb{R}_{lin}) formula of size f containing v variables and only bound constraints, is a union of boxes of size less than $(n.f)^{2v}$.*

Proof 4.2 *Let us consider the number of possible bounds appearing in the validity domain \mathcal{D}_ϕ of a formula ϕ for a given variable x .*

Let us first consider the case where ϕ is a bound constraint $x \leq c$ or $c \leq x$. Such a constraint is evaluated on each time point of the trace, by creating at most n different bounds for x . Hence the maximum number of bounds for variable x is n times the number of occurrences of x in ϕ which is less or equal to

$n \times f$. Note that this maximum number of bounds is reached for the formula $F([A] = u \vee [A] + 1 = u \vee \dots \vee [A] + f = u)$ for instance.

Now, the validity domains for the logical connectives, quantifiers and temporal operators are defined by union, intersection, projection and complementation operations, which do not create new bound values for the variables.

As a box is a Cartesian product of intervals, it is defined by two bounds for each variable. With less than $n \times f$ bounds per variable, one can thus form at most $(nf)^{2v}$ boxes. Therefore, the solution domain computed by the algorithm is a union of boxes of size less than $(nf)^{2v}$.

As for the tightness of these bounds, note that the following formula

$$F([A_1] = X_1 \vee [A_1] + 1 = X_1 \vee \dots \vee [A_1] + f = X_1) \wedge \dots \\ \wedge F([A_v] = X_v \vee [A_v] + 1 = X_v \vee \dots \vee [A_v] + f = X_v)$$

has a solution domain of size $(nf)^v$ on a trace of n values for the $[A_i]$'s such that the values $[A_i] + k$ are all different for $1 \leq i \leq v$ and $0 \leq k \leq f$.

In practice, the systematic use of the projection operation on the subdomain computed for a subformula, in order to remove unnecessary variables for the rest of the formula dimensions, ensures that the number of variables remains as low as possible. However, when formulae require many variables to transcribe complex behaviors, the exponential cost of evaluating the validity domains induces long or even intractable times.

4.2 Dedicated solvers for complex specifications

Defining and implementing specific functions to replace common complex patterns is useful for achieving two goals:

- provide user-friendly macros to define a dynamical behavior for biologists who are not familiar with temporal logic,
- assign to each pattern a more efficient solver than the generic solver, in order to speed up the computation of validity domains and parameter search.

Temporal Operator Patterns

The first goal has been discussed in [Monteiro et al., 2008], who provide a list of proposed temporal operator macros defined for operators that apply to one or several temporal logic formulae, which we recall in Table 4.1.

This first step is however limited to simple temporal specifications.

Behavior	Formula	Macro
Occurrence of ϕ	$\mathbf{F}(\phi)$	$\text{Occurs}(\phi)$
Exclusion	$\mathbf{G}(\neg\phi)$	$\text{Excludes}(\phi)$
Invariance	$\mathbf{G}(\phi)$	$\text{Invariates}(\phi)$
Sequence: ϕ occurs before ψ	$\mathbf{F}(\phi \wedge \mathbf{F}(\psi))$	$\text{WeakSequence}(\phi, \psi)$
Sequence: ϕ immediately before ψ	$\mathbf{F}(\phi \wedge \mathbf{X}(\psi))$	$\text{ExactSequence}(\phi, \psi)$
Sequence: ϕ always occurs until ψ	$\mathbf{G}(\phi \mathbf{U}(\psi))$	$\text{Sequence}(\phi, \psi)$
Consequence: if ϕ then later ψ	$\mathbf{G}(\phi \Rightarrow \mathbf{F}(\psi))$	$\text{Consequence}(\phi, \psi)$
Implication: ϕ implies ψ at the same time	$\mathbf{G}(\phi \Rightarrow \psi)$	$\text{Implication}(\phi, \psi)$

Table 4.1: A list of macros proposed in [Monteiro et al., 2008] to replace some common operators for temporal logic formulae

Temporal Logic Patterns

In the rest of this section, we generalize this approach to propose a list of relations (or predicates), which specify some important behavior constraints by defining temporal logic relations between a set of species of the model and a set of variables. To address the solving complexity issue, each relation, defined as a user-friendly macro which can be used instead of an FO-LTL formula or subformula, is associated with a dedicated solver implemented in Biocham.

We also relate each predicate to the equivalent FO-LTL formula, some of which have already been introduced in Chapter 3. This allows us to compare the solving time with dedicated solvers and with the generic solver. We first begin by commenting a few examples of relations and comparing how the dedicated solver operate compared with the generic solver. We also show how the defined relations can be combined inside an FO-LTL formula with the usual logical operators to specify more complex behaviors.

Finally, we summarize in Table 4.2 to Table 4.5 the full list of specific temporal logic relations, implemented in Biocham as macros and associated with dedicated solvers, and provide the equivalent FO-LTL(\mathbb{R}_{lin}).

Defining validity domains with single values

A great advantage of temporal logic is its generic nature, that allows to express and evaluate any dynamical property with a unique tool. By defining specific predicated associated with dedicated solvers, we choose to sacrifice generality in an attempt to improve evaluation performance. The choice of each predicate becomes a trade-off between specificity and generality: the evaluation will be all the more efficient as it is specific, but at the same time it will only be usable in specific cases. On the other hand, allowing the combination of semi-specific predicates widens the expressivity and the number of cases where they can be useful.

This trade-off between generality and specificity in the choice of implementation can be illustrated by going back to the reachability constraint $\phi_1 = \mathbf{F}(A \geq v)$ defined in Example 3.2 in Chapter 3, whose validity domain is $D =]-\infty, \max A]$. Replacing this pattern with a specific predicate could be done by defining:

- A macro **reached(A,v)**
- A validity domain: $\mathcal{D} =]-\infty, \max A]$ directly computed from the trace with a specific solver.
- The violation degree $vd = \max(0, v* - \max A)$ and robustness $ro = \max(0, \max A - v*)$ could also be directly computed from the trace with specific predicates, given the objective value $v*$. This would be more efficient than computing first the validity domain and then its distance to the objective in the cases when the validity domain is not needed as well.

Such a specific implementation reduces the capacity for combinations of validity domains, and thereby the use of each defined relation. It would therefore require a long list of relations to cover most frequent specifications. For example, the dual function to $reached(A,v)$, satisfied when the threshold is not reached, could be useful: $unreached(A,v)$ with the complementary Validity domain $\mathcal{D} = [maxA, +\infty[$.

Instead, we choose a more generic implementation, that computes validity domains directly from the trace but still relies on computing a distance to obtain the violation and satisfaction degrees, but returns results that can be easily combined. For reachability specifications, we define the following relation $max(molecule,value)$:

- **Macro: $max(A,v)$.**
- Equivalent FO-LTL(\mathbb{R}_{lin}) formula: $\mathbf{G}(A \leq v) \wedge \mathbf{F}(A = v)$.
- Dedicated validity domain: $\mathcal{D} = \{maxA\}$, computed directly on the trace with algorithm 2 where A_{s_i} the value of A at the state s_i in the trace (s_0, \dots, s_n) and $maxA$ is the maximum value of A on the trace.

Algorithm 2 Dedicated solver for $max(A,v)$

```

v ← As0
for i ∈ [1, n] do
  if Asi > v then
    v ← Asi
  end if
end for

```

The reachability property can then be deduced with the following formula that combines the predicate $max(molecule,value)$ with an additional variable:

$$\phi_{13} = \exists m | (max(A, m) \wedge m \geq v)$$

And the unreachability property uses the same function:

$$\phi_{14} = \exists m | (max(A, m) \wedge m \leq v)$$

In Listing 4.1, the validity domain of the relation $max(molecule,value)$ is computed on the trace of the molecule X in the toy oscillator model introduced in Chapter 3 simulated with a time horizon of 400h. In Listing 4.2, the relation $max(molecule,value)$ is combined to specify a reachability property, and the satisfaction degree is computed for two objective values: 3 and 10. In the first case, the satisfaction degree is equal to 1 because the property is satisfied.

```

biocham: validity_domain(max([X],[v])).
v = 3.26102
Time elapsed : 4 ms

```

Listing 4.1: Evaluating with Biocham the validity domain of the relation max , computed with a dedicated solver, on the model defined in Appendix A.3 simulated with a 400h time horizon

```

biocham: satisfaction_degree(Exists([m],max([X],[m]) & m >= v),[v],[3],400).
Simulation time: 0.52s
Violation degree 0
Satisfaction degree 1
Time elapsed 0.56 s
biocham: satisfaction_degree(Exists([m],max([X],[m]) & m >= v),[v],[10],400).
Simulation time: 0.52s
Violation degree 6.73898
Satisfaction degree 0.129216

```

4. EFFICIENT SOLVING OF FO-LTL(\mathbb{R}_{LIN}) CONSTRAINTS

Time elapsed 0.56 s

Listing 4.2: Evaluating with Biocham the satisfaction degrees of a reachability property with objective 3 or 10, computed with a dedicated solver for the relation *max*, on the model defined in Appendix A.3 simulated with a 400h time horizon.

Example 4.3 Normalization of violation degree Combining specific functions with additional variables is useful not only to change the shape of the validity domain, but also to normalize the violation degree of the formula with respect to the objective value. For example, the violation degree of ϕ_{14} , evaluated on the trace of *A*, which reaches its maximum *maxA*, with respect to the objective value of 10, is: $vd = 10 - \text{max}A$. In comparison, the violation degree of the following formula will be in the interval $[0, 1]$, as shown in Listing 4.3, and easily comparable to other violation degrees:

$$\exists(m) | (\text{max}(A, m) \wedge m/10 - 1 \geq v)$$

Different normalizations can be chosen for specifications grouped in a combination to put more weight on some properties compared to the others.

```
biocham: satisfaction_degree(Exists([m],max([X],[m]) & (m-10>=val*10))
, [val], [0], 400).
Simulation time: 0.52s
Violation 0.673898
Satisfaction degree 0.597408
Time elapsed 0.56 s
```

Listing 4.3: Normalize reachability with objective 10

The pattern *amplitude(molecule,value)* specifies the amplitude of the concentration of a molecule. It is equivalent to the formula ϕ_3 in Chapter 3.

- **Macro: amplitude(A,a).**
- Equivalent FO-LTL(\mathbb{R}_{lin}) formula: $\exists(v) | (\mathbf{G}(A \geq v \wedge A \leq v + a) \wedge \mathbf{F}(A = v) \wedge \mathbf{F}(A = v + a))$
- Dedicated validity domain: $\mathcal{D} = \{\text{ampl}A\}$, computed with algorithm 3, where $\text{ampl}A = \text{max}A - \text{min}A$, with *minA* the minimum value of *A* on the trace.

Algorithm 3 Dedicated solver for **amplitude(A,a)**

```
max ← As0
min ← As0
for i ∈ [1, n] do
  if Asi > max then
    max ← Asi
  end if
  if Asi < min then
    min ← Asi
  end if
end for
a ← max - min
```

The equivalent FO-LTL(\mathbb{R}_{lin}) formula needs two variables to define the amplitude: the amplitude *a* and the minimal value of the molecule *v*. Therefore the validity domain computed with the generic solver has two dimensions, and finding the possible values for the amplitude requires a projection on the variable *a*, as shown on the figure 4.1. The dedicated solver directly computes the one-dimension validity domain in one run.

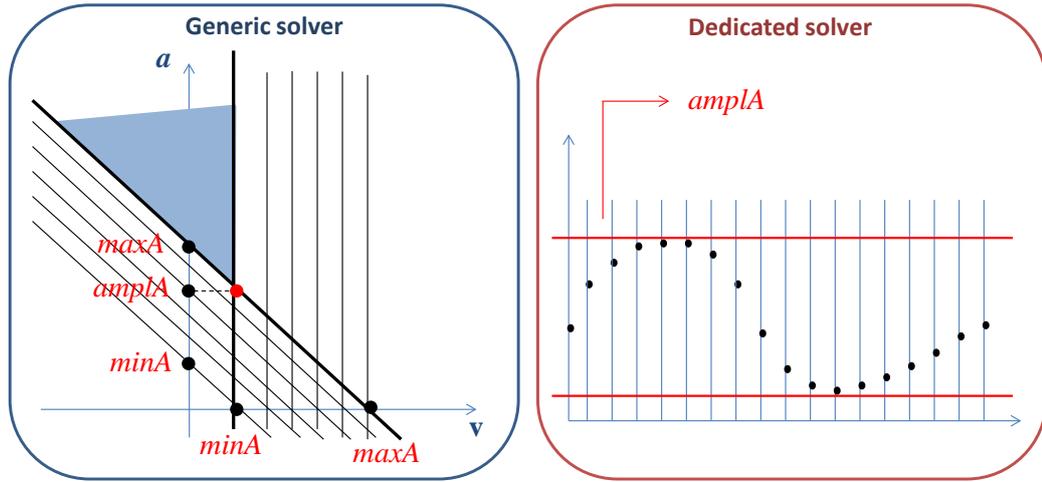


Figure 4.1: Comparing the validity domain solving procedures for $\text{amplitude}(A,a)$. On the left, the generic solver computes a combination of two-dimensional subdomains: the subdomain in blue for the subformula $\mathbf{G}(A \geq v \wedge A \leq v + a)$, and the lines in black for the subformulae $\mathbf{F}(A = v)$ and $\mathbf{F}(A = v + a)$, corresponding to the values of A in the trace. The single point $(\text{min}A, \text{ampl}A)$ at the intersection, in red, is projected on the one-dimensional a -axis with the operator \exists . On the right, the specific solver associated with the function $\text{amplitude}(A,a)$ computes directly the amplitude on the trace.

Variants of a relation

Some functions corresponding to dynamical behaviors are defined with several versions corresponding to different amount of information related to the specification. For example, rather than only specifying constraints on the maximum value of a trace, it can sometimes be useful to constrain as well the time point where this maximum is reached. We therefore define a variant of the function max previously introduced, that returns a two-dimensional validity domains containing the time of the maximum as well as its value:

- **Macro:** $\text{max}(A,v,t)$
- Equivalent FO-LTL(\mathbb{R}_{lin}) formula: $\mathbf{G}(A \leq v) \wedge \mathbf{F}(A = v \wedge \text{Time} = t)$
- Dedicated validity domain: $\mathcal{D} = \{(\text{max}A, T\text{max}A)\}$

As another example, the relation $\text{distanceSuccPeaks}(\text{molecule}, \text{variables})$ returns the set of successive peak-to-peak intervals to estimate the pseudo-period of oscillating trace, and supports two different callings. With three variables, it returns each interval value along with the corresponding peak times. The algorithm corresponding to the dedicated solver is detailed in Algorithm 4, and an example of application with Biocham is provided in Listing 4.4. However, since peak times are usually not needed, calling the same relation with one variable executes the same algorithm but returns only the interval values, which prevents the user from projecting the validity domain to reduce its dimension.

- **Macro:** $\text{distanceSuccPeaks}(A,d,t1,t2)$
- Equivalent FO-LTL(\mathbb{R}_{lin}) formula: $t2 - t1 = d$
 $\wedge \mathbf{F}((\frac{dA}{dt} \geq 0) \wedge \mathbf{X}((\frac{dA}{dt} < 0) \wedge (\text{Time} = t1)$
 $\wedge ((\frac{dA}{dt} < 0) \mathbf{U}((\frac{dA}{dt} \geq 0) \wedge$
 $((\frac{dA}{dt} \geq 0) \mathbf{U}((\frac{dA}{dt} < 0) \wedge (\text{Time} = t2))))))$
- Dedicated validity domain: $\mathcal{D} = \{(T_{i+1} - T_i, T_{i+1}, T_i), i \in [1; n - 1]\}$, where $\{T_i\}_n$ is the set of local maxima times of the trace .

Algorithm 4 Dedicated solver for **distanceSuccPeaks(A,d,t1,t2)**

```

D ← ∅
t ← 0
for i ∈ [1, n] do
  if  $\frac{dA_{s_i}}{dt} > 0 \wedge \frac{dA_{s_{i+1}}}{dt} \leq 0$  then
    if t ≠ 0 then
      D ← D ∪ (Timesi - Timet, t, Timesi)
    end if
    t ← Timesi
  end if
end for

```

```

biocham: validity_domain(distanceSuccPeaks([X], [dist, t1, t2])).
t2 = 35.3191, t1 = 5.29792, dist = 30.0212
|
t2 = 60.0201, t1 = 35.3191, dist = 24.701
|
...
|
t2 = 371.486, t1 = 347.54, dist = 23.9457
|
t2 = 395.393, t1 = 371.486, dist = 23.9069
Time elapsed : 32 ms

```

Listing 4.4: Computing successive peak-to-peak intervals on the 400h-horizon simulation of the model in Appendix A.3 with the dedicated solver in Biocham.

- **Macro: distanceSuccPeaks(A,d)**
- Equivalent FO-LTL(\mathbb{R}_{lin}) formula: $\exists(t_1, t_2) | ((t_2 - t_1 = d) \wedge \mathbf{F}((\frac{dA}{dt} \geq 0) \wedge \mathbf{X}((\frac{dA}{dt} < 0) \wedge (Time = t_1) \wedge ((\frac{dA}{dt} < 0) \mathbf{U}((\frac{dA}{dt} \geq 0) \wedge ((\frac{dA}{dt} \geq 0) \mathbf{U}((\frac{dA}{dt} < 0) \wedge (Time = t_2))))))))))$
- Dedicated validity domain: $\mathcal{D} = \{T_{i+1} - T_i, i \in [1; n - 1]\}$, where $\{T_i\}_n$ is the set of local maxima times of the trace .

Combining relations

To compute the validity domain of a formula, the generic solver executes the procedure described in Algorithm 1 in a bottom-up manner: it first computes the validity domain of each subformula on each point of the trace, and deduces the whole validity domain on each point of the trace by combining these subdomains. In contrast, dedicated solvers analyze the trace with specific functions and compute directly the validity domain on the whole trace only. This main difference makes dedicated solvers more efficient but prevents the combination of built-in functions with temporal operators.

We first provide examples of efficient combinations without temporal operators. Then we show how the implementation of dedicated solvers can be adapted to the combination with temporal operators, by computing intermediary results for each point of the trace when it is necessary.

Combination without temporal operators

As already shown in Example 4.3, the defined macros associated with dedicated solvers can easily be combined with the logical operators \wedge , \vee , \exists and \forall since they return validity domains that support the same geometrical operations as validity domains computed with the generic solver.

In particular, the operations of projection used to compute the maximum or minimum value of a set of values returned as a validity domain can be applied as described in Example 3.8. For instance, in Listing 4.5, *distanceSuccPeaks* is combined with the macro *peak*, introduced below and solved with Algorithm 5, to extract the validity domain containing the concentration values of the two successive peaks corresponding to the longest peak-to-peak interval in the trace.

- **Macro: *peak(A,t,v)***
- Equivalent FO-LTL(\mathbb{R}_{lin}) formula: $\mathbf{F}(\frac{dA}{dt} > 0 \wedge \mathbf{X}(\frac{dA}{dt} \leq 0 \wedge \text{Time} = t \wedge A = v))$
- Dedicated validity domain: $D = \bigcup(\{(t', m')\}, \frac{dA}{dt}(t'-dt) > 0 \wedge \frac{dA}{dt}(t') \leq 0 \wedge A(t') = m')$

Algorithm 5 Dedicated solver for *peak(A,t,v)*

```

D ← ∅
for i ∈ [1, n] do
  if  $\frac{dA_{s_i}}{dt} > 0 \wedge \frac{dA_{s_{i+1}}}{dt} \leq 0$  then
    D ← D ∪ (Timesi, Asi)
  end if
end for

```

```

biocham: validity_domain((Exists([t1,t2],
  distanceSuccPeaks([X],[dist,t1,t2])
  & !Exists([dist2],distanceSuccPeaks([X],[dist2]) & dist2>=dist)
  & peak([X],[t1,v1]) & peak([X],[t2,v2]))).
v2 = 2.46786, v1 = 3.26102, dist = 30.0212
Time elapsed : 2364 ms

```

Listing 4.5: Computing the values of the successive peaks corresponding to the longest interval in the trace from Figure 3.6 by combining relations the dedicated solvers in Biocham.

Evaluating the corresponding FO-LTL formula with the generic solver takes 3900ms. This shows that dedicated solvers can bring a significant time gain to compute simple validity domains, but when combining simple validity domains is necessary to obtain a more complex solution, the largest part of the solving time, which is attributed to the combinations of subdomains with geometrical operations, remains unchanged. In particular, the completion operation triggered by $\neg\exists$ operators is the most costly.

This limitation illustrates the trade-off between generality and specificity that we have tried to find, by defining relatively simple dedicated solvers that can be combined together. Indeed only a fully dedicated evaluation method can be really efficient for complex specifications, but would suffer for reusability purpose from its high specificity.

Nonetheless, we can note that the semi-generic method proposed already brings a significant gain, and provides a formula which is much simpler to use than the full FO-LTL formula.

Combination with temporal operator

Combining a temporal logic function with temporal operators (**F**, **G**, **X**, **U** and **W**) can sometimes be useful. For instance, to extract all peak-to-peaks intervals from an oscillating trace, while avoiding potential transitory irregularities due to the initial values, say before 70h in the trace, one could use the following formula:

$$\phi_{15}(dist) = F(\text{Time} > 70 \wedge \text{distanceSuccPeaks}(X,dist))$$

In that case, the dedicated solver associated with *distanceSuccPeaks* has to compute the validity domain on the first time point higher than 70 in a trace. Therefore in the implementation in Biocham, each dedicated solver is computed on each point of the trace if it is used in combination with a temporal operator, and only on the whole trace otherwise. Algorithm 6, rather than

Algorithm 4, is then used to evaluate ϕ_{15} . This amounts to executing Algorithm 4 for each point on the trace in conjunction with $\text{Time} > 70$, and the temporal operator **F** then computes the union of all the computed validity domains.

Algorithm 6 Dedicated solver for ϕ_{15}

```

 $D \leftarrow \emptyset$ 
for  $i \in [n, 1]$  do
  if  $\text{Time}_{s_i} > 70$  then
     $D_{i_1} \leftarrow \Omega$ 
  else
     $D_{i_1} \leftarrow \emptyset$ 
  end if
   $D_{i_2} \leftarrow \emptyset$ 
   $t \leftarrow 0$ 
  for  $j \in [i, n]$  do
    if  $\frac{dA_{s_j}}{dt} > 0 \wedge \frac{dA_{s_{j+1}}}{dt} \leq 0$  then
      if  $t \neq 0$  then
         $D_{i_2} \leftarrow D_{i_2} \cup (\text{Time}_{s_j} - \text{Time}_t, t, \text{Time}_{s_j})$ 
      end if
       $t \leftarrow \text{Time}_{s_j}$ 
    end if
  end for
   $D \leftarrow D \cup (D_{i_1} \cap D_{i_2})$ 
end for

```

The computation time to evaluate ϕ_{15} with Algorithm 4 on the toy oscillator model defined in Appendix A.3, for a 400h-time horizon is 564 ms, while the computing time to evaluate *distanceSuccPeaks*($[X], [dist]$) with Algorithm 6 on the same trace is only 12ms. The difference shows the high cost of combining a relation associated with a dedicated solver with temporal operators. It is even faster to use the generic solver alone, albeit in the same magnitude order: it takes 524ms to evaluate the FO-LTL(\mathbb{R}_{lin}) formula equivalent to $\phi_{15}(dist)$.

Since discarding the first part of the trace is a frequent need when analyzing models with uncertain initial values, an optional third argument has been implemented for dedicated solvers, specifying that the dedicated solver should be computed only once, at the first point in the trace after the value given in argument.

An equivalent formula for $\phi_{15}(dist)$ is therefore *distanceSuccPeaks*($[A], [dist], 70$), which is computed on the same trace in 12ms with Algorithm 7.

Algorithm 7 Dedicated solver for *distanceSuccPeaks*($[A], [dist], 70$)

```

 $D \leftarrow \emptyset$ 
 $t \leftarrow 0$ 
 $i \leftarrow \min_i (\text{Time}_{s_i} > 70)$ 
for  $j \in [i, n]$  do
  if  $\frac{dA_{s_j}}{dt} > 0 \wedge \frac{dA_{s_{j+1}}}{dt} \leq 0$  then
    if  $t \neq 0$  then
       $D \leftarrow D \cup (\text{Time}_{s_j} - \text{Time}_t, t, \text{Time}_{s_j})$ 
    end if
     $t \leftarrow \text{Time}_{s_j}$ 
  end if
end for

```

Relations without dedicated solver

In some cases, it is not possible to define a more efficient dedicated solver.

The pattern *Macro: increasingSwitch*($A, t, v1, v2$) defines a concentration switch: it specifies that concentration of the molecule A should increase from smaller than $v1$ to greater than $v2$ during a time interval of length t .

- **increasingSwitch($A, t, v1, v2$)**
- Equivalent FO-LTL(\mathbb{R}_{lin}) formula: $\exists(t_1, t_2) | \mathbf{G}((Time \leq t_1 \Rightarrow A < v1) \wedge (Time \geq t_2 \Rightarrow A < v2) \wedge v2 > v1 \wedge t_2 - t_1 = t)$.
- Dedicated validity domain: $\mathcal{D} = \{(\{t\} \times [A(t), +\infty[\times]-\infty, A(t+d)], A(t+d) > A(t)\}$, computed with the generic solver.

The validity domain of this pattern is a set of polyhedrons delineating the possible concentration values taken by the molecule and the time duration between the two values. Computing this validity domain requires to compute each possible combination of values that the variables ($t, v1, v2$) can take, which is exactly what the generic solver does. In that case, the function *increasingSwitch* is a simple macro for the equivalent FO-LTL formula.

Replacing formulae for the analysis of the toy oscillator model

In Section 3.3, we used complex FO-LTL(\mathbb{R}_{lin}) formulae to analyze how the dynamical behavior of a toy oscillator model evolves when one or two of its parameters change.

We give here the equivalent formulae using specific relations. The full list of relations is defined with the equivalent FO-LTL(\mathbb{R}_{lin}) formulae in Tables 4.2 to 4.5.

- The macro *peakAmplitude*(a) corresponds to $\phi_6(a)$
- The macro *distanceSuccPeaks*(d) corresponds to $\phi_8(d)$
- The macro *period*($p_1, p_2, t_1, t_2, t_{max}$) corresponds to $\phi_9(p_1, p_2, t_1, t_2, t_{max})$
- The macro *distanceSuccPeaks*(d) applied to two molecules corresponds to $\phi_{10}(d)$
- The macro *maxDiffDistancePeaks*($diff$) corresponds to $\phi_{11}(diff)$
- The macro *maxDiffAmplPeaks*($diff$) corresponds to $\phi_{12}(diff)$
- The macro *periodErrors*($dist, diff1, diff2, ampl$) corresponds to $\psi_4(dist, diff1, diff2, ampl)$

In particular, we define the function *periodErrors* for the complete characterization of a trace oscillations performed with ψ_4 , in order to avoid the computing cost resulting from multiple combinations.

Listings 4.6 and 4.7 compare the solving time for ψ_4 on the 200h-trace of the molecule X from the toy oscillator model with the default value $kdx = 0.05$, with the dedicated solver associated with the function *periodErrors* or with an equivalent combination of other built-in functions.

```
biocham: validity_domain(
  Exists([period],
    periodErrors([X], [period, errordiffdist, errordiffampl, errorampl], 40)
    & 10*errordist=(24-period))).
10*errordist = 0.0961439, errorampl = 0, errordiffampl = 0, errordiffdist = 0
Time elapsed : 56 ms
```

Listing 4.6: Evaluating in Biocham the relation *periodErrors* on the 200h-trace of the molecule X from the toy oscillator model defined in Appendix A.3, with the dedicated solver.

```

biocham: validity_domain(
  Exists([period,maxdiffampl,maxdiffdist,maxampl],period([X],[period])
    & maxDiffAmplPeaks([X],[maxdiffampl],40)
    & maxDiffDistancePeaks([X],[maxdiffdist],40)
    & peakAmplitude([X],[maxampl]) & !(Exists([ampl2],peakAmplitude([X],[ampl2])
      & ampl2>maxampl))
    & errordiffdist > 4*maxdiffdist-period
    & errordiffampl > 10*maxdiffampl-maxampl
    & errorampl > 20*(0.1-maxampl)
    & 10*errordist=(24-period))).
10*errordist = 0.0961439, errorampl > -31.334, 4*errordiffampl > -4.45487, 9*
  errordiffdist > -218.862
Time elapsed : 472 ms

```

Listing 4.7: Evaluating in Biocham a combination of relations equivalent to the relation *periodErrors*, on the 200h-trace of the molecule X from the toy oscillator model defined in Appendix A.3.

Furthermore, the response curves displayed in Figure 3.9 can finally be computed with the relatively simple command shown in Listing 4.8.

```

biocham: domains_parameter(kdx,[(0,0.5)],100,
  {amplitude([X],[ampl],40),
  maxDiffAmplPeaks([X],[diffampl],40),
  distanceSuccPeaks([X],[dist],40),
  period([X],[period])}
  ,[ampl,diffampl,dist,period],200,data_rc_kdx).

```

Listing 4.8: Command generating data for response curves.

This command performs 100 simulations with different values for the kinetic parameter *kdx*, taken uniformly in the interval $[0, 0.5]$, and stores the values of the validity domains of the variables *ampl*, *diffampl*, *dist* and *period* in the file *data_rc_kdx.csv*.

Computing gain

The next section explores a second approach for efficient solving of complex temporal specifications, and provides in Table 4.6 an illustration of the time gain obtained with dedicated solvers, as well as with this second approach. The figures in this table come from the evaluation of ten temporal logic patterns, including the list above, on the trace of the toy oscillator model introduced in Chapter 3, simulated for a 200h or 400h-time horizon.

Macro and behaviour	Equivalent LTL(\mathbb{R}_{lin}) formula
min(A,v): Minimal value $D = \{\text{min}A\}$	$\mathbf{G}(A \geq v) \wedge \mathbf{F}(A = v)$
min(A,v,t): Minimal value and time for the max $D = \{\text{min}A, \text{Tmin}A\}$	$\mathbf{G}(A \geq v) \wedge \mathbf{F}(A = v \wedge \text{Time} = t)$
max(A,v): Maximal value $D = \{\text{max}A\}$	$\mathbf{G}(A \leq v) \wedge \mathbf{F}(A = v)$
max(A,v,t): Maximal value and time for the max $D = \{\text{max}A, \text{Tmax}A\}$	$\mathbf{G}(A \leq v) \wedge \mathbf{F}(A = v \wedge \text{Time} = t)$
amplitude(A,a): Amplitude $D = \{\text{ampl}A\}$	$\exists(v1, v2) (\mathbf{G}(A \geq v1 \wedge \leq v2) \wedge \mathbf{F}(A = v1) \wedge \mathbf{F}(A = v2) \wedge a = v2 - v1)$
peak(A,t): there exists a local maximum at a precise time point $D = \bigcup(\{t'\} / \frac{dA}{dt}(t' - dt) > 0 \wedge \frac{dA}{dt}(t') \leq 0)$	$\mathbf{F}(\frac{dA}{dt} \geq 0 \wedge \mathbf{X}(\frac{dA}{dt} < 0 \wedge \text{Time} = t))$
peak(A,t,v): there exists a local maximum at the time point t with a value v $D = \bigcup((t', v') / \frac{dA}{dt}(t' - dt) \geq 0 \wedge \frac{dA}{dt}(t') < 0) \wedge A(t') = v'$	$\mathbf{F}(\frac{dA}{dt} \geq 0 \wedge \mathbf{X}(\frac{dA}{dt} < 0 \wedge \text{Time} = t \wedge A = v))$
peak(A,t,v,a): there exists a local maximum at the time point t with a value v and an amplitude a $D = \bigcup((t', v') / \frac{dA}{dt}(t' - dt) \geq 0 \wedge \frac{dA}{dt}(t') < 0) \wedge A(t') = v'$	$\exists([\text{min}]) \mathbf{F}((\frac{dA}{dt} < 0) \wedge \mathbf{X}(\frac{dA}{dt} \geq 0 \wedge A = \text{min} \wedge \mathbf{X}((\frac{dA}{dt} \geq 0) \mathbf{U}(\frac{dA}{dt} < 0 \wedge \text{Time} = \text{time} \wedge A = v)))) \wedge v - \text{min} = a$
IncrInterv(A,t1,t2): Interval of increase $D = \{(t1, t2) \in]-\infty; T1[\times]T2; +\infty[/ \forall t \in [T1, T2], \frac{dA}{dt}(t) > 0\}$	$\mathbf{F}(d([A])/dt \leq 0 \wedge \mathbf{X}(d([A])/dt \leq 0 \wedge \text{Time} = t1 \wedge (d([A])/dt \leq 0) \mathbf{U}(d([A])/dt \leq 0 \wedge \text{Time} = t2))$
IncreasingSwitch(A,t,v1,v2): Concentration switch $D = \{(\{t\} \times [A(t); +\infty[\times]-\infty, A(t+d)] / A(t+d) > A(t)\}$ computed with the generic solver	$\exists(t1, t2) \mathbf{G}((\text{Time} \leq t1 \Rightarrow A < v1) \wedge (\text{Time} \geq t2 \Rightarrow A < v2) \wedge v2 > v1 \wedge t2 - t1 = t)$

Table 4.2: Specific relations defined with dedicated solvers, and equivalent FO-LTL(\mathbb{R}_{lin}) formulae

Macro and behaviour	Equivalent FO-LTL(\mathbb{R}_{lin}) formula
DistancePeaks(A,d): There should be two peaks of A distant by d $D = \{d_{i,j} = T_j - T_i, i \in [1; n], j \in [i; n]\}$ where $\{T_i\}$ is the set of local maxima times of the trace	$\begin{aligned} & \exists(t_1, t_2) \mathbf{F}(\frac{dA}{dt} \geq 0 \\ & \wedge \mathbf{X}(\frac{dA}{dt} < 0 \wedge \text{Time} = t_1 \\ & \wedge \mathbf{X}(\mathbf{F}(\frac{dA}{dt} \geq 0 \\ & \wedge \mathbf{X}(\frac{dA}{dt} < 0 \wedge \text{Time} = t_2 \\ & \wedge (t_2 > t_1) \wedge (t_2 - t_1 = d)))))) \end{aligned}$
DistancePeaks(A,B,d): There should be a peak of A and a peak of B distant by d $D = \{T_{A_j} - T_{B_i}\}$ with $i \in [1; n], j \in [i; n]$ where $\{T_{A_i}\}$ is the set of local maxima times for the trace of A and $\{T_{B_i}\}$ is the set of local maxima times for the trace of B	$\begin{aligned} & \exists(t_1, t_2) \mathbf{F}(\frac{dA}{dt} \geq 0 \\ & \wedge \mathbf{X}(\frac{dA}{dt} < 0 \wedge \text{Time} = t_1)) \\ & \wedge \mathbf{F}(\frac{dB}{dt} \geq 0 \\ & \wedge \mathbf{X}(\frac{dB}{dt} < 0 \wedge \text{Time} = t_2 \\ & \wedge (t_2 - t_1 = d))) \end{aligned}$
DistanceSuccPeaks(A,d): There should be two successive peaks of A distant by d $D = \{T_{i+1} - T_i, i \in [1; n-1]\}$ where $\{T_i\}$ is the set of local maxima times of the trace	$\begin{aligned} & \exists(t_1, t_2) (t_2 - t_1 = d) \\ & \wedge \mathbf{F}((\frac{dA}{dt} \geq 0) \wedge \mathbf{X}((\frac{dA}{dt} < 0) \wedge (\text{Time} = t_1) \\ & \wedge ((\frac{dA}{dt} < 0) \mathbf{U}((\frac{dA}{dt} \geq 0) \wedge \\ & ((\frac{dA}{dt} \geq 0) \mathbf{U}((\frac{dA}{dt} < 0) \wedge (\text{Time} = t_2)))))))) \end{aligned}$
DistanceSuccPeaks(A,d,t1,t2): There should be two successive peaks of A at times t_1 and t_2 distant by d $D = \{(T_i, T_{i+1}, T_{i+1} - T_i), i \in [1; n-1]\}$ where $\{T_i\}_n$ is the set of local maxima times of the trace	$\begin{aligned} & \exists(t_1, t_2) (t_2 - t_1 = d) \\ & \wedge \mathbf{F}((\frac{dA}{dt} \geq 0) \wedge \mathbf{X}((\frac{dA}{dt} < 0) \wedge (\text{Time} = t_1) \\ & \wedge ((\frac{dA}{dt} < 0) \mathbf{U}((\frac{dA}{dt} \geq 0) \wedge \\ & ((\frac{dA}{dt} \geq 0) \mathbf{U}((\frac{dA}{dt} < 0) \wedge (\text{Time} = t_2)))))))) \end{aligned}$
DistanceSuccPeaks(A,B,d): There should be two successive peaks of A and B distant by d $D = \{T_{B_j} - T_{A_i} i \in [1; n], j \in [1; m], j = \min_{T_{B_k} > T_{A_i}} \{k\}\}$ where $\{T_{A_i}\}_n$ is the set of local maxima times for the trace of A and $\{T_{B_i}\}_m$ is the set of local maxima times for the trace of B	$\begin{aligned} & \exists(t_1, t_2) (t_2 - t_1 = d) \\ & \wedge \mathbf{F}((\frac{dA}{dt} \geq 0) \wedge \mathbf{X}((\frac{dA}{dt} < 0) \wedge (\text{Time} = t_1) \\ & \wedge ((\frac{dB}{dt} < 0) \mathbf{U}((\frac{dB}{dt} \geq 0) \wedge \\ & ((\frac{dB}{dt} \geq 0) \mathbf{U}((\frac{dB}{dt} < 0) \wedge (\text{Time} = t_2)))))))) \end{aligned}$
DetailedSuccPeaks(A,t1,t2,m1,m2,dp,da1,da2): There are two successive peaks at time t_1 and t_2 with levels m_1 and m_2 , where dp is $t_2 - t_1$, da_1 is $ m_1 - m_2 $ and da_2 is $\min(m_1, m_2) - \min_{[t_1, t_2]} M$	$\begin{aligned} & \exists([\min, tmin, minpeak], \\ & (m_2 - m_1 = da_1 m_1 - m_2 = da_1) \wedge da_1 > 0 \\ & \wedge dp = p_2 - p_1 \wedge minpeak \leq m_1 \wedge minpeak \leq m_2 \\ & \wedge da_2 = minpeak - min \\ & \wedge (da_2 = m_1 - min da_2 = m_2 - min) \\ & \wedge \mathbf{F}((\frac{dA}{dt} \geq 0) \wedge \mathbf{X}(\frac{dA}{dt} < 0 \wedge A = m_1 \wedge \text{Time} = p_1 \\ & \wedge (\frac{dA}{dt} \leq 0) \mathbf{U}(\frac{dA}{dt} > 0 \wedge A = min \wedge \text{Time} = tmin \\ & \wedge (\frac{dA}{dt} > 0) \mathbf{U}(\frac{dA}{dt} \leq 0 \wedge A = m_2 \wedge \text{Time} = p_2)))) \end{aligned}$
peakAmplitude(A,ampl): There should be a peak of amplitude $ampl$ $D = \{a_i\}$: the set of amplitudes of the local maxima for the trace of A, defined as the difference between the local maximum value and the preceding local minimum value	$\begin{aligned} & \exists(m_1) \mathbf{F}(\frac{dA}{dt} < 0 \wedge \mathbf{X}(\frac{dA}{dt} \geq 0 \wedge A = m_1 \\ & \wedge (\frac{dA}{dt} \geq 0) \mathbf{U}(\frac{dA}{dt} < 0 \wedge A = m_1 + ampl))) \end{aligned}$

 Table 4.3: Specific relations defined with dedicated solvers, and equivalent FO-LTL(\mathbb{R}_{lin}) formulae

Macro and behaviour	Equivalent FO-LTL(\mathbb{R}_{lin}) formula
<p>MaxDiffDistancePeaks(A,d): The variation of successive peak-to-peak distances for A should be smaller than d $D = \lceil \max_{(i) \in \llbracket 1, n \rrbracket} T_{i+2} - 2 * T_{i+1} + T_i , +\infty \lceil$ where $\{T_i\}$ is the set of local maxima times of the trace</p>	$\begin{aligned} & \exists (t_1, t_2, d_1, d_2) (t_2 > t_1 \wedge (d_2 - d_1 = \text{maxdiffdist} d_1 - \\ & d_2 = \text{maxdiffdist}) \wedge \text{maxdiffdist} \geq 0 \\ & \wedge \mathbf{F}(\frac{dX}{dt} \geq 0 \wedge \mathbf{X}(\frac{dX}{dt} < 0 \wedge \text{Time} = t_1 \\ & \wedge (\frac{dX}{dt} < 0) \mathbf{U}(\frac{dX}{dt} \geq 0 \\ & \wedge (\frac{dX}{dt} \geq 0) \mathbf{U}(\frac{dX}{dt} < 0 \wedge \text{Time} = t_1 + d_1)))) \\ & \wedge \mathbf{F}(\frac{dX}{dt} \geq 0 \wedge \mathbf{X}(\frac{dX}{dt} < 0 \wedge \text{Time} = t_2 \\ & \wedge (\frac{dX}{dt} < 0) \mathbf{U}(\frac{dX}{dt} \geq 0 \\ & \wedge (\frac{dX}{dt} \geq 0) \mathbf{U}(\frac{dX}{dt} < 0 \wedge \text{Time} = t_2 + d_2)))) \\ & \wedge !(\exists (t_1, t_2, d_1, d_2, d_3) (d_3 > \text{maxdiffdist} \\ & \wedge t_2 > t_1 \wedge (d_2 - d_1 = d_3 d_1 - d_2 = d_3) \wedge d_3 \geq 0 \\ & \wedge \mathbf{F}(\frac{dX}{dt} \geq 0 \wedge \mathbf{X}(\frac{dX}{dt} < 0 \wedge \text{Time} = t_1 \\ & \wedge (\frac{dX}{dt} < 0) \mathbf{U}(\frac{dX}{dt} \geq 0 \\ & \wedge (\frac{dX}{dt} \geq 0) \mathbf{U}(\frac{dX}{dt} < 0 \wedge \text{Time} = t_1 + d_1)))) \\ & \wedge \mathbf{F}(\frac{dX}{dt} \geq 0 \wedge \mathbf{X}(\frac{dX}{dt} < 0 \wedge \text{Time} = t_2 \\ & \wedge (\frac{dX}{dt} < 0) \mathbf{U}(\frac{dX}{dt} \geq 0 \\ & \wedge (\frac{dX}{dt} \geq 0) \mathbf{U}(\frac{dX}{dt} < 0 \wedge \text{Time} = t_2 + d_2)))))) \end{aligned}$
<p>maxDiffAmplPeaks(A,d): The variation of peaks amplitudes for A should be smaller than d $D = \lceil \max_{(i,j) \in \llbracket 1, n \rrbracket^2} a_i - a_j , +\infty \lceil$ where $\{a_i\}$ is the set of amplitudes of the local maxima for the trace of A, defined as the difference between the local maximum value and the preceding local minimum value</p>	$\begin{aligned} & \exists (t, m_1, m_2, a_1, a_2) \\ & ((a_2 - a_1 = \text{maxdiffampl} \vee a_1 - a_2 = \\ & \text{maxdiffampl}) \wedge \text{maxdiffampl} > 0 \\ & \wedge \mathbf{F}((\frac{dX}{dt} < 0) \wedge \mathbf{X}(\frac{dX}{dt} \geq 0 \wedge X = m_1 \wedge \text{Time} = t \\ & \wedge (\frac{dX}{dt} \geq 0) \mathbf{U}(\frac{dX}{dt} < 0 \wedge X = m_1 + a_1 \\ & \wedge \mathbf{F}((\frac{dX}{dt} \geq 0) \wedge \mathbf{X}(\frac{dX}{dt} \geq 0 \wedge X = m_2 \wedge \text{Time} > t \\ & \wedge (\frac{dX}{dt} \geq 0) \mathbf{U}(\frac{dX}{dt} < 0 \wedge X = m_2 + a_2)))))) \\ & \wedge !(\exists (t, m_1, m_2, a_1, a_2, \text{diff}2) \\ & (\text{diff}2 > \text{maxdiffampl} \\ & \wedge (a_2 - a_1 = \text{diff}2 a_1 - a_2 = \text{diff}2) \wedge \text{diff}2 > 0 \\ & \wedge \mathbf{F}((\frac{dX}{dt} < 0) \wedge \mathbf{X}(\frac{dX}{dt} \geq 0 \wedge X = m_1 \wedge \text{Time} = t \\ & \wedge (\frac{dX}{dt} \geq 0) \mathbf{U}(\frac{dX}{dt} < 0 \wedge X = m_1 + a_1 \\ & \wedge \mathbf{F}((\frac{dX}{dt} < 0) \wedge \mathbf{X}(\frac{dX}{dt} \geq 0 \wedge X = m_2 \wedge \text{Time} > t \\ & \wedge (\frac{dX}{dt} \geq 0) \mathbf{U}(\frac{dX}{dt} < 0 \wedge X = m_2 + a_2)))))) \end{aligned}$
<p>maxDiffAmplSuccPeaks(A,d): The variation of peaks amplitudes for A should be smaller than d $D = \lceil \max_{(i) \in \llbracket 1, n \rrbracket} a_i - a_{i+1} , +\infty \lceil$ where $\{a_i\}$ is the set of amplitudes of the local maxima for the trace of A, defined as the difference between the local maximum value and the preceding local minimum value</p>	$\begin{aligned} & \exists ([\text{max}1, \text{min}1, \text{max}2, \text{min}2, \text{ampl}1, \text{ampl}2], \\ & \text{max}1 - \text{min}1 = \text{ampl}1 \wedge \text{max}2 - \text{min}2 = \text{ampl}2 \wedge \\ & (\text{ampl}2 - \text{ampl}1 = d \text{ampl}1 - \text{ampl}2 = d) \wedge \text{diff} > 0 \\ & \wedge \mathbf{F}((\frac{dA}{dt} < 0) \wedge \mathbf{X}(\frac{dA}{dt} \geq 0 \wedge A = \text{min}1 \\ & \wedge (\frac{dA}{dt} \geq 0) \mathbf{U}(\frac{dA}{dt} < 0 \wedge A = \text{max}1 \wedge \text{Time} > 10 \\ & \wedge (\frac{dA}{dt} < 0) \mathbf{U}(\frac{dA}{dt} \geq 0 \wedge A = \text{min}2 \\ & \wedge (\frac{dA}{dt} \geq 0) \mathbf{U}(\frac{dA}{dt} < 0 \wedge A = \text{max}2)))))) \\ & \wedge \neg \exists ([d2], d2 > d \wedge \\ & \exists ([\text{max}1, \text{min}1, \text{max}2, \text{min}2, \text{ampl}1, \text{ampl}2], \\ & \text{max}1 - \text{min}1 = \text{ampl}1 \wedge \text{max}2 - \text{min}2 = \text{ampl}2 \wedge \\ & (\text{ampl}2 - \text{ampl}1 = d2 \text{ampl}1 - \text{ampl}2 = d2) \wedge d2 > 0 \\ & \wedge \mathbf{F}((\frac{dA}{dt} < 0) \wedge \mathbf{X}(\frac{dA}{dt} \geq 0 \wedge A = \text{min}1 \\ & \wedge (\frac{dA}{dt} \geq 0) \mathbf{U}(\frac{dA}{dt} < 0 \wedge A = \text{max}1 \wedge \text{Time} > 10 \\ & \wedge (\frac{dA}{dt} < 0) \mathbf{U}(\frac{dA}{dt} \geq 0 \wedge A = \text{min}2 \\ & \wedge (\frac{dA}{dt} \geq 0) \mathbf{U}(\frac{dA}{dt} < 0 \wedge A = \text{max}2)))))) \end{aligned}$

Table 4.4: Specific relations defined with dedicated solvers, and equivalent FO-LTL(\mathbb{R}_{lin}) formulae

Macro and behaviour	Equivalent FO-LTL(\mathbb{R}_{lin}) formula
period(A,p): Mean of the last two peak-to-peak intervals	$\begin{aligned} & \exists(p1, p2, t1, t2, tmax) (\\ & p1 = t2 - t1 \wedge p2 = tmax - t2 \wedge 2 * period = p1 + p2 \\ & \wedge \mathbf{F}((\frac{dA}{dt} \geq 0) \wedge \mathbf{X}(\frac{dA}{dt} < 0 \wedge Time = t1 \\ & \wedge (\frac{dA}{dt} < 0) \mathbf{U}(\frac{dA}{dt} \geq 0 \\ & \wedge (\frac{dA}{dt} \geq 0) \mathbf{U}(\frac{dA}{dt} < 0 \wedge Time = t2 \\ & \wedge (\frac{dA}{dt} < 0) \mathbf{U}(\frac{dA}{dt} \geq 0 \\ & \wedge (\frac{dA}{dt} \geq 0) \mathbf{U}(\frac{dA}{dt} < 0 \wedge Time = tmax)))))) \\ & \neg \exists(t) (\mathbf{F}((\frac{dA}{dt} > 0) \wedge \mathbf{X}(\frac{dA}{dt} \leq 0 \wedge Time = t)) \wedge t > tmax) \end{aligned}$
period(A,p,d1,d2): Mean of the last two peak-to-peak intervals, maximal difference between two successive peak-to-peak intervals, maximal difference between two successive amplitudes	$\begin{aligned} & period(A,p) \\ & \wedge \maxDiffDistancePeaks(A,d1) \\ & \wedge \maxDiffAmplPeaks(A,d1) \end{aligned}$
phase(A,B,p): Mean of the last two peak-to-peak intervals between A and B	$\begin{aligned} & \exists(p1, p2, t1, t2, tmax1, tmax2) \\ & (p1 = t2 - t1 \wedge p2 = tmax2 - tmax1 \wedge 2 * phase = p1 + p2 \\ & \wedge \mathbf{F}((\frac{dA}{dt} > 0) \wedge \mathbf{X}(\frac{dB}{dt} \leq 0 \wedge Time = t1 \\ & \wedge (\frac{dB}{dt} \leq 0) \mathbf{U}(\frac{dB}{dt} > 0 \\ & \wedge (\frac{dB}{dt} > 0) \mathbf{U}(\frac{dB}{dt} \leq 0 \wedge Time = t2 \\ & \wedge (\frac{dA}{dt} \leq 0) \mathbf{U}(\frac{dA}{dt} > 0 \\ & \wedge (\frac{dA}{dt} > 0) \mathbf{U}(\frac{dA}{dt} \leq 0 \wedge Time = tmax1 \\ & \wedge (\frac{dB}{dt} \leq 0) \mathbf{U}(\frac{dB}{dt} > 0 \\ & \wedge (\frac{dB}{dt} > 0) \mathbf{U}(\frac{dB}{dt} \leq 0 \wedge Time = tmax2)))))) \\ & \wedge \forall(t) (\mathbf{F}((\frac{dB}{dt} > 0) \wedge \mathbf{X}(\frac{dB}{dt} \leq 0 \wedge Time = t)) \wedge t \leq tmax2) \\ & \wedge \exists(t) (\mathbf{F}((\frac{dB}{dt} > 0) \wedge \mathbf{X}(\frac{dB}{dt} \leq 0 \wedge Time = t)) \wedge t = tmax2)) \end{aligned}$

 Table 4.5: Specific relations defined with dedicated solvers, and equivalent FO-LTL(\mathbb{R}_{lin}) formulae

4.3 Trace simplification preserving temporal specifications

As explained in the introduction of this chapter, the usual computation of the validity domains involves computing domains for each subformula on each point of the trace s_i . When dealing with temporal data coming from numerical integration, especially of stiff systems, n can be very high, which induces a high computational cost, $\mathcal{O}(n^k)$, where k is the number of variables.

An efficient solution to this issue involves simplifying the numerical trace without changing the generic domain solving algorithm. In particular for analyzing oscillating traces, one can take advantage of the fact that most oscillations properties such as pseudo-period and pseudo-phase rely only on local extrema values in the trace, while all other points can be neglected.

In this section we provide a series of trace simplifications which are correct to perform for some common temporal logic formulae. We give general soundness theorems that indicate some general conditions on the syntax of the formulae under which it is correct to keep only the time points corresponding to the local extrema of the molecules in the trace, or the crossing points between molecular concentrations, and apply this approach to period and phase constraints on the toy oscillator model introduced in Chapter 3.

Speed-ups by several orders of magnitude are obtained by trace simplification, even when produced by smart numerical integration methods (e.g. Rosenbrock's implicit method), making trace simplification comparable with specific solvers. Trace simplification can also be used in combination with specific solvers, resulting in even more efficient model checking.

Trace Simplifications

Let us first define more precisely the formal framework for defining trace simplifications.

Definition 4.4 (Trace simplification) Let T be a finite trace (s_0, \dots, s_n) and ϕ an FO-LTL(\mathbb{R}_{lin}) formula with constraints over the states of T .

T' is a simplification of T for ϕ at i , written $T' \preceq_\phi^i T$ if:

- $T' = (s_{j_0}, \dots, s_{j_k})$ for $J = \{j_0, \dots, j_k\}$ a subset of the indices $\{0, \dots, n\}$ such that $j_0 < \dots < j_k$, i.e., T' is a subtrace of T ;
- $\mathcal{D}_{s_i, \phi}^T = \mathcal{D}_{s_{j_i}, \phi}^{T'}$ where j_i is the smallest index in J such that $j_i \geq i$, i.e. the validity domains on T at i and T' at j_i are equal.

T' is a simplification of T for ϕ , written $T' \preceq_\phi T$ when it is a simplification of T at s_0 , i.e., $\mathcal{D}_{T, \phi} = \mathcal{D}_{T', \phi}$.

T' is a strict simplification of T for ϕ , written $T' \prec T$ if $J \subsetneq \{0, \dots, n\}$.

T' is an optimal simplification of T for ϕ if its cardinal is minimal in the set of the simplifications of T for ϕ .

Property-driven reduction of the system under analysis is a technique that has been addressed many times in the history of computer science. In the framework of abstract interpretation [Cousot and Cousot, 1977], not only the states but also the transitions can be abstracted in a new system for simplifying the analysis of some given properties. The definition above can be seen as a particular instance of this framework where a subset of states on the trace is preserved without abstraction, and the transitions are abstracted accordingly to this subset. This abstraction reflects our motivation of computing exact validity domains for formula variables (no state domain abstraction) more efficiently (transition abstraction).

Examples

Most of the equations for $\mathcal{D}_{s_i, \phi}^T$ in Definition 3.1 are local, in the sense that they only need information about the state at s_i . One obvious case of simplification is when the unions or intersections involved in the domains for **F**, **G** and **U** can be computed on a strict subset of the points, sometimes even a singleton.

Since it will come up often in the following examples, let us define a simple subtrace containing all the local extrema and the initial point of the trace.

Definition 4.5 (Extrema Subtrace) Let $T = (s_0, \dots, s_n)$ be a trace, T_x^e is the subtrace of T defined as follows:

$$T_x^e = \{s_i \in T \mid (dx/dt)_{i-1} > 0 \wedge (dx/dt)_i \leq 0\} \\ \cup \{s_i \in T \mid (dx/dt)_{i-1} < 0 \wedge (dx/dt)_i \geq 0\} \cup \{s_0\}$$

We shall write $T^e = \bigcup_x T_x^e$

T^e is shown in Figure 4.2 for two molecules.

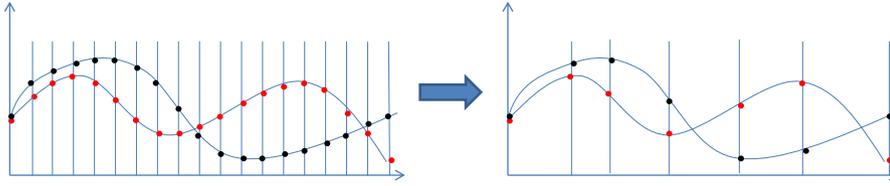


Figure 4.2: Trace simplification by discarding the temporal points that are not local extrema for at least one molecule

In the following examples, we present some of the FO-LTL(\mathbb{R}_{lin}) formulae given in Chapter 3. For each of these, we compute the corresponding validity domain and examine possible trace simplifications.

Minimal Amplitude

Formula: $\phi = \exists v \mid \mathbf{F}(A < v) \wedge \mathbf{F}(A > v + a)$

Validity Domain Let $s_{\min A}$ and $s_{\max A}$ be the points of the trace where A is respectively minimum and maximum.

$$\begin{aligned} \mathcal{D}_{T, \phi} &= \Pi_a(\mathcal{D}_{s_0, \mathbf{F}(A < v)}^T \cap \mathcal{D}_{s_0, \mathbf{F}(A > v+a)}^T) \\ &= \Pi_a\left(\left(\bigcup_{i=0}^n \mathcal{D}_{s_i, A < v}^T\right) \cap \left(\bigcup_{i=0}^n \mathcal{D}_{s_i, A > v+a}^T\right)\right) \quad (*) \\ &= \Pi_a(\mathcal{D}_{s_{\min A}, A < v}^T \cap \mathcal{D}_{s_{\max A}, A > v+a}^T) \quad (*) \end{aligned}$$

Trace Simplification From the computation of the domain (equations marked with a (*)), one can see that both unions are actually equal to a single domain, only dependent on the maximal or minimal value reached by A , and not on the other points of T . Therefore knowing the values of $s_{\min A}, s_{\max A}$ leads to an optimal trace simplification T_J where $J = \{s_{\min A}, s_{\max A}\}$.

Note that because of the semantic link between A and $\frac{dA}{dt}$, T_A^e contains $s_{\min A}$ and $s_{\max A}$ and therefore will result in the same unions in the computation of the domain, hence T_A^e is a simplification of T for ϕ .

Threshold

Formula: $\phi = \mathbf{F}(Time > 20 \wedge A < v)$

Validity Domain Let T be a trace (s_0, \dots, s_n) and $T_{>20}$ its subtrace on the points $J = \{0 \leq i \leq n \mid Time_{s_i} > 20\}$. As before, we chose some $s_{minA>20}$, a point where A is minimum on $T_{>20}$.

$$\begin{aligned}
 \mathcal{D}_{T,\phi} &= \mathcal{D}_{s_0, \mathbf{F}(Time > 20 \wedge A < v)}^T \\
 &= \bigcup_{i=0}^n \mathcal{D}_{s_i, Time > 20 \wedge A < v}^T \\
 &= \bigcup_{i=0}^n (\mathcal{D}_{s_i, Time > 20}^T \cap \mathcal{D}_{s_i, A < v}^T) \\
 &= \bigcup_{i \in J} \mathcal{D}_{s_i, A < v}^T \quad (*) \\
 &= \mathcal{D}_{s_{minA>20}, A < v}^T \quad (*)
 \end{aligned}$$

Trace Simplification As shown by the marked equations, the single point $\{s_{minA>20}\}$ is enough to compute the big union of the domain, it defines an optimal trace simplification of T for ϕ .

Notice that T_A^e is not a simplification unless it does contain a local minimum such that $Time > 20$: if that is not the case, e.g. always increasing trace, $s_{minA>20}$ will be the first state after $Time = 20$, which is not a local extremum.

Crossing

Formula: $\phi = \mathbf{F}(A > B \wedge \mathbf{X}(A \leq B \wedge Time = t))$

$$\begin{aligned}
 \text{Validity Domain} \quad \mathcal{D}_{T,\phi} &= \bigcup_{i=0}^n (\mathcal{D}_{s_i, A_{s_i} > B_{s_i}}^T \cap (\mathcal{D}_{s_{i+1}, A_{s_{i+1}} \leq B_{s_{i+1}}}^T \cap \mathcal{D}_{s_{i+1}, Time=t}^T)) \\
 &= \bigcup_{i \in \{0, \dots, n\} \mid A_{s_i} > B_{s_i} \wedge A_{s_{i+1}} \leq B_{s_{i+1}}} \{Time_{s_{i+1}}\}
 \end{aligned}$$

The computation above simply discards from the union the trace points where the intersection is empty because one of the two first members is empty.

Trace Simplification Once again, for any trace $T = (s_0, \dots, s_n)$, the validity domain is a big union that can be restricted to the points of $J = \{i, i+1 \in \{0, \dots, n\} \mid A_{s_i} > B_{s_i} \wedge A_{s_{i+1}} \leq B_{s_{i+1}}\}$, which defines a simplification T_J of T for ϕ . As in Example 4.3, T_A^e is not a simplification of T for ϕ since it obviously misses the points at which $Time$ has to be computed.

Peak

Formula: $\phi = \mathbf{F}(\frac{dA}{dt} > 0 \wedge \mathbf{X}(\frac{dA}{dt} \leq 0 \wedge Time = t))$

Validity Domain The reasoning is the same as for Example 4.3.

$$\begin{aligned}
 \mathcal{D}_{T,\phi} &= \mathcal{D}_{s_0,\phi}^T = \bigcup_{i=0}^n (\mathcal{D}_{s_i, \frac{dA}{dt} > 0}^T \cap (\mathcal{D}_{s_{i+1}, \frac{dA}{dt} \leq 0}^T \cap \mathcal{D}_{s_{i+1}, \text{Time}=t}^T)) \\
 &= \bigcup_{i \in \{0, \dots, n\} \mid (\frac{dA}{dt})_{s_i} > 0 \wedge (\frac{dA}{dt})_{s_{i+1}} \leq 0} \mathcal{D}_{s_{i+1}, \text{Time}=t}^T \\
 &= \bigcup_{i \in \{0, \dots, n\} \mid (\frac{dA}{dt})_{s_i} > 0 \wedge (\frac{dA}{dt})_{s_{i+1}} \leq 0} \{ \text{Time}_{s_{i+1}} \}
 \end{aligned}$$

Trace Simplification As above, for any trace $T = (s_0, \dots, s_n)$, $J = \{i, i+1 \in \{0, \dots, n\} \mid \frac{dA}{dt}_{s_i} > 0 \wedge \frac{dA}{dt}_{s_{i+1}} \leq 0\}$ defines a simplification T_J of T for ϕ .

Note that T_A^e is also a simplification of T for ϕ since it contains all $i+1$ at which A_{s_i} is used and a predecessor with the right sign of the derivative, either s_0 or a local minimum preceding the peak. Note also that $|T_A^e| \leq |T_J| + 2$ since there can be one local minimum more than there are peaks, plus the origin s_0 . T_J and T_A^e are shown in Figure 4.3.

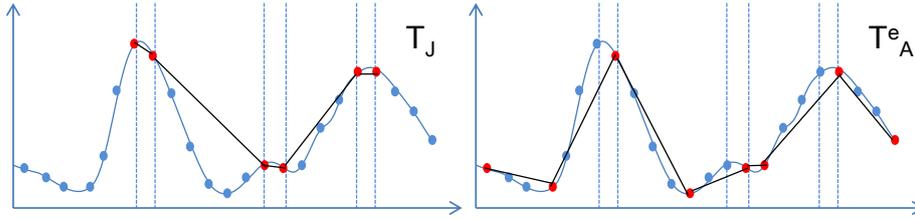


Figure 4.3: Trace simplifications T_J (left) and T_A^e (right).

Period

Formula:

$$\begin{aligned}
 \phi &= \exists(t_1, t_2) \mid p = t_2 - t_1 \wedge t_1 < t_2 \\
 &\wedge \mathbf{F}(\frac{dA}{dt} > 0 \wedge \mathbf{X}(\frac{dA}{dt} \leq 0 \wedge \text{Time} = t_1)) \\
 &\wedge \mathbf{F}(\frac{dA}{dt} > 0 \wedge \mathbf{X}(\frac{dA}{dt} \leq 0 \wedge \text{Time} = t_2)) \\
 &\wedge \neg \exists t_3 \mid t_1 < t_3 < t_2 \wedge \mathbf{F}(\frac{dA}{dt} > 0 \wedge \mathbf{X}(\frac{dA}{dt} \leq 0 \wedge \text{Time} = t_3))
 \end{aligned}$$

ϕ encodes the fact that t_1 and t_2 are peaks, with no peak in between.

Trace Simplification One can notice that the domain is formed by the same kind of union as in Example 4.3, repeated three times, and under top-level projections/intersections/complementations. Now, remark that a simplification for the formula of Example 4.3 will, by definition, allow to compute correctly the domains for all three Fformulae, and therefore is a simplification for the compound ϕ . This is a special case of Theorem 4.6 detailed in the next section.

It follows that T_J of Example 4.3 and T_A^e are simplifications of T for ϕ .

Equivalent Formula:

$$\begin{aligned} \phi = \exists(t_1, t_2) \mid & p = t_2 - t_1 \wedge \mathbf{F}\left(\frac{dA}{dt} > 0 \wedge \mathbf{X}\left(\frac{dA}{dt} \leq 0 \wedge \text{Time} = t_1\right.\right. \\ & \left.\left. \wedge \left(\frac{dA}{dt} \leq 0\right) \mathbf{U}\left(\frac{dA}{dt} > 0\right)\right.\right. \\ & \left.\left. \wedge \left(\left(\frac{dA}{dt} > 0\right) \mathbf{U}\left(\frac{dA}{dt} \leq 0 \wedge \text{Time} = t_2\right)\right)\right)\right) \end{aligned}$$

Validity Domain Note first that the validity domain of the subformula $\psi = \frac{dA}{dt} > 0 \wedge \left(\left(\frac{dA}{dt} > 0\right) \mathbf{U}\left(\frac{dA}{dt} \leq 0 \wedge \text{Time} = t_2\right)\right)$ is computed at each time point s_i like this:

$$\mathcal{D}_{s_i, \psi}^T = \mathcal{D}_{s_i, \frac{dA}{dt} > 0}^T \cap \bigcup_{j=i}^n \left(\mathcal{D}_{s_j, \frac{dA}{dt} \leq 0 \wedge \text{Time} = t_2}^T \cap \left(\bigcap_{k=i}^{j-1} \left(\mathcal{D}_{s_k, \frac{dA}{dt} > 0}^T \right) \right) \right)$$

Since $\mathcal{D}_{s_i, \frac{dA}{dt} > 0}^T$ is either empty or equal to the whole space when $\frac{dA}{dt}_{s_i}$ is respectively negative or strictly positive, it holds that $\mathcal{D}_{s_i, \psi}^T$ is empty if $\frac{dA}{dt}_{s_i} \leq 0$, otherwise:

$$\begin{aligned} \mathcal{D}_{s_i, \psi}^T &= \bigcup_{j=i}^n \left(\mathcal{D}_{s_j, \frac{dA}{dt} \leq 0 \wedge \text{Time} = t_2}^T \cap \left(\bigcap_{k=i}^{j-1} \left(\mathcal{D}_{s_k, \frac{dA}{dt} > 0}^T \right) \right) \right) \\ &= \bigcup_{j=i}^n \left(\mathcal{D}_{s_j, \frac{dA}{dt} \leq 0}^T \cap \mathcal{D}_{s_j, \text{Time} = t_2}^T \cap \left(\bigcap_{k=i}^{j-1} \left(\mathcal{D}_{s_k, \frac{dA}{dt} > 0}^T \right) \right) \right) \\ &= \bigcup_{j \in \{i, \dots, n\} \mid \left(\frac{dA}{dt}\right)_{s_j} \leq 0 \wedge \forall k \in \{i, \dots, j-1\}, \left(\frac{dA}{dt}\right)_{s_k} > 0} \mathcal{D}_{s_j, \text{Time} = t_2}^T \\ &= \bigcup_{j \in \{i, \dots, n\} \mid \left(\frac{dA}{dt}\right)_{s_j} \leq 0 \wedge \forall k \in \{i, \dots, j-1\}, \left(\frac{dA}{dt}\right)_{s_k} > 0} \{ \text{Time}_{s_j} \} \end{aligned}$$

This union is in fact restricted to the first point s_j after s_i where $\frac{dA}{dt}$ is no longer strictly positive.

With the same reasoning, the validity domain for the whole formula becomes:

$$\mathcal{D}_{T, \phi} = \bigcup_{(i, j) \in P} \{ \text{Time}_{s_{j+1}} - \text{Time}_{s_{i+1}} \}$$

where P is the set of pairs of successive peaks:

$$\begin{aligned} P = \{ (i, j) \mid & \left(\frac{dA}{dt}\right)_{s_i} > 0 \wedge \left(\frac{dA}{dt}\right)_{s_{i+1}} \leq 0 \wedge \left(\frac{dA}{dt}\right)_{s_j} > 0 \wedge \left(\frac{dA}{dt}\right)_{s_{j+1}} \leq 0 \\ & \wedge \neg \exists i < k < j \mid \left(\frac{dA}{dt}\right)_{s_k} > 0 \wedge \left(\frac{dA}{dt}\right)_{s_{k+1}} \leq 0 \} \end{aligned}$$

T_A^e is a simplification of T for ϕ since it contains all the peaks of the trace.

General Simplification Results

Example 4.3 shows that if one can simplify subformulae, one might obtain a simplification for the whole formula. Indeed, with some hypotheses, the patterns described in the previous section can actually be composed.

The first theorem simply notices that if the highest-level temporal subformulae have a simplification, it also holds for the compound formula.

Theorem 4.6 Let T be a trace containing a state s_i , ϕ and ψ two formulae and T' such that $T' \preceq_{\phi}^i T$ and $T' \preceq_{\psi}^i T$. Then $T' \preceq_{\mu}^i T$ for μ equal to

$$\phi \wedge \psi \text{ or } \phi \vee \psi \text{ or } \neg\phi \text{ or } \exists x\phi \text{ or } \forall x\phi$$

Proof 4.7 We have $\mathcal{D}_{s_i, \phi}^T = \mathcal{D}_{s_{j_i}, \phi}^{T'}$ and the same for ψ , therefore $\mathcal{D}_{s_i, \phi \wedge \psi}^T = \mathcal{D}_{s_i, \phi}^T \cap \mathcal{D}_{s_i, \psi}^T = \mathcal{D}_{s_{j_i}, \phi}^{T'} \cap \mathcal{D}_{s_{j_i}, \psi}^{T'} = \mathcal{D}_{s_{j_i}, \phi \wedge \psi}^{T'}$ and the same for the other operators.

Note that it is not true that if T' is a simplification for ϕ and T'' a simplification for ψ , then the union of the points in T' and T'' defines a simplification for $\phi \vee \psi$: indeed, adding points to a simplification can invalidate it, for instance if the formula contains \mathbf{X} . Now, remark that if a subtrace contains extreme domains, it is a simplification for **Fand G**:

Theorem 4.8 Let $T = (s_0, \dots, s_n)$ be a trace, ϕ a formula and $T' = T_J$ a subtrace of T such that: $\forall j \in J, T' \preceq_{\phi}^j T$ and $\forall 0 \leq i \leq n, \exists j \in J, \mathcal{D}_{s_i, \phi}^T \subset \mathcal{D}_{s_j, \phi}^{T'}$ (resp. $\mathcal{D}_{s_i, \phi}^T \supset \mathcal{D}_{s_j, \phi}^{T'}$) then: $T' \preceq_{F\phi} T$ (resp. $T' \preceq_{G\phi} T$)

Proof 4.9 We have, $\forall 0 \leq i \leq n, \mathcal{D}_{s_i, \phi}^T \subset \mathcal{D}_{s_j, \phi}^{T'}$ it follows that $\bigcup_{i=0}^n \mathcal{D}_{s_i, \phi}^T \subset \bigcup_{j \in J} \mathcal{D}_{s_j, \phi}^{T'}$. The other inclusion is immediate since J is a subset of the indices $\{0, \dots, n\}$ and we have simplification for ϕ at those indices. The result for **G** is obtained similarly.

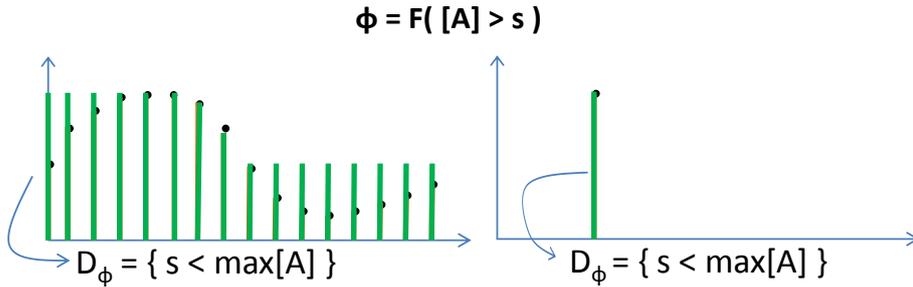


Figure 4.4: Illustration of Theorem 4.8 for the FO-LTL formula $\phi = \mathbf{F}([A] > s)$.

Consider now the case of formulae without free variables, their domain is either empty or full, which can be taken advantage of:

Corollary 4.10 Let $T = (s_0, \dots, s_n)$ be a trace, ϕ a formula, c a constraint without free variables and J_c be the subset of indices defined by $J_c = \{0 \leq i \leq n \mid s_i \models c\}$. If $\forall i \in J_c, T_{J_c} \preceq_{\phi}^i T$ then $T_{J_c} \preceq_{F(c \wedge \phi)} T$ and $T_{J_c} \preceq_{G(\neg c \vee \phi)} T$

Proof 4.11 Let us prove the result for **F**, then Thm. 4.6 can give it for **G**. We will simply apply the above theorem to $c \wedge \phi$. The first hypothesis of Thm. 4.8 is satisfied by T_{J_c} since $T_{J_c} \preceq_{\phi}^i T \Rightarrow T_{J_c} \preceq_{c \wedge \phi}^i T$. For the second hypothesis, it is enough to notice that if $i \notin J_c$ then $\mathcal{D}_{s_i, c \wedge \phi}^T = \mathcal{D}_{s_i, c}^T \cap \mathcal{D}_{s_i, \phi}^T = \emptyset$.

Note that in general it is not easy to simplify the validity domain computing procedure for a pattern $\mathbf{F}(\phi \wedge \psi)$. On the contrary $\mathcal{D}_{T, \mathbf{F}(\phi \vee \psi)} = \mathcal{D}_{T, \mathbf{F}(\phi) \vee \mathbf{F}(\psi)}$, which can benefit from Theorem 4.6.

In many cases it is worth noticing that T_A^e satisfies the hypothesis of Thm. 4.8 for any formula $\mathbf{F}(\frac{dA}{dt} > 0 \wedge \mathbf{X}(\frac{dA}{dt} \leq 0 \wedge c))$.

Proposition 4.12 Let $\phi = \mathbf{F}(\frac{dA}{dt} > 0 \wedge \mathbf{X}(\frac{dA}{dt} \leq 0 \wedge c))$ be a formula, $T_A^e \preceq_{\phi} T$

Proof 4.13 We will apply Thm. 4.8. First note that for any extremum j in T_A^e we have $T_A^e \stackrel{j}{\prec}_{\phi} T$. Indeed, s_0 is in T_A^e but will not be used to compute D_c , on the other hand it ensures that even the first extremum does have a predecessor of the correct sign for the derivative. Now, notice that $D_{s_i, \phi}^T$ will be empty at each point not a predecessor of a state of T_A^e . At those points the domain on T is the same as that at the preceding extremum (or s_0 for the first) on T_A^e . This enforces the inclusion needed for the second hypothesis of Thm. 4.8.

Taken together, these results prove all the simplifications of the previous examples except the second formula of Example 4.3, which is a deeply nested formula with U that relies on the semantics of the *Time* variable. In this example it is easier to prove the simplification for the first formula, which is equivalent since it returns the same validity domain.

With the same reasoning, it is possible to prove that simplifying a trace with the extrema sub-trace does not change the values of validity domains for all the temporal specifications patterns defined in Tables 4.2 to 4.5, except some that do not characterize the shape of the oscillations: *incrInterv* and *IncreasingSwitch*.

If the optional transitory time argument is used, the pattern $amplitude([A],[a],T)$ additionally fails to support this trace simplification, since the first point considered in the trace, ie. the first point after time T , can also appear in the result but would not be selected in the simplified trace.

Evaluation of trace simplification

The trace simplification that keeps only local extrema has been implemented in Biocham under the command `set_simplification_trace(extrema)`. After this command has been set and at the first call of a validity domain solving on a trace, the optimal extrema trace simplification is computed: only local extrema of the molecules that appear in the temporal logic formula are kept in the simplified trace, along with the first and last points of the initial trace. If another temporal logic specification is called later on the same simulation, the trace simplification is not recomputed unnecessarily: it is updated only in the case where another molecule is involved in the specification.

Table 4.6 illustrates the time gain in solving temporal specifications with trace simplification. Ten temporal logic patterns were evaluated on the trace of the toy oscillator model introduced in Chapter 3, simulated for a 200h or 400h-time horizon. Each specification corresponds to a specific function described in Tables 4.2 to 4.5.

The trace simplification keeps only the local extrema of the molecules involved in the formula: only X for all cases except for computing $phase([X, Y_cyto])$ on the last row of the table.

18 points are selected out of 342 for a 200h time horizon when X is the only molecule involved in the specification. This simplification takes less than 1ms. For a 400h time horizon, the trace is simplified from 672 points to 35 points in about 4ms. When the two molecules X and Y_cyto are necessary to evaluate the specification (last row), 34 points out of 342 are kept for a 200h time horizon, and 68 out of 672 for a 400h time horizon. The simplification always takes less than 8ms.

It is clear from the times in the Table that both trace simplification and dedicated solvers provide a significant gain of time, sometimes with several orders of magnitude. For example, for the relation $amplitude([X],[a])$ the generic solver is particularly inefficient and both trace simplification and dedicated solvers result in a correct evaluation in a time more than 10^3 -fold shorter than the generic solver. The relation $periodErrors([X],[p,e1,e2,e3])$, which requires the evaluation of four variables, each corresponding to complex behaviors, is so long to evaluate with the generic solver that the computing time on the 400h-trace (672 points) could not be identified with Biocham before the program was automatically killed. In this case, although simplifying the trace reduces significantly the computation cost, only the dedicated solver brings a really efficient evaluation.

With this improvement, it is now possible to apply temporal logic constraints in methods that require numerous simulations and evaluations, such as parameter optimization. Continuous models can thus be calibrated to satisfy flexible constraints on dynamical behaviors.

Table 4.6: Computing time (in ms) for the validity domain of different formula patterns. Comparison between the trace simulated for 200h and 400h, before (**Bef.**) and after (**Aft.**) simplification, with generic or specific solver. The missing time for *Amplitude* is due to the fact that trace simplification is not supported by the FO-LTL(\mathbb{R}_{lin}) formula equivalent to the relation $Amplitude(Mol,a,Time)$. The missing time for *PeriodErrors* is due to a too long evaluation: the program is killed before the result can be returned.

Formula	Solver	200h		400h	
		Bef.	Aft.	Bef.	Aft.
Max([X],[m])	generic	6212	8	44207	24
	specific	4	0	8	0
Amplitude([X],[a])	generic	24029	16	176143	56
	specific	4	0	4	0
Amplitude([X],[a],30)	generic	23789	16	177127	-
	specific	4	4	4	0
DistancePeaks([X],[d])	generic	392	20	4368	228
	specific	8	4	40	28
DistanceSuccPeaks([X],[d])	generic	188	12	460	24
	specific	8	0	12	0
DistanceSuccPeaks([X],[d],50)	generic	200	8	488	28
	specific	4	0	12	0
Period([X],[p])	generic	916	48	2628	140
	specific	4	0	12	0
MaxDiffAmplPeaks([X],[d])	generic	3016	152	41994	2208
	specific	12	0	40	0
PeakAmplitude([X],[a])	generic	116	4	212	12
	specific	20	0	36	0
PeriodErrors([X],[p,e1,e2,e3])	generic	30149	1584	-	33854
	specific	12	4	40	4
Phase([X,Y_cyto],[p])	generic	1048	96	2552	248
	specific	8	0	20	0

Moreover, we have performed this comparison with trajectories simulated with Rosenbrock's implicit method for numerical integration, which results in relatively sparse simulation traces. In some cases, for example when continuous models are extended with discrete events, stiffer integration methods can be needed and produce traces with many more points. The improvement brought by trace simplification in that case is even more significant.

4.4 Trace simplification for selecting main peaks

We have seen so far that temporal logic patterns provide an elegant way to extract meaningful information from numerical traces, obtained from simulations or experiments. However, temporal specifications used to capture oscillatory characteristics such as period and phase depend heavily on local peaks and troughs. It allows the drastic trace simplification defined in Section 4.3, but on the other hand it makes them very vulnerable to irrelevant peaks that might appear in the trace and distort the result.

In particular, such sensitivity to noise in the trace is often a problem when treating experimental traces, where noise is much heavier than in simulation trajectories.

When performing experimental traces processing, an obvious solution is brought by data preprocessing to fit oscillating patterns such as sinusoid to the experimental trace. However, this preprocessing relies on assumptions on the expected shape of the processed trace.

In the case of simulation traces, the best solution is usually to merely neglect "small" peaks

which are irrelevant. We therefore define a new trace simplification T_m as a subtrace of the extrema trace T_e defined in Definition 4.5, that selects the main peaks of the trace, based on their amplitudes as described in the following selection procedure and illustrated in Figure 4.5:

- In the list of peaks in the trace, compare each pair of successive peaks.
- The "right amplitude" $AmplR$ of the first peak is computed as the difference between the value of the peak and the value of the first following point where the trace increases again.
- The "left amplitude" $AmplL$ of the second peak is also computed as the difference between the value of the peak and the value of the next point where the trace increases again.
- If one of the values is smaller than the other one multiplied by a coefficient C , the corresponding peak is discarded. Thus, if $AmplR < C * AmplL$, the first peak is discarded. If $C * AmplR > AmplL$, the second peak is discarded. The conserved trough is point where the trace reaches its minimum between the two. If none of these conditions is True, both peaks are kept.

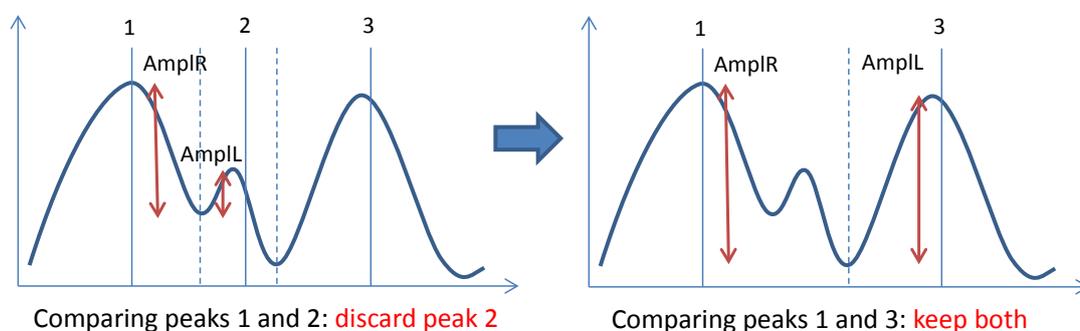


Figure 4.5: Selecting the main peaks in a trace with three peaks, with a coefficient $C = 2$: the first two peaks are first compared, and the second one is discarded. The first peak is then compared to the third and both are conserved in the simplified trace.

Thus this trace simplification keeps only the local minima and maxima corresponding to the main peaks and troughs in the trace.

This trace simplification is implemented in Biocham with the following command that allows the user to set the desired coefficient:

```
set_simplification_trace(mainpeaks, Coefficient).
```

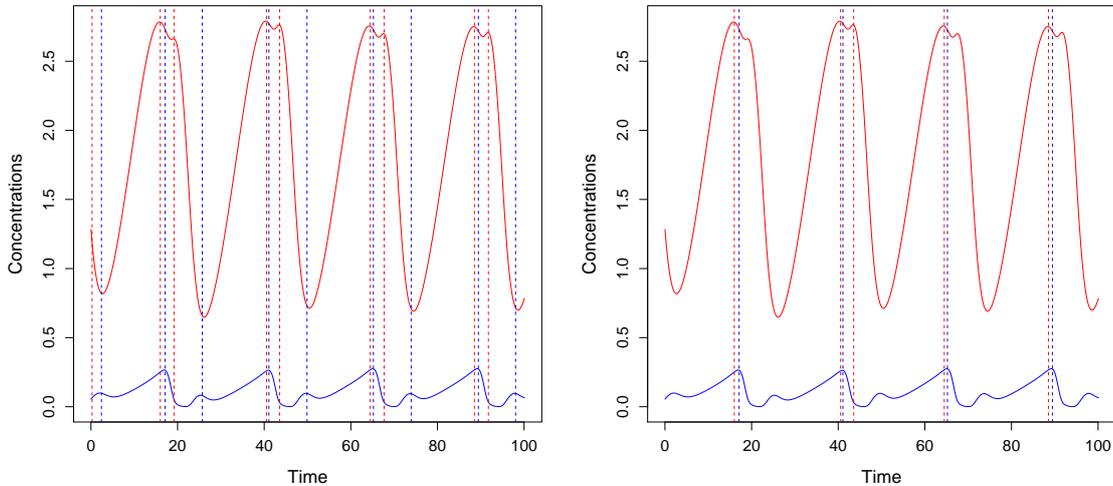


Figure 4.6: Example of a simulated trace from a model of coupling between the cell cycle and the circadian clock defined in Section 5. Peaks captured for two molecules (red and blue) with temporal logic specifications are marked with vertical dashed lines. On the left panel, no trace simplification is applied. On the right panel, the trace is simplified to select the main peaks with a coefficient $C = 2$.

Application to data analysis

We show here how the trace simplification selecting the main peaks in a trace can be useful for the analysis of experimental time series. Specifically, we aim at analyzing single-cell fluorescent data, produced by the lab of Franck Delaunay in Nice and used in [Feillet et al., 2014], that we obtained through a collaboration. The set of data is composed of 627 single-cell time series recording three fluorescent markers in NIH3T3 mouse fibroblasts during 72h. An example of time series can be seen in Figure 4.7 (dashed colored curves). The marker in red characterizes the G1 phase of the cell cycle, the one in green indicates the S/G2/M phases of the cell cycle, and the blue marker reflects the circadian clock rhythm.

The multiple-molecule trace is imported in Biocham after a basic curve smoothing (black dashed curves in Figure 4.7), and the following command is used to extract all peaks of the trace.

```
biocham: validity_domain(peak([marker1], [peak1,])).
biocham: validity_domain(peak([marker2], [peak2,])).
biocham: validity_domain(peak([marker3], [peak3,])).
```

The extracted peaks are indicated in Figure 4.7 (top panel) with dashed vertical lines. On this noisy data, spurious local maxima appear even after the curve smoothing and would result in incorrect estimations of important data characteristics such as the periods or phase relationships.

In Figure 4.8, peaks are selected with the same command preceded by the following setting: `set_simplification_trace(mainpeaks,2)`.

All spurious peaks are filtered out, and most of the important peaks are still selected.

The following command is then used to extract peak-to-peak intervals as estimations of the period of each marker, on all time series of the data set:

```
biocham: set_simplification_trace(mainpeaks,2).
biocham: validity_domain(Exists([t1,t2,amp1,amp2],
  distanceSuccPeaks([marker1], [period1,t1,t2])
  & peakAmplitude([marker1], [t1,amp1]) & peakAmplitude([marker1], [t2,amp2])
  & amp1>1 & amp2>1)).
```

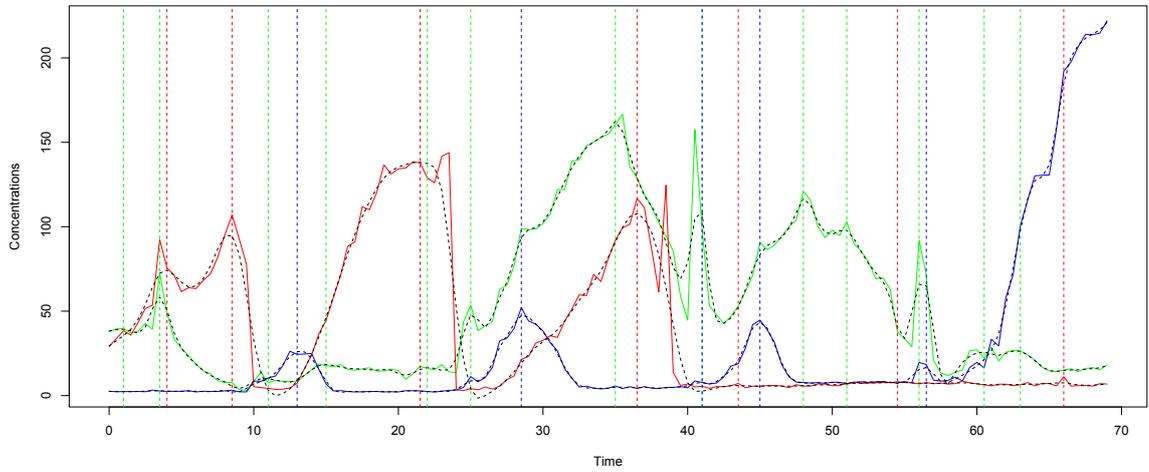


Figure 4.7: Example of an experimental trace from fluorescent cell-tracking for three markers (red, blue, green). Each trace is first preprocessed with a basic curve smoothing (black dashed lines). Extracted peak times are marked with vertical dashed lines.

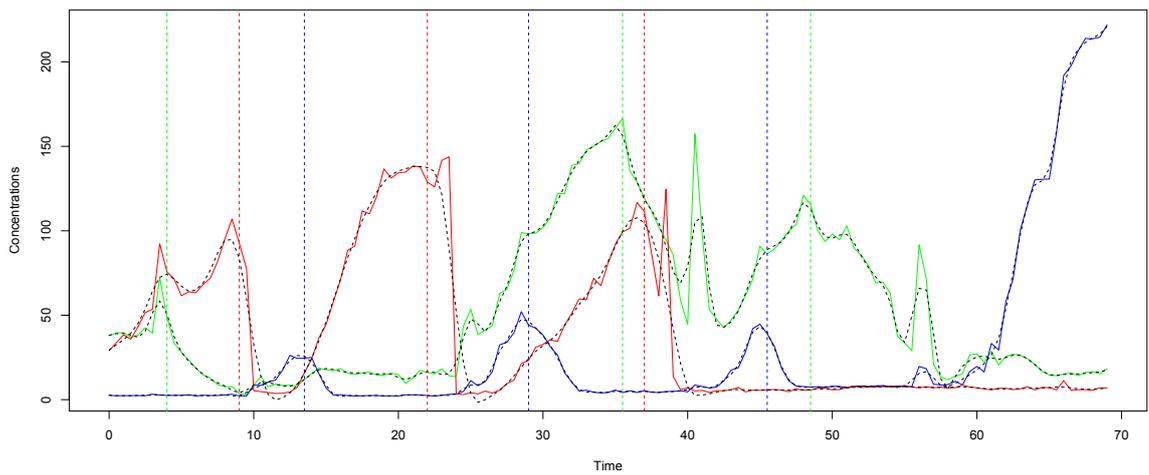


Figure 4.8: On this trace from Figure 4.7, peaks selected with the trace simplification procedure with a coefficient $C = 2$ are marked with vertical dashed lines.

An additional constraint in this formula ensures that peaks have an amplitude higher than 1, in order to avoid selecting peaks in some experimental traces where the overall amplitude is too low because there was no significant signal.

The distributions of peak-to-peak intervals measured on the whole data set are displayed in Figure 4.9. Although still slightly noisy, all three distributions exhibit a clear peak close to 20h, hence providing a reasonable estimation of the period for the rhythms of the cell cycle and the circadian clock in the cells.

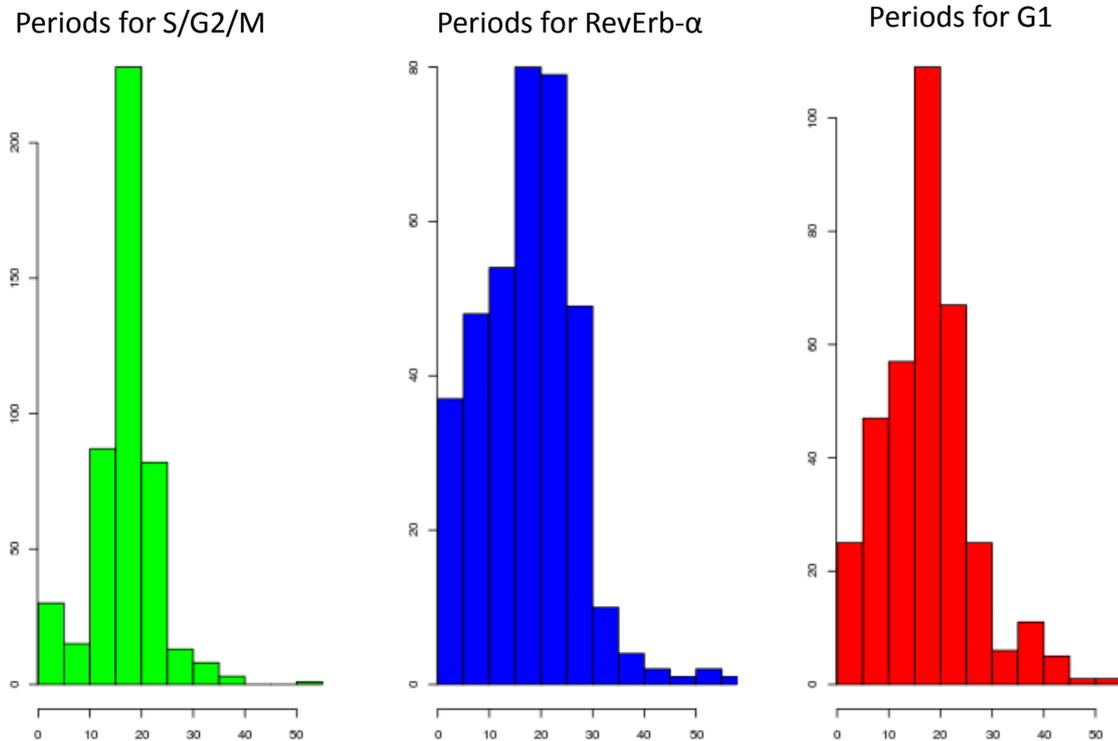


Figure 4.9: Distribution of periods captured from 627 experimental traces with temporal logic specifications after a trace simplification keeping only the main peaks with a coefficient $C = 2$.

4.5 Conclusion

Two approaches have been explored in this chapter to improve the performance in evaluating dynamical properties, with a focus on oscillatory properties: first-order patterns of formulae defined as relations associated with efficient dedicated constraint solvers, and a trace simplification procedure that abstracts the underlying data while preserving the results of analysis.

Both strategies attempt to find a compromise between generality and specificity in order to improve performance while being useful in as many situations as possible. In this perspective, the rich list of defined relations and affiliated dedicated solvers, and the possibility to combine them, allow to express diverse and complex dynamical properties regarding oscillatory behaviors.

Moreover, we have given some general conditions on the syntax of the formulae under which it is correct to keep in the trace only the time points corresponding to the local extrema of the molecules, or the crossing points between molecular concentrations. Although a tool allowing to automatically detect the compliance of a formula to these conditions would be useful and is still lacking, the trace simplification keeping local extrema in the trace is at least supported by all the

relations defined in Tables 4.2 to 4.5, except *incrInterv*, *IncreasingSwitch*, and *amplitude* when this last one is used with a third argument.

Both approaches facilitate the use of temporal logic by modelers, and its implementation in computational systems biology tools. The evaluation of complex dynamical behaviors on long traces is sped up by several orders of magnitude. With this improvement, it is now possible to apply temporal logic constraints in methods that require numerous simulations and evaluations, such as parameter optimization.

5. Application to the coupling between the cell cycle and the circadian clock

5.1 Introduction

Oscillatory systems in systems biology

The interest for oscillatory systems and the underlying circuits is of a prime position in systems biology. Indeed oscillations are a striking emerging property of biochemical networks, which also benefits from dedicated methods in the dynamical systems field. Depending on different conditions, such as binding affinity, degradation speed and temporal delays, the behavior of a certain network may differ, displaying oscillations or not, exhibiting different phase-times and complex behaviors such as harmonic behavior.

In contrast with physics, the notion of oscillation in biology is usually understood informally and in a wider scope. In wet experiments, oscillatory behaviors can be found with decreasing or increasing amplitude, and oscillations can be partly hidden in noisy time series. The calibration of oscillatory models on experimental data needs to focus on the most relevant properties, such as period and phase constraints, with flexibility on the value precision and other properties such as amplitude.

Robustness and flexibility

Different conditions and a diverse amount of interconnected processes have to be taken into account in order to explain properties of oscillatory systems. This makes computational and mathematical methods and analyses complex.

A number of general rules that enable and further facilitate oscillations in a mathematical model have been revealed by theoretical analysis [Novák and Tyson, 2008]. A negative feedback is the first requirement, in order to link the end of a cycle to its starting point, and the negative-feedback signal must be sufficiently delayed in time to avoid a steady state. This delay may be present in different forms. Most frequently, combinations of activating and inhibiting processes

apply a delay, and longer negative feedback loops are indeed more prone to oscillations than shorter ones. An additional feedback loop is another mechanism that can delay the negative-feedback signal. Not only does it facilitate the generation of oscillations, it can also bring robustness to the system [Gérard et al., 2012]. Another requirement for oscillations is a sufficient nonlinearity in the kinetic rate laws of the reaction mechanism. Indeed, nonlinear kinetics destabilize the steady state and in particular switch-like, ultrasensitive response functions within the negative feedback loop promote oscillations. Finally, the opposing reactions involved in the cycle must also occur on appropriate timescales.

The cell cycle and the circadian clock are two major cellular rhythms. The complexity of both systems and the number of regulators linking them to other physiological processes are such that there still remain many details to discover or refine. Nonetheless, the main interaction loops governing each rhythm have long been the subject of various fruitful modeling studies.

Circadian clock

Most organisms have developed an internal clock with a near 24-hour periodicity, allowing them to optimally tune metabolic, physiological, and behavioral functions such as the sleep-wake-phases succession, body temperature, blood pressure, hormone levels, to the needs they have at specific times during the day. Such circadian (from *circa diem*: "approximately one day") rhythms are defined by their persistence in the absence of any environmental cue (and are therefore self-sustained), their temperature-independence and their close approximation to the period of the earth's rotation [Pittendrigh, 1960]. Circadian clocks are found across all three kingdoms of life and are thought to have evolved multiple times during evolution [Dunlap et al., 1999]. A driving force for their apparition could be to segregate incompatible processes to different times of the day. For example, in the Cyanobacterium *Oscillatoria*, nitrogen fixation and photosynthesis, which are incompatible processes, are temporally separated by the circadian clock allowing both processes to occur in the same cell [Stal and Krumbein, 1987].

In mammals, the circadian clock relies on autonomous oscillators active in most cells of the organisms. A central clock in the suprachiasmatic nucleus (SCN) is sensitive to light and entrained by the day-night alternation. It synchronizes molecular clocks in peripheral tissues by hormonal signals (see Figure 5.1).

It has been shown that in absence of synchronization by a central clock, autonomous circadian oscillators are maintained in peripheral tissues with the same period, although they are progressively desynchronized [Yoo et al., 2004]. This has been confirmed in cultured cells from the standard mouse embryonic fibroblast cell line NIH3T3, first in [Nagoshi et al., 2004] and then in [Bieler et al., 2014] and [Feillet et al., 2014]. In each of those studies, confluent fibroblasts have a circadian clock period close to 24h regardless of the medium concentration.

The circadian oscillator is composed of several intertwined post-transcription feedback loops, with time delays characterized by intermediary reactions such as phosphorylations and complex formations, and their kinetic parameters.

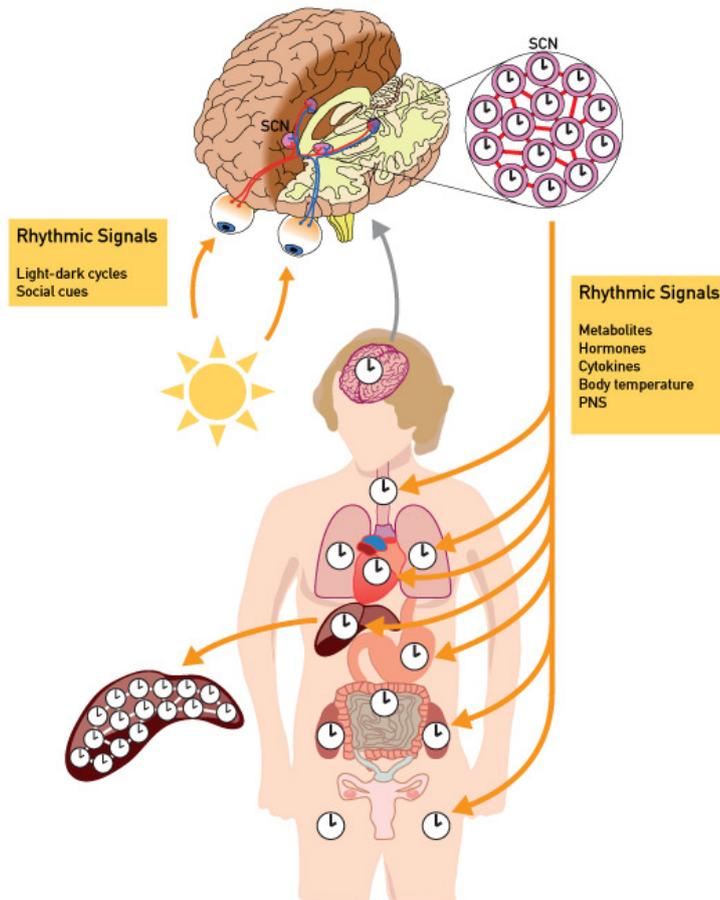


Figure 5.1: Schema of the central circadian clock in the SCN, and peripheral clocks, from [Bollinger and Schibler, 2014].

Previous view of the coupling between the cell cycle and the circadian clock

Integration of multiple molecular processes is fundamental to the living cell, and biochemical networks are commonly coupled with other molecular processes or external stimuli. It is not surprising that the cell cycle and the circadian clock, which are both periodic processes with a 24h period under some conditions, are coupled together. The "flight-from-light" hypothesis explains the temporal regulation of cellular divisions as a mechanism to avoid UV exposure during the process of DNA replication, when cells are most sensitive to UV.

Moreover, uncontrolled cell proliferation is one of the key features leading to cancer. Seminal works in chronobiology have revealed that disruption of the circadian timing system in mice, either by surgical, genetic, or environmental manipulation, increased tumor development. In humans, shift work is a risk factor for cancer. The toxicity and efficacy of many chemotherapeutic agents has been shown to depend on the time of administration [Lévi and Schibler, 2007]. This has triggered the design of chemo- and radiotherapeutical protocols, referred to as chronotherapy, that take into account the presence of the circadian system and deliberately make use of chronotoxic effects of radiation and pharmaceuticals [Levi et al., 2007].

Experimental results have shown a regulation of the cell division cycle by the circadian clock [Matsuo et al., 2003, Barnes et al., 2003, Ünsal-Kaçmaz et al., 2005, Hong et al., 2014], with possible applications to cancer chronotherapies [Ballesta et al., 2011, De Maria et al., 2011]. Molecular

links between these two cycles have been exhibited to explain this regulation. In particular the regulation of Wee1, an inhibitor of the G2/M transition, by the clock genes has been proposed to explain the circadian gating of mitosis during the liver regeneration process [Matsuo et al., 2003] and the 48h period doubling phenomena of the cell cycle [Nagoshi et al., 2004]. Other similar molecular links going in the same direction, through p21 [Gréchez-Cassiau et al., 2008] and Chk1 and Chk2 [Ünsal-Kaçmaz et al., 2005, Gery et al., 2006], have been shown in different cells in the literature. A few models have also been developed to further investigate those hypotheses, by coupling a model of the cell cycle with a model of the circadian clock through those direct molecular links, and analyzing the conditions of entrainment in period [Calzone and Soliman, 2006, Gérard and Goldbeter, 2012].

Recent paradoxical experimental observations

However, in mouse embryonic fibroblasts NIH3T3, several studies using large-scale time-lapse microscopy to monitor circadian gene expression and cell division events in real time and in individual cells during several days have unveiled unexpected behaviours, hinting that the relationship might be more complex. [Nagoshi et al., 2004] have first shown that circadian gene expression in fibroblasts continues during mitosis, but with a consistent pattern in circadian period variation relatively to the circadian phase at division, leading them to hypothesize that mitosis elicits phase shifts in circadian cycles.

A more recent study of [Bieler et al., 2014] reporting similar experiments on dividing fibroblasts found the two oscillators synchronized in 1:1 mode-locking leading the authors to hypothesize a predominant influence of the cell cycle on the circadian clock in NIH3T3 cells. This is in agreement with another detailed experimental study of [Feillet et al., 2014], which found several different synchronization states in NIH3T3 fibroblasts in different conditions of culture. In particular, it was observed in [Feillet et al., 2014] that enriching the milieu with Foetal Bovine Serum (FBS) not only accelerates the cell division cycle but also the circadian clock. For cells cultured in 10% FBS, both distributions of the cell cycle length and the circadian clock are centered around 22h. For cells cultured in 15 % FBS, both the cell cycle and the circadian clock accelerate, with period distributions centred around 19h. However, when cells reach confluence and stop dividing, the circadian clock slows down and the period distribution is then centred around 24h. None of the currently available models coupling the cell cycle and the circadian clock can explain these observations since they are based on an unidirectional influence of the circadian clock on the cell cycle [Gérard and Goldbeter, 2012, Calzone and Soliman, 2006].

In this chapter, in order to explain these observations, we study the reverse influence of the cell cycle on the circadian clock, using computational modeling tools. We develop a mathematical model of the influence of the cell cycle on the circadian clock through the regulation of clock genes triggered by mitosis, and investigate the conditions in which the cycles are entrained in period and phase as observed in [Feillet et al., 2014]. For this, we use the circadian clock model of [Relógio et al., 2011], which has been carefully fitted to phase data on suprachiasmatic cells, and a simple model of the cell cycle defined in [Calzone and Soliman, 2006] which focuses on the mitosis phase.

5.2 Experimental data and their formal specification in temporal logic

Experimental observations and measurements

In this section we describe the single cell experiments and analyses performed in [Feillet et al., 2014] and the conclusions drawn by the authors. The reported experiments have been done using

time lapse videomicroscopy and cell tracking using various fluorescent reporters for the cell cycle and the circadian clock observed during 72 hours in proliferating NIH3T3 mouse fibroblasts.

The NIH3T3 embryonic mouse fibroblasts were modified to include three fluorescent markers of the circadian clock and the cell cycle: the RevErb- α ::Venus clock gene reporter for measuring the expression of the circadian protein RevErb- α , and the Fluorescence Ubiquitination Cell Cycle Indicators (FUCCI), Cdt1 and Geminin, two cell cycle proteins which accumulate during the G1 and S/G2/M phases respectively, for measuring the cell cycle phases.

The cells were left to proliferate in regular medium supplemented with different concentrations of FBS (10% and 15%). Long-term recording was performed in constant conditions with one image taken every 15 minutes during 72 hours. The lengths of the cell cycles were measured as the time interval between two consecutive cell divisions.

The expression traces of RevErb- α proteins were detrended and smoothed. Spectrum resampling was used to estimate the clock period. Cells with less than two RevErb- α peaks within their lifetime, a period length outside the interval between 5 hours and 50 hours or a relative absolute error (RAE) bigger than 0.25 (showing a confidence interval wider than twice the estimated period) were classified as non-rhythmic and discarded, assuming that they do not have a functioning clock. Finally, the delay between mitosis and the next clock marker peak was measured. It revealed that RevErb- α peaked slightly more than 5 h after cell division in all conditions, consistently with [Bieler et al., 2014], and as also reported in [Feillet et al., 2015].

Medium	Clock period	Division period	Delay
FBS 10%	21.9h \pm 1.1h	21.3h \pm 1.3h	8.6h
FBS 15%	19.4h \pm 0.5h	18.6h \pm 0.6h	7.1h

Table 5.1: Estimated periods of the circadian molecular clock and the cell division cycle measured in fibroblast cells without treatment by Dexamethasone, for two concentrations of FBS [Feillet et al., 2014]. The time delay is between the cell division time and the next peak of RevErb- α .

The quantitative data on the periods of the cell cycle and the circadian clock and the phase between them are summarised in Table 5.1 [Feillet et al., 2014]. Surprisingly, increasing FBS from 10% to 15%, not only decreases the mean period of the cell cycle from 21.9h to 19.4h, but also the clock period from 21.3h to 18.6h, i.e. to essentially the same period. This shows that both oscillators remain unexpectedly in 1:1 mode locking. While the speedup of the cell cycle can be directly attributed to the growth factors in increasing concentration of FBS, it can not account for the speedup of the clock the same way, since confluent cells keep a 24-hours period for the circadian clock independently of the FBS concentration.

Experimental observations after treatment by dexamethasone

Furthermore, a series of experiments were done with a pulse of dexamethasone (Dex) before recording. This glucocorticoid agonist is known to exert a resetting/synchronizing effect on the circadian molecular clocks in cultured cells through the induction of *Per1*. In that case, cells were incubated for 2 hours in the same medium supplemented with Dex, just before returning to a Dex-free medium for the recording.

In that case, the results are more complex. As summarized in Table 5.2, which contains values from [Feillet et al., 2014], the cells in 10% FBS show an increased clock period and a low cell cycle period, with an overall ratio of 5:4. In 20% FBS the cell lineages are dominated by two groups. The first group shows close periods, i.e. a 1:1 mode-locking similarly to the experiments without dexamethasone. The second group shows a high clock period and a fast cell cycle, with an overall ratio close to 3:2 between the clock and cell cycle, explaining the three-peaks distribution of the circadian phase at division, as already observed ten years before by [Nagoshi et al., 2004]. It has to be noted that the 20% FBS dexamethasone-synchronized experiment was repeated with

Medium	Clock period	Division period	Delay
FBS 10%	24.2h \pm 0.5h	20.1h \pm 0.94h	10.7h
FBS 20%	21.25h \pm 0.36h 29h \pm 1.05h	19.5h \pm 0.42h 16.05h \pm 0.48h	8.3h 6h/12h/22h

Table 5.2: Estimated periods of the circadian molecular clock and the cell division cycle measured in fibroblast cells after treatment by Dexamethasone, for two concentrations of FBS [Feillet et al., 2014]. The time delay is between the cell division and the next peak of RevErb- α . The experiment done with 20% FBS have been clustered by the authors in two groups with different periods.

similar results available in the Supplementary Information of [Feillet et al., 2014], although the distribution of the period ratios for the second group is wider in the interval [1.2, 2].

In [Feillet et al., 2014], the authors suggest that these observations might be interpreted by the existence of distinct oscillatory stable states coexisting in the cell populations, in particular with 5:4 and 1:1 phase-locking modes for the condition 10% FBS, and 3:2 and 1:1 phase-locking modes for the condition 20% FBS, and that the dexamethasone could knock the state out of the 1:1 mode toward other attractors. A mechanistic explanation remains to be found to support this interpretation.

Formal specification of oscillation properties in quantitative temporal logic

Based on the experimental data from [Feillet et al., 2014] that we have presented, and before detailing the mathematical models used to simulate the system, we formalize here, with temporal logic specifications, the oscillatory properties that simulations will have to satisfy.

Setting versatile constraints on the amplitudes and regularity of the simulated peaks will allow filter out modeling choices that result in incorrect simulations, keeping only simulated traces with sustained oscillations even with small irregularities, as it is the case for example in Fig. 5.12.

More precisely, we use the temporal logic formula patterns described in Chapter 4.2 to specify the constraints about the successive peaks of concentrations between either the same molecular species (period constraints) or different molecular species (phase constraints).

For instance, the following command computes the validity domain for a formula pattern used to extract the period of MPF (a component of the cell cycle, see the detailed model in Section 5.3) and RevErb- α for the circadian clock (again, see Section 5.3) in a trace, and their relative phase:

```
biocham: validity_domain(
Exists([e1,e2,e3],
  periodErrors([RevErb_nucl],[periodselect,e1,e2,e3],100)
  & e1<2 & e2<2 & e3<2))
& period([MPF],[periodMPF])
& phase([MPF,RevErb_nucl],[phase])).
```

The result for the simulation trace obtained with the kinetic parameter $k_{die}=0.25$ and displayed in Fig. 5.10 is

```
periodselect = 17.7763, periodMPF = 18.066, phase = 2.09856
```

The period constraint on the oscillations of RevErb- α is expressed by the temporal logic function *periodErrors*, whose validity domain provides the mean of the last two RevErb- α peak-to-peak intervals in the variable *periodselect*, along with several variables characterizing irregularity features of the trace, after a transient time of 100h to avoid irregularities caused by the initial state: *e1* for the irregularities in distances between peaks (it denotes the maximum difference between

two intervals), $e2$ for the irregularities in the amplitudes of the peaks (it quantifies the differences between the amplitudes of the peaks), and $e3$ being a non-null error if the concentration amplitude is too small (below 0.1). Setting thresholds on these variables ensures that irregular traces are filtered out. The operator *Exists* projects the resulting validity domain on the single dimension for *periodselect*. Since the trace of MPF shows sustained and regular oscillations in all simulations, the simple function *period* is used to extract the mean of the last two peak-to-peak intervals. Moreover, the function *phase* captures the mean of the last two time intervals between MPF and RevErb- α peaks.

On the other hand, it is worth remarking that for the purpose of exploring the parameter search, irregular traces should not be filtered out in order to orient the search algorithm in a promising direction, when oscillations begins to appear for instance. This can be specified in temporal logic with the following constraint:

```
biocham: add_search_condition(
periodErrors([RevErb_nucl],[period,e1,e2,e3],100) &
Exists([phase],
  phase([MPF,RevErb_nucl],[phase])
  & phase>minphase & phase<maxphase)
,[period,e1,e2,e3,minphase,maxphase],[21.3,0,0,0,3,5.5],
300,[kdie,kdie21]).
```

For each set of parameter values tested during the calibration procedure, this command computes the euclidian distance between the values found for $[period,e1,e2,e3,minphase,maxphase]$ and the objectives $[21.3,0,0,0,3,5.5]$, in the condition $kdie=kdie21$ and a simulation time of 300h. This distance is used as a score for the satisfaction of the temporal logic formula on the trace. The additional variables *minphase* and *maxphase* allow to score the distance from the phase to the objective interval $[3,5.5]$, enabling some flexibility in the searched value for the phase. For instance, the best set of parameters found as a solution after the calibration procedure detailed in 5.4 has the score 0.54 for this specification.

The temporal logic language allows us to combine the variables, for example with the difference between the periods of the cell cycle and the circadian clock, which can then be used to score the entrainment in period of the circadian clock by the cell cycle. This is illustrated by the following command:

```
biocham: satisfaction_degree(
Exists([p1,p2],Exists([e1,e2,e3],
  periodErrors([RevErb_nucl],[p1,e1,e2,e3],100)
  & e1<3 & e2<3 & e3<3)
  & period([MPF],[p2]) & diff=p2-p1),
[diff],[0],300).
```

This specification will be used to compute the satisfaction degrees displayed in Fig. 5.9 in Section 5.4. Each value scores the difference between the two periods with a value between 0 (infinite difference) to 1 (null difference, perfect entrainment).

5.3 Cell Cycle and Circadian Clock Models

Model of the cell cycle

As already explain in Chapter 2, the cell cycle of mammalian cells is composed of five phases: the quiescent phase G0 where cells can stay without dividing, the growing phase G1 for entering the cell cycle, the DNA replication phase S, the gap phase G2, and the chromosome segregation and mitosis phase M phase. Each phase is characterized by a particular protein of the cyclin family, which forms a complex with a cyclin-dependent-kinase (cdk) and determines the activity of the phase.

Mouse embryonic fibroblast cells considered in this chapter are quickly dividing cells. However, these cells also reach confluence and the G0 quiescent phase when they have no more space to divide.

For our purpose of investigating the hypothesis of a regulation of clock genes triggered by mitosis, it is sufficient to use a cell cycle model focusing on the mitotic phase. [Qu et al., 2003] have proposed a simple generic mathematical model of a cell cycle signaling network in higher eukaryote in which the cycle is divided in two different phases, and that could be used to simulate both the G1/S and G2/M transitions. This model inspired a model defined in [Calzone and Soliman, 2006], describing the G1/S/G2 phase and the M phase, and that we will reuse for this work. Of course, more detailed models distinguishing the four phases of the cell cycle exist, e.g. [Gérard and Goldbeter, 2012], making possible to represent various regulations of the cell cycle by the circadian clock genes, for instance through p21 and c-Myc on G1, and Wee1 on the G2/M transition. However, since the consequences of these regulations have not been observed in the experimental data considered in this chapter, those extra details are irrelevant and we concentrate on the reverse effect of the cell cycle on the circadian clock by the regulation of clock genes triggered by mitosis, for which the simpler two-phase model of [Calzone and Soliman, 2006] adapted from [Qu et al., 2003] is sufficient.

In this model, the M phase is triggered by the complex Cdk1/Cyclin-B. This complex appears in two forms, an active form called MPF (M-phase Promoting Factor) and a phosphorylated, inactive form called preMPF. MPF is phosphorylated and inactivated by the kinase Wee1, and dephosphorylated and activated by the phosphatase Cdc25. Both the kinase and phosphatase activities are themselves inactivated and activated by MPF, respectively.

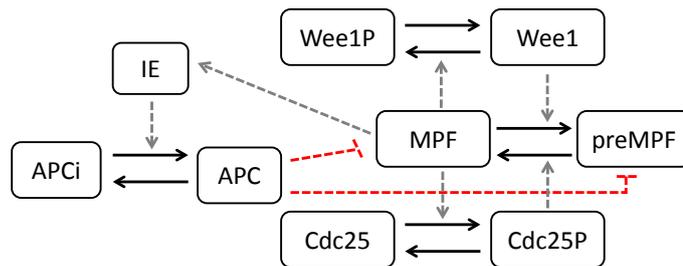


Figure 5.2: Schema of the cell cycle model of [Calzone and Soliman, 2006]. Solid arrows represent biochemical reactions while dashed arrows denote enzyme catalysis. Red arrows denote an inhibition through activation of the degradation.

In this model, we assessed the effect of the different reaction rate constants on the period of the cell cycle by sensitivity analysis. We found that two parameters are able to change widely the range of the cell cycle period without changing significantly the strength of the coupling: k_{die} , the degradation rate constant of the intermediary enzyme involved in the negative feedback loop between MPF and the proteasome APC, particularly important in G1/S, and k_{ampf} , the activation rate constant of MPF by Cdc25p, which plays a role in G2/M. In the supplementary material of [Feillet et al., 2014], both the phases G1 and S/G2/M seem to be shortened in enriched FBS. Therefore we arbitrarily choose to modulate the cell cycle period by varying k_{die} . Similar results are obtained with k_{ampf} .

A simple parameter search gives the following values for k_{die} : 0.147 for a cell division period of 21.3 hours (corresponding to 10% FBS), 0.23 for a period of 18.6 hours (15% FBS).

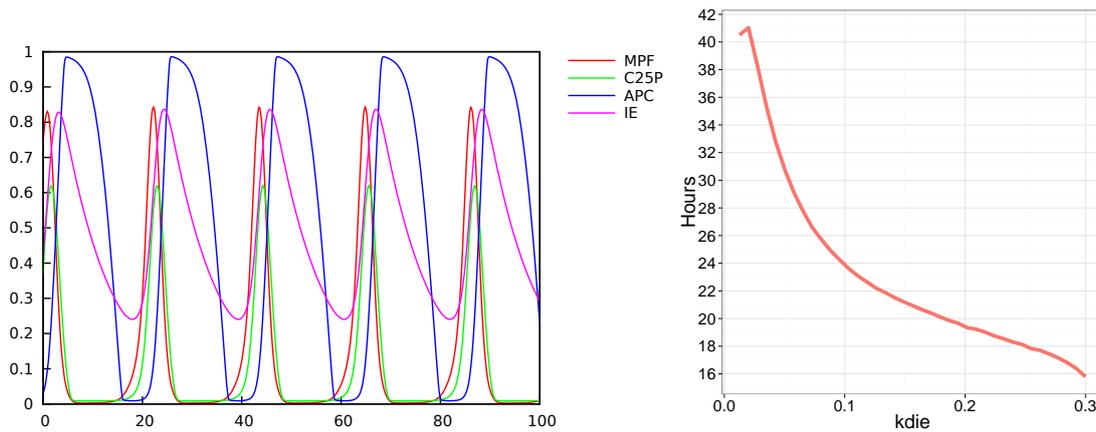


Figure 5.3: **Left:** Simulation of the cell division cycle model of [Calzone and Soliman, 2006] **Right:** Period of the cell division cycle (measured as the distance between successive peaks of MPF) as a function of the parameter k_{die} .

Model of the circadian clock

The core network driving the mammalian circadian clock

Two major transcription factors, *Clock* and *Bmal1* heterodimerize and activate the transcription of the period (*Per1* and *Per2*), cryptochrome (*Cry1* and *Cry2*), *RevErb- α* and *Ror* clock genes. The newly formed *Per* and *Cry* proteins associate and inhibit their own expression and that of the *RevErb- α* and *Ror* proteins through direct inhibition of the *Clock/Bmal1* transcriptional activity. Furthermore, the antagonistic *RevErb- α* and *Ror* transcription factors regulate the rhythmic transcription of *Bmal1* and *Clock*. These interlocked negative feedback loops generate robust 24h self-sustained oscillations, which in turn control the expression of a large set of downstream clock-controlled genes.

Models of the circadian clock

The circadian clock has usually been modeled with ODEs [Leloup and Goldbeter, 2003, Relógio et al., 2011], or in a few cases using explicit time delays with delayed differential equations (DDEs) [Korenčič et al., 2014]. Finally, a few simple discrete models have been published with qualitative predictions [Comet et al., 2012].

So far, focus is set on the modelling of the core-clock in an attempt to elucidate its characteristic properties, such as self-sustained oscillations, circadian period, robustness, synchronization among cells, temperature-compensation and entrainment to external inputs. Other models focus on investigating alterations in clock properties upon perturbations of model parameters, allowing the simulation of pathological situations where the clock is known to be deregulated [Relógio et al., 2014]. More comprehensive models have also been used to simulate the circadian network at a systems level, by including clock-controlled genes (CCGs) [Korenčič et al., 2014]. Indeed, a large fraction of genes from a variety of pathways is subject to circadian regulation, usually in a tissue-specific manner.

In this dissertation we use the circadian clock model of [Relógio et al., 2011] which has been fitted on mouse suprachiasmatic cells with precise data on the amplitude and phases of the different components. This model, schematized in Figure 5.4, is composed of 20 species, 71 parameters, and all the feedback loops described above.

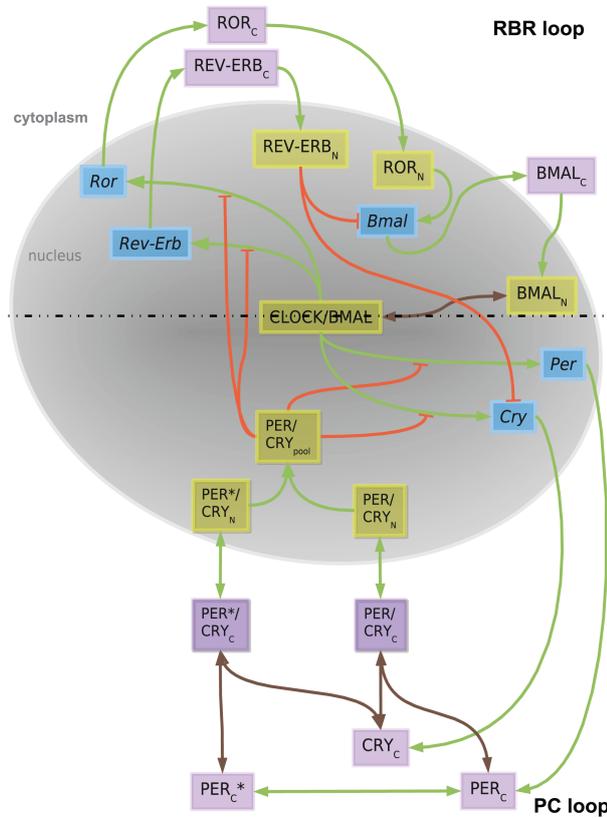


Figure 5.4: Schema of a model of the mammalian circadian clock from [Relógio et al., 2011].

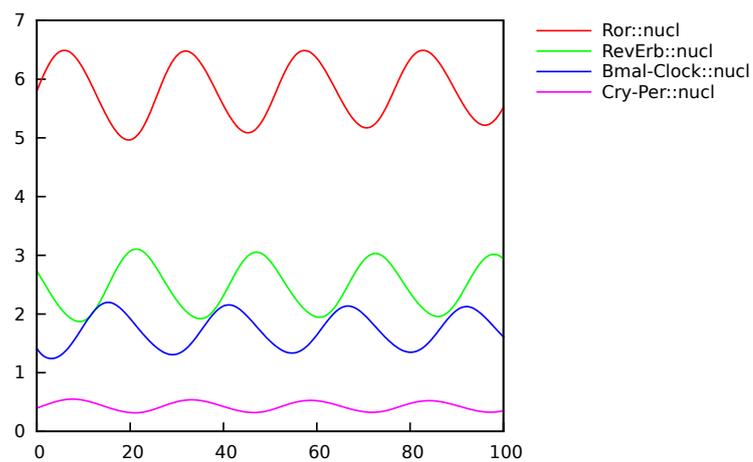


Figure 5.5: Simulation of the Circadian Clock model of [Relógio et al., 2011].

5.4 Coupled Model

Hypothesis of a Selective Regulation of Clock Gene Transcription triggered by Mitosis

In this section, we investigate the hypothesis that a differential inhibition/activation of the clock genes could explain the observations made and reproduce consistent values for the period of the circadian clock, and the delay between the divisions and RevErb- α peaks for the different values of FBS.

To assess this hypothesis, we model the inhibition or activation of clock genes transcription triggered by mitosis with five multiplicative coefficients I_i ($i \in \{1, \dots, 5\}$), associated with the synthesis rate parameters of the model for each of the five clock genes. Each coefficient takes the value 1, except during a window starting at mitosis where its value is changed with an event, triggered by the decrease of MPF. Another event is triggered at the end of the inhibition window to reset the coefficient. During this window, whose length is defined by the parameter *duration*, the coefficient for circadian core gene i takes the value of the dimensionless parameter $coef_{synth_i}$, that defines the inhibition/activation strength of this clock gene. This value is included in the real interval $[0, 3]$, where 0 denotes a full inhibition (i.e., strong coupling), 1 marks no effect of the mitosis on the synthesis (no coupling), and more than 1 induces some activation (coupling again). The value of *duration* is also considered in the real interval $[0, 3]$ hours.

Our coupled model of the cell cycle and the circadian clock thus uses six parameters: the regulation strengths of the clock genes ($coef_{synth_i}$, $i \in \{1, \dots, 5\}$), and the duration of the regulation. The implementation of the coupling in Biocham is shown in Listing 5.1.

```
parameter(duration,2).
parameter(endMitosis,0).

% For each clock gene i:
parameter(I_i,1).
parameter(coefsynth_i,0).

add_event([MPF]<0.5,endMitosis,Time+duration).
add_event([MPF]<0.5, I_i, coefsynth_i).

add_event(Time>=endMitosis, I_i, 1).
```

Listing 5.1: Implementation of the regulation of the circadian clock by the cell cycle in Biocham.

It is worth noting that this way of coupling the models enforces the fact that for quiescent cells, whatever the FBS concentration, the transcription rate will be unaffected by mitosis, and therefore the clock will keep a period of 24h as observed in the experiments.

Note also that since the only coupling is via a periodic activation/inhibition, and since the phases between the peak of MPF and that of RevErb- α are fixed in the experimental data, looking for a similar effect, triggered by another cell-cycle event but constrained to the same time window with a delay mechanism, would in principle lead towards similar results.

Search of coupling parameters

We perform a parameter search procedure in order to find a set of values for the coupling parameters that reproduces the entrainment in period and phase observed in the data. Using the specification detailed in the section 5.2, we define a multi-condition objective: in the conditions $kdie=0.1$, $kdie=0.147$, $kdie=0.18$ and $kdie=0.23$, the period of the circadian clock must be equal to the period of the cell cycle: respectively 24h, 21.3h, 20h and 18.6h. In each condition, the delay between MPF and RevErb- α peaks must be between 6.5h and 8.6h. Several different results are

found, depending on the starting point of the parameter search. Two distinct solutions are shown in Table 5.3.

Parameters	First set	Second set
Synthesis coefficient for <i>Per</i>	0.66	2.40
Synthesis coefficient for <i>Cry</i>	2.30	0.67
Synthesis coefficient for <i>RevErb-α</i>	1.04	1.92
Synthesis coefficient for <i>Ror</i>	2.1	1.51
Synthesis coefficient for <i>Bmal1</i>	0	0.78
Duration	2.97h	2.81h

Table 5.3: Two sets of parameter values found by the calibration procedure. The first set was found with null initial values for all synthesis coefficients, while the second was found with initial values of 1.

If the parameter search starts from a full inhibition triggered by mitosis for all clock genes, corresponding to initial values of 0 for all synthesis coefficients, the best result found after 260 iteration on a population of 95 sets of parameters gives the first set of parameters reported in Table 5.3. Three restarts occur during the search because of slow convergence: initial values for the search are generated again to move the search in another region of the parameter space.

This solution corresponds to a full inhibition found for the transcription of *Bmal1* (null value for the synthesis coefficient for *Bmal1*), and a smaller inhibition for the transcription of *Per* (value 0.66). The transcription of *Cry* and *Ror* are activated with a rough two-fold change (2.30 and 2.1), while the transcription of *RevErb- α* is mostly unaffected (value 1.04, close to 1).

However, when the initial values are 1 for all synthesis coefficients, corresponding to no inhibition during mitosis, the second set is the best result, found after 60 iterations on a population of 95 sets of parameters. The convergence of the violation degree of the temporal logic specification taken as fitness cost during the optimization procedure is seen in Figure 5.6.

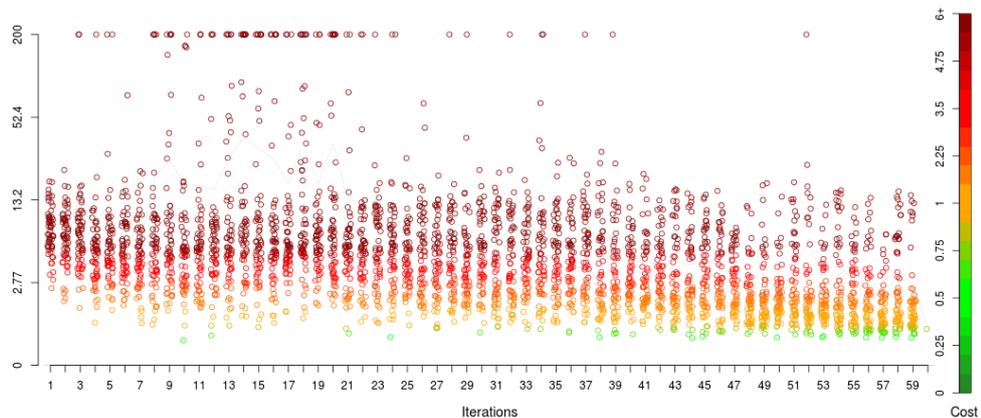


Figure 5.6: Fitness cost of the successive populations generated for each iteration of the optimization procedure with CMA-ES, with log scale. The result of this parameter search is the second set of parameter values in Table 5.3.

Here, the transcriptions of *Bmal1* and *Cry* are weakly inhibited, while the transcription of the other clock genes are activated.

The simulation of the model with any of these two sets of parameters shows a delay between the starting time of the mitosis effect and the circadian clock consistent with the experimental data (close to 7h for the first set, and between 7 and 8.5h for the second one).

However, as we will see in the following of the analysis, the first solution yields more consistent results for the time between mitosis and the next *RevErb- α* peak in cells where the cell cycle is slow. Therefore we first focus on this solution in this section. The second solution found is further discussed in section 5.5.

Assessing the effect of the inhibition of each clock gene in the coupling

The parameter search returns a value for each coupling parameter, however it is important to assess which of these values are necessary for the correct entrainment, and which values have no impact on the coupling.

The response curves for the period of the circadian clock and the delay between mitosis and the next *RevErb- α* peak for each calibration parameter, in the condition where the cell cycle has a period of 21h, are useful to assess the role of each clock gene for the entrainment of the circadian clock by the cell cycle. The results are shown in Fig. 5.7.

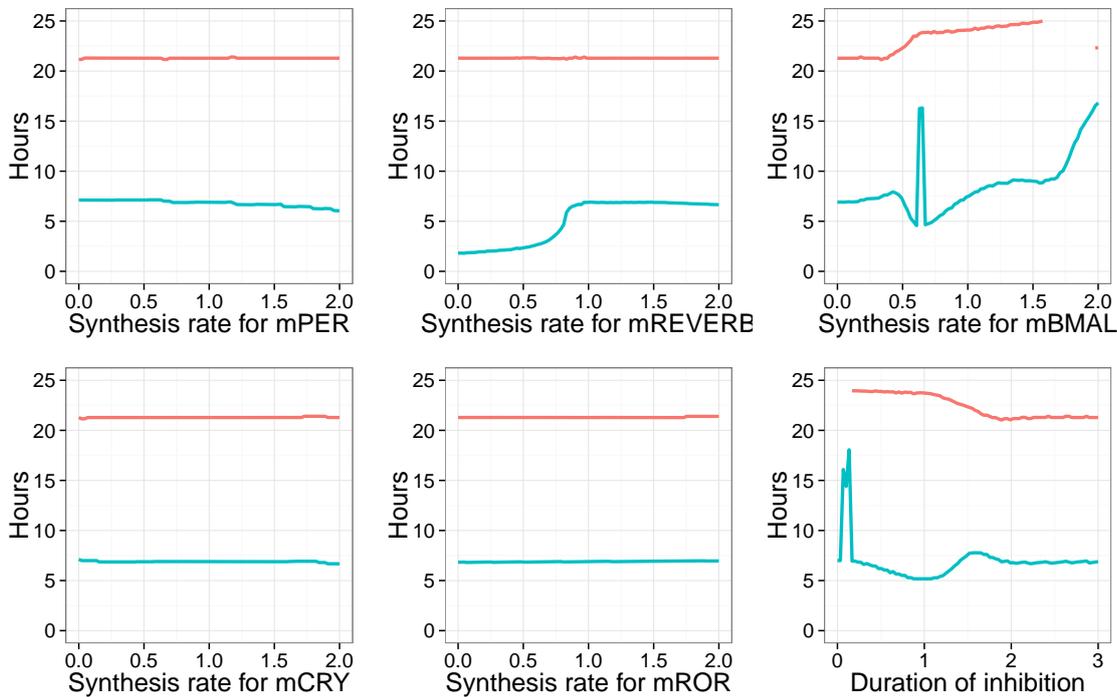


Figure 5.7: Period of the circadian clock (red) and time delay between MPF and *RevErb- α* peaks (blue) in the coupled model calibrated with the first set of parameters, and when the cell cycle has a period of 21h. The peaks on the blue curves on the right panels characterize irregular oscillating traces (due to a partial entrainment in period). Synthesis rates are normalized to exhibit the effect of the applied coefficient (0: full inhibition, 1: no effect, 2: two-fold activation).

The sets of simulations reveal that the entrainment in period and phase of the circadian clock depends only on the effect of the mitosis on *Bmal1* and *RevErb- α* , and the duration of this effect. Varying the synthesis coefficients during mitosis for *Per*, *Cry* or *Ror* has no significant effect on the entrainment.

The inhibition of *Bmal1* is crucial for the entrainment in period. More specifically, the clock is entrained to the cell cycle period of 21h if the coefficient multiplied to the synthesis rate of *Bmal1* is at most 0.4, and the inhibition lasts at least 2h. Interestingly, the inhibition on *RevErb- α* has no effect on the period of the circadian clock, although it has an effect on the circadian phase at division. With a low or absent inhibition of *RevErb- α* (corresponding to a coefficient higher than 0.8), the time delay between divisions and *RevErb- α* peaks is consistent with the data,

with RevErb- α peaks occurring 7h after division. Adding an inhibition of RevErb- α triggered by mitosis would conserve the entrainment of the circadian period, but the RevErb- α peaks occur just after mitosis.

These new simulations point toward a new hypothesis of a selective inhibition of the circadian clock gene *Bmal1* triggered by mitosis. In the rest of the chapter, the coupling parameters are simplified to consider a single inhibition of *Bmal1* (corresponding to a coefficient equal to 0), and no effect on the other clock genes.

Result for the selective inhibition of *bmal1* during mitosis

<i>kdie</i>	FBS %	Circadian clock period (h)	Cell division period (h)	Phases (h)
0.077	5?	26.09	26.10	23.2
0.147	10	21.28	21.28	6.8
0.229	15	17.99	18.60	6.3

Table 5.4: Periods and time delays measured in the coupled model with different values of *kdie* for modeling the different culture conditions (the correspondence with 5% FBS is speculative since no experiment was done in this condition). The delays are the time observed by simulation between the peaks of concentration of MPF and RevErb- α .

Table 5.4 shows the periods of the circadian clock and the cell division cycle and the delay between the starting time of the inhibition of *Bmal1*, when the peak of MPF overtakes the threshold 0.5, and the following peak of RevErb- α in our model with different values of *kdie* corresponding to the different culture conditions. In all cases, the cell division manages to entrain the circadian clock (that has a free period around 24h) to its period, simply through this mechanism of selective transcription inhibition, as depicted in Fig. 5.8. These simulation results reproduce quite well the data of Table 5.1 when there is no treatment by Dex. Note that our model can also have a cell division time higher than 24h, for instance with *kdie* = 0.077, which might correspond to a concentration of FBS around 5%. In that case the model predicts that the cell cycle will still entrain the circadian clock, lowering its period. Moreover, RevErb- α peaks are predicted to occur slightly before cell divisions in this condition, with a delay of 23.2h between the inhibition of *Bmal1* and the following peak of RevErb- α .

As shown in the right panel of Fig. 5.3, it is possible to simulate the experimental milieu enrichment with 10 or 15% FBS by varying the parameter *kdie* of the cell cycle model to obtain correct values for the period of the cell division cycle.

The landscape in Fig. 5.9 is computed to assess the role of the inhibition or activation duration. It shows the variation of the difference between the periods of RevErb- α for the circadian clock and MPF for the cell cycle when the two parameters *kdie* and *duration* vary. The value of each period is captured with a temporal logic specification as seen in the subsection 5.2. The result for the inhibition of *Bmal1* is shown in Fig. 5.9, and is very similar to the result in the case of an activation of RevErb- α .

Three domains can be distinguished in this parameter space: in the domain in yellow, the circadian clock is entrained to the same period as the cell cycle. This domain of entrainment is wider for a long duration of inhibition. For a short duration, the circadian clock can only be entrained by the cell cycle if the entraining period is close to 24h. In the purple domain at the bottom (for a low value of *duration*), the difference between the two periods is high because the clock is not entrained, hence it keeps its period constant and close to 24h. Finally, these two domains are separated by a black domain where the clock oscillations are partially entrained and become irregular.

One can notice that the longer the inhibition of *Bmal1*, the wider the range of values of *kdie* over which the circadian clock can be entrained. In particular, the clock can be entrained by the

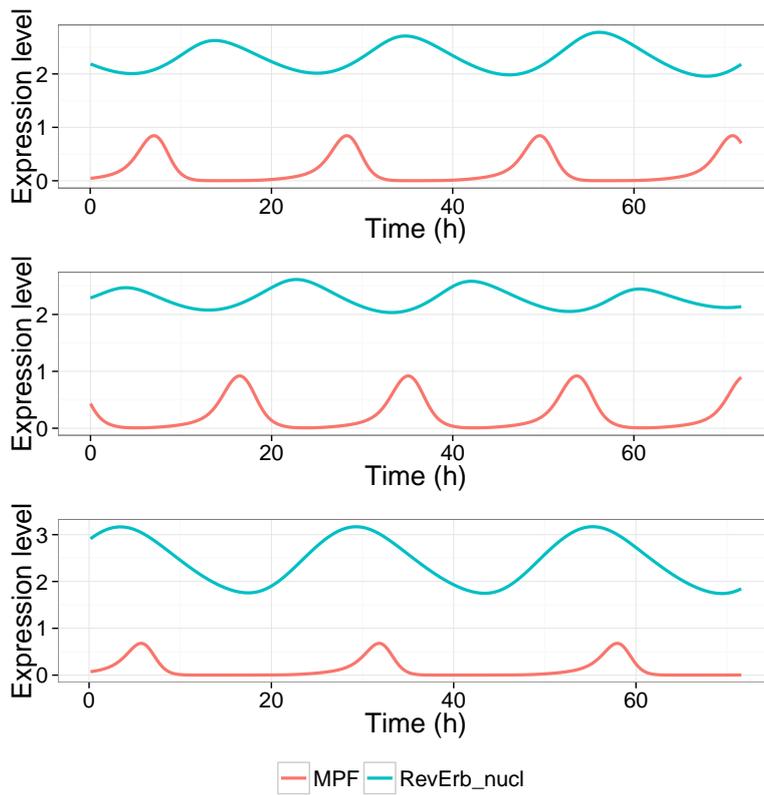


Figure 5.8: Simulation of the model with the inhibition of the transcription of *Bmal1* triggered by mitosis, during 72h. Top: the cell cycle has a period of 21.3h. Middle: the cell cycle has a period of 20.1h. Bottom: : the cell cycle has a period of 26h.

cell cycle when $k_{die} = 0.23$, corresponding to the smallest period (18h) reported in the data, if the duration of the inhibition is at least 3h.

The entrainment both in period and phase with an inhibiting effect during 3h is visualized in Fig. 5.10. It shows the response curve for the periods of the cell cycle and the circadian clock, and the time delay between the peaks of MPF and RevErb- α in the simulations when the parameter k_{die} varies.

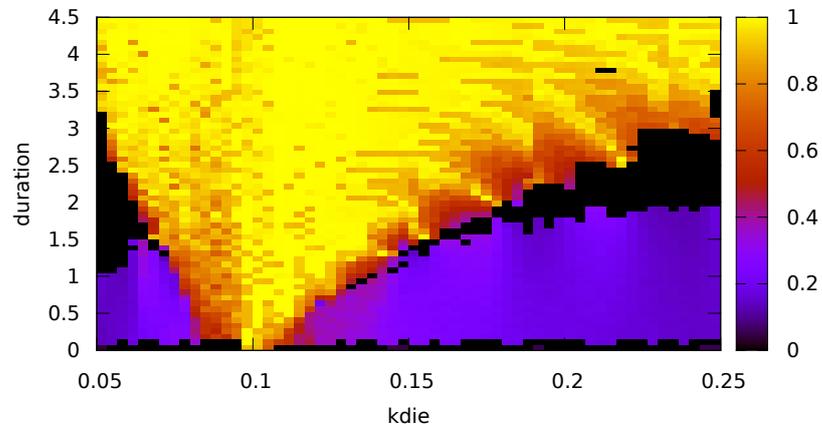


Figure 5.9: Absolute difference between the periods of the circadian clock and the cell cycle, as a function of $kdie$ for varying the cell cycle period, $duration$, the duration of the inhibition of *Bmal1* transcription triggered by mitosis. The landscape is computed as the satisfaction degree of the third formula detailed in 5.2. The color translates the distance from the value found for the period difference $diff$ to the objective 0. Full satisfaction in yellow indicates equal periods for MPF and RevErb- α , while the other colours indicate the absolute difference. Black indicates an absence of result for the specification, meaning that the regularity constraints set on the trace of RevErb- α with the function *periodErrors* were not met.

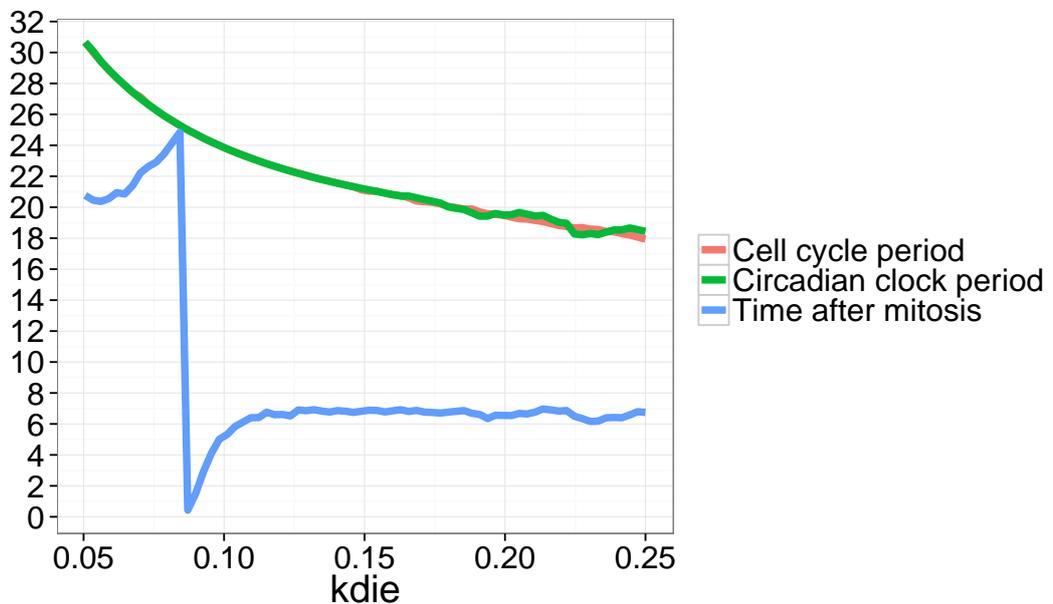


Figure 5.10: Entrainment in period and phase of the circadian clock when the period of the cell cycle varies with the parameter $kdie$, with the inhibition of *Bmal1* triggered by mitosis. The blue curve depicts the time delay between peaks of MPF and RevErb- α in the simulations.

Predictions on the phases in the clock

Data on the phases between clock components in proliferating cells are sparse. The model allows us to investigate whether the coupling from the cell cycle affects the phases between clock mRNAs and proteins. The following in-silico experiment is performed to this end: in the coupled model with a fast cell cycle (21h), the strength of the inhibition of *Bmal1* is changed in a set of simulations and the phases between clock components are captured in each simulation, normalised by the clock period, as shown in Fig. 5.11.

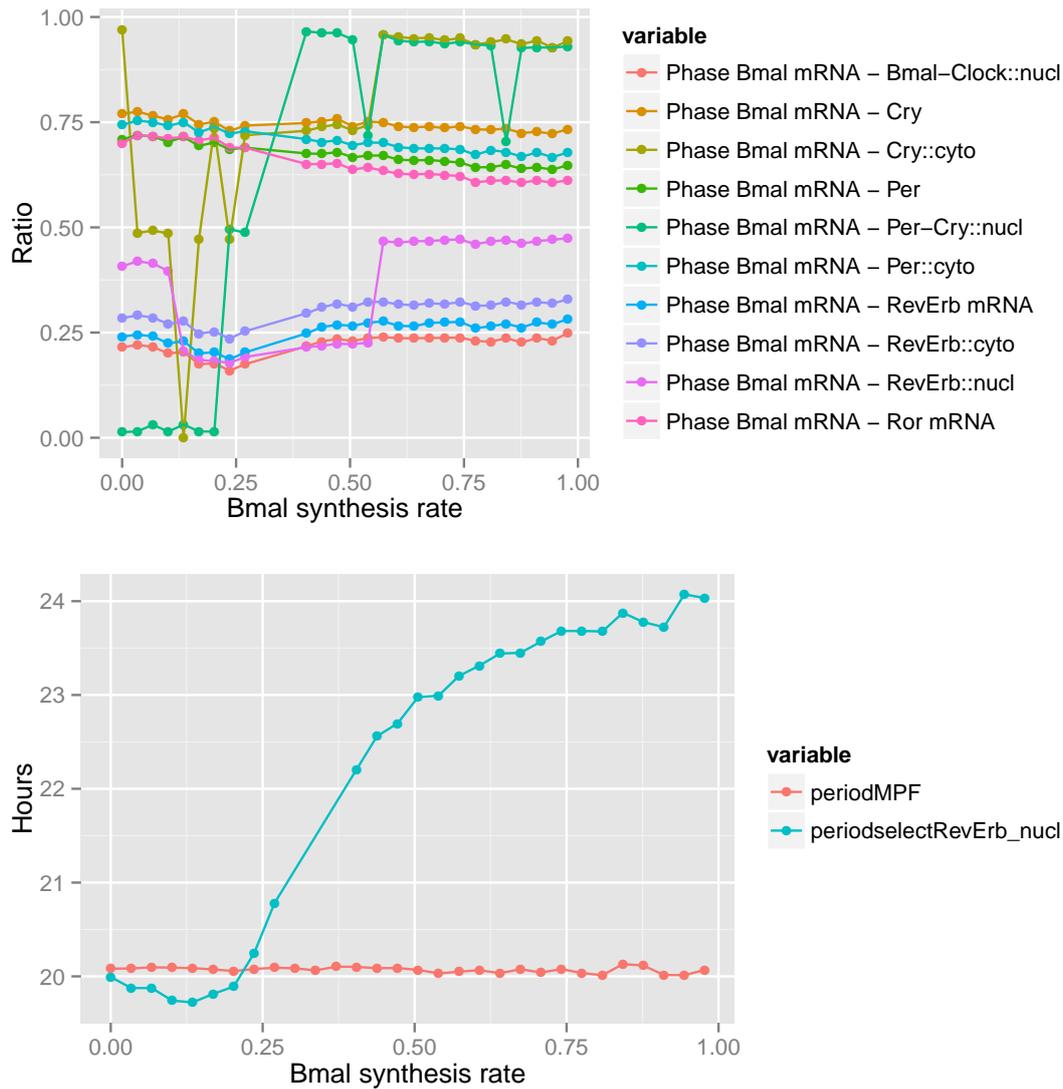


Figure 5.11: Circadian clock phases in the coupled model when the cell cycle has a period of 21h and the synthesis rate of *Bmal1* varies due to the inhibition. The synthesis rate is normalized to exhibit the effect of the applied inhibition (0: full inhibition, 1: no inhibition.)

The simulations reveal that in the entrained condition (when the synthesis rate of *Bmal1* is inhibited with a coefficient smaller than 0.2), the phases between clock components are not deeply impacted by the periodic inhibition on *Bmal1* resulting from the coupling with the cell cycle, compared to their values in the free clock (when the synthesis rate of *Bmal1* is inhibited with a

coefficient close to 1). However, the phase between *Bmal1* and *RevErb- α* mRNAs shows a small advance, that impacts similarly the phases between Bmal1 and RevErb- α in the cytoplasm and in the nucleus. Notably, the other clock genes and proteins targeted by Bmal1 exhibit a phase delay when the synthesis of *Bmal1* is inhibited during mitosis.

Comparison to Experimental Data after Treatment by Dexamethasone

In order to take into account the experiments with Dexamethasone, the model can be extended with an event, lasting for two hours, and inducing *Per* mRNA while inhibiting the other clock genes.

Fig. 5.12 shows that in our models, regardless of the milieu (i.e. of the value of *kdie*), the Dex pulse results in a perturbation of the clock and then returns to the observed entrainment.

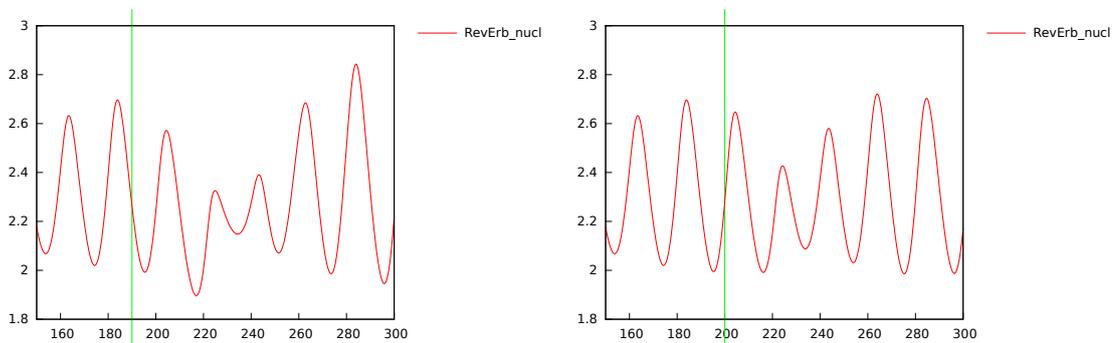


Figure 5.12: Effect of a Dexamethasone pulse on the entrainment resulting from the periodic inhibition of *Bmal1* synthesis by the cell cycle. The pulse alters the clock before returning to the previously observed entrainment regime. In the left panel the pulse is from time 190 to 192 while on the right it is from 200 to 202. The left panel's peak-to-peak distance is in the $[18.2, 21.2]$ interval, while the right one remains in the $[19.31, 20.7]$ interval. This might correspond to the two groups observed in [Feillet et al., 2014]. The time to recover normal entrainment varies but is often larger than 72h.

These simulations point us to the possibility that the noisy data reported in Table 5.1 after the Dex pulse might simply be due to the various cellular states in which the pulse happened and to the time necessary for the cells to recover their clock entrainment, rather than to two different oscillatory attractors of the system.

A pulse at time 200h disrupted only slightly the circadian clock, which remained mostly in mode-locking 1:1 with the cell cycle, whereas advancing that same pulse by 10h (corresponding to giving the pulse to a cell in a different state) leads to a bigger disturbance, some peak-to-peak distances close to 21h, others to 18h. Although this is transitory, this might correspond to the type of data observed in the Group 2 of Table 5.1.

5.5 Alternative hypotheses

Selective activation of *revErb- α*

The second solution found by the calibration procedure in section 5.4 is discussed here. The response curves for the period and phase are depicted in Fig. 5.13 and the corresponding values are summarized in Table 5.5.

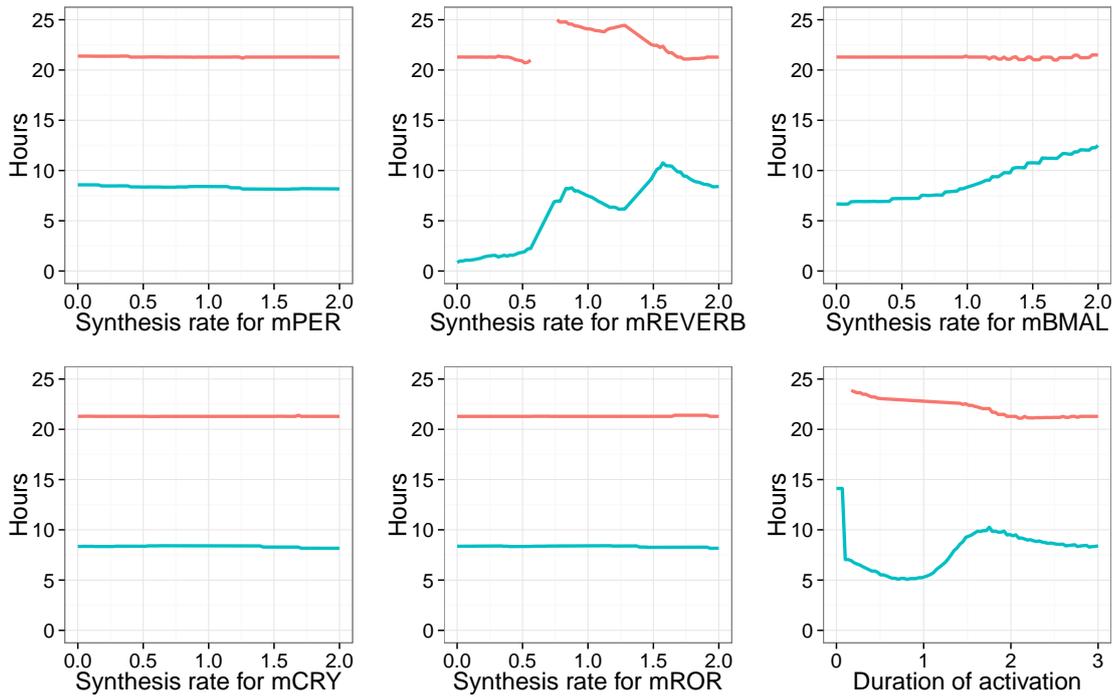


Figure 5.13: Period of the circadian clock (red) and phase between the division and $RevErb-\alpha$ (blue) in the coupled model calibrated with the second set of parameters, and when the cell cycle has a period of 21h.

k_{die}	FBS %	Circadian clock period (h)	Cell division period (h)	Phases (h)
0.077	5?	26.14	26.12	6.1
0.147	10	21.52	21.28	8.5
0.229	15	18.48	18.60	7.2

Table 5.5: Periods and time delays measured in the coupled model with different values of k_{die} for modeling the different culture conditions (the correspondence with 5% FBS is speculative since no experiment was done in this condition). The delays are the time observed by simulation between the peaks of concentration of MPF and $RevErb-\alpha$.

For this solution, similarly, the entrainment in period and phase of the circadian depends only on the effect of the mitosis on $Bmal1$ and $RevErb-\alpha$, and the duration of this effect. Varying the synthesis coefficients during mitosis for Per , Cry or Ror has no significant effect on the entrainment.

However in this case the entrainment in period depends on the effect on $RevErb-\alpha$, which has to be activated with a coefficient close to 2 in order for the circadian clock to be entrained at 21h with a correct delay after division. Like in the previous case, an inhibition would also preserve the entrainment in period, but the phase would become inconsistent with the data. One can notice that an activation on $Bmal1$ would also increase the time delay of $RevErb-\alpha$ peaks after division.

This solution points toward the activation of $RevErb-\alpha$ as an alternative hypothesis to explain the experimentally observed entrainment of the circadian clock in period and phase. Since $RevErb-\alpha$ is a repressor of $Bmal1$ and $Bmal1$ an activator of $RevErb-\alpha$, these two hypothesis correspond to two alternative mechanisms for a similar effect, and further experimental data

would be needed to discriminate between them.

This hypothesis is modeled with a single activation of *RevErb- α* with a coefficient of 2 during mitosis, and no effect of mitosis on the other clock genes. This activation of *RevErb- α* might be caused by the transcription factor c-Myc, which displays bursts of transcriptional activity during G1 phase (i.e., just after mitosis) and the S to G2/M transition of the cell cycle [Seth et al., 1993]. The c-Myc protein regulates its target genes through the same E-box DNA response element as the Clock/Bmal1 heterodimer. It is therefore conceivable that during the G2/M phase of the cell cycle *RevErb- α* is positively regulated by c-Myc leading to the transcriptional repression of *Bmal1* as suggested by a recent study [Altman et al., 2015]. In this scenario the E-box regulated *Per* and *Cry* genes are expected to be also upregulated by the higher transcriptional activity of c-Myc. This is compatible with our simulations showing that the phase and period are resilient to variation of the coefficient synthesis for *Per* and *Cry*.

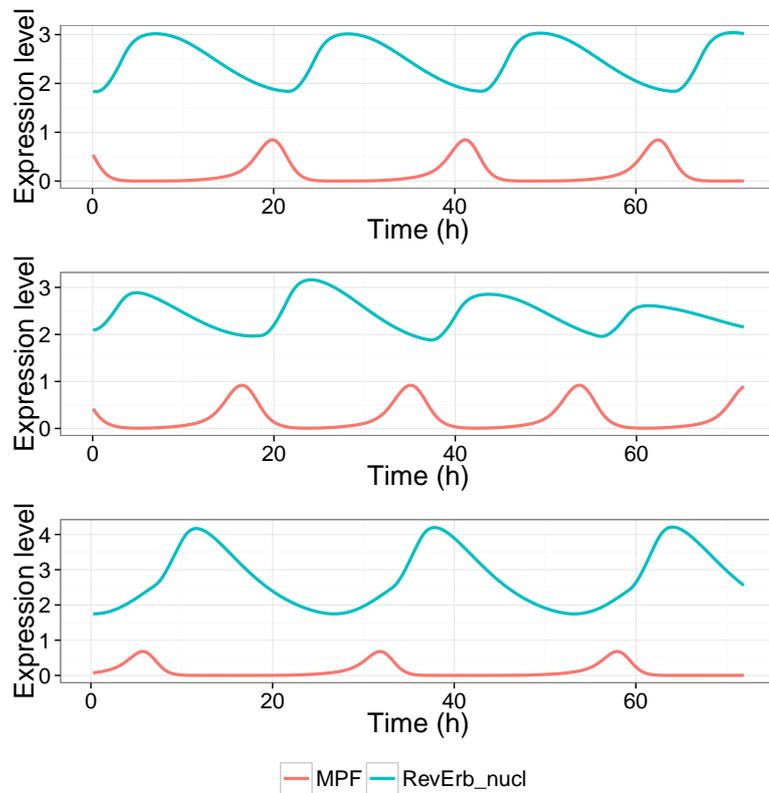


Figure 5.14: Simulation of the model with the activation of the transcription of *RevErb- α* triggered by mitosis, during 72h. Top: the cell cycle has a period of 21.3h. Middle: the cell cycle has a period of 20.1h. Bottom: the cell cycle has a period of 26h.

Like the inhibition of *Bmal1* transcription, the activation of *RevErb- α* during 3h, triggered at mitosis, is able to entrain the circadian clock in a wide range of period (18-28h). Resulting traces are displayed in Fig. 5.14 for different cell cycle length conditions. However, the entrainment in phase, i.e. the duration between mitosis and the following *RevErb- α* peak, differs between the two couplings, as seen in Fig. 5.15. In both cases, when the cell cycle has a period smaller than 24h, the circadian clock marker *RevErb- α* peaks 6h to 8h after mitosis. When the cell cycle period is greater than 24h, a notable difference can be seen for the predicted phase: with an activation of *RevErb- α* , *RevErb- α* still peaks just after the mitosis. But with an inhibition of *Bmal1*, *RevErb- α* peaks 18 to 24 hours after the mitosis. Although no experimental observations exist in a slowed down cell cycle condition, it is noteworthy that [Nagoshi et al., 2004] and [Bieler et al., 2014]

report some cells dividing not long after the circadian peak where the circadian clock was found to be slowed down. This is the reason why we favor the hypothesis of *Bmal1* inhibition to that of *RevErb- α* activation.

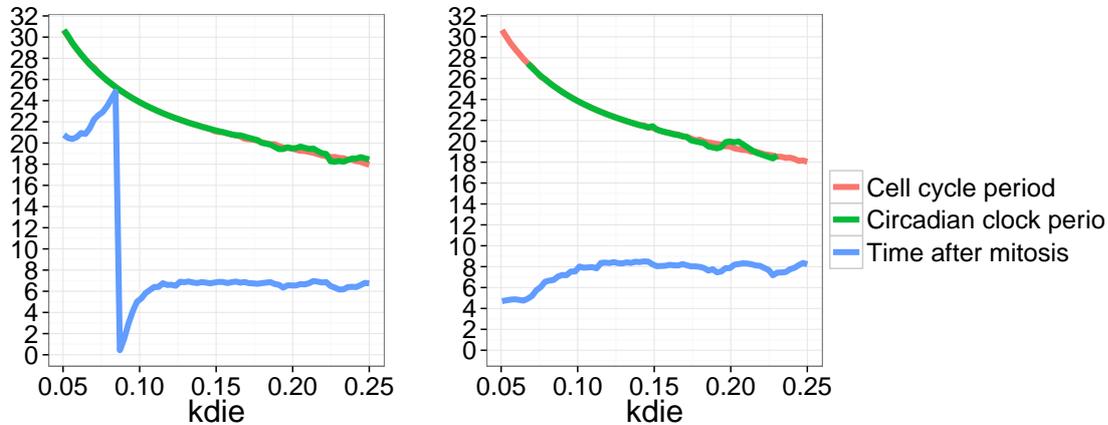


Figure 5.15: Entrainment in period and phase of the circadian clock when the period of the cell cycle varies with the parameter k_{die} , with the inhibition of *Bmal1* (left) or the activation of *RevErb- α* (right) triggered by mitosis. In the right panel, the circadian clock period is missing for low k_{die} values because the oscillations are irregular. The blue curves depict the time delay between peaks of MPF and *RevErb- α* in the simulations.

Uniform inhibition of transcription during mitosis

In eukaryotes, it has been shown that gene transcription can be significantly inhibited during mitosis [Weisenberger and Scheer, 1995]. The impact of a global transcription inhibition of clock genes during mitosis on the circadian oscillator has been studied by modelling in [Kang et al., 2008]. In this study, the authors found that a periodic inhibition of transcription during one hour was able to entrain a model of the mammalian circadian clock, but only when the inhibition period was close to the intrinsic circadian model period, or close to its half or double value. In these cases, a phase locking between the circadian clock and the periodic inhibition was observed, albeit with one or two preferential circadian phases for the inhibition and values that varied greatly with the inhibition period. The discrepancies with the recent data could come from the short inhibition duration considered or from the arbitrarily parameterized model used for the circadian clock, taken from [Leloup and Goldbeter, 2003].

We also investigated the uniform inhibition of all clock genes with Relgio's model of the circadian clock, and found that it was sufficient to reproduce the entrainment of the circadian clock by the cell cycle in period, but not in phase.

The variation of the period of *RevErb- α* can be seen in Figure 5.16 when two parameters vary: the x-axis parameter accelerates the cell cycle and the parameter on the y-axis is the coefficient for the transcription inhibition during mitosis, corresponding to a weaker control of the circadian clock by the cell cycle as the parameter value increases. Two domains can be distinguished in this parameter space: in the domain on the top left (above the black line), the inhibition of the transcription is weak and the cell cycle is slow. In this condition, the clock is not entrained but rather keeps its period constant and close to 24h. On the contrary, in the domain on the bottom right (below the black line), clock genes transcription is strongly inhibited and the cell cycle is faster. In that case the clock is entrained to the same period as the cell cycle.

One can notice that, as with the previous coupling hypothesis, the model explains the acceleration of the circadian clock by the cell cycle and predicts that, provided a strong enough coupling, the circadian clock can also be slowed down by the cell cycle to periods higher than 24h.

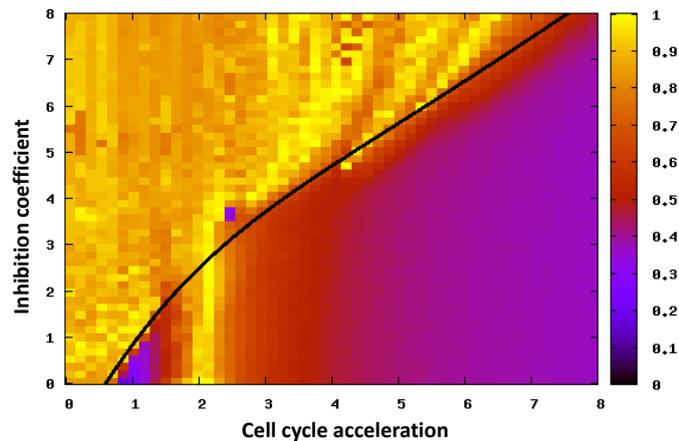


Figure 5.16: Periods of Rev-Erb- α as a function of a parameter that accelerates the cell cycle, and a parameter involved in the coupling by transcription inhibition during mitosis. Landscape computed as the satisfaction degree of the formula `distanceSuccPeaks([RevErb_nucl],[period],transT)`, that defines the period of RevErb- α after a transient time $\text{transT}=80$, and with an objective of 24h for the *period*. Full satisfaction in yellow indicates a period of 24h for RevErb- α , while the other colors indicate the absolute difference to 24h.

On the other hand, the delay between MPF and RevErb- α remained inconsistent with the data, as depicted in Fig. 5.17: the mitosis triggered by MPF occurs just after the peaks of RevErb- α , while the experimental studies consistently report peaks of RevErb- α 7h to 8h after divisions.

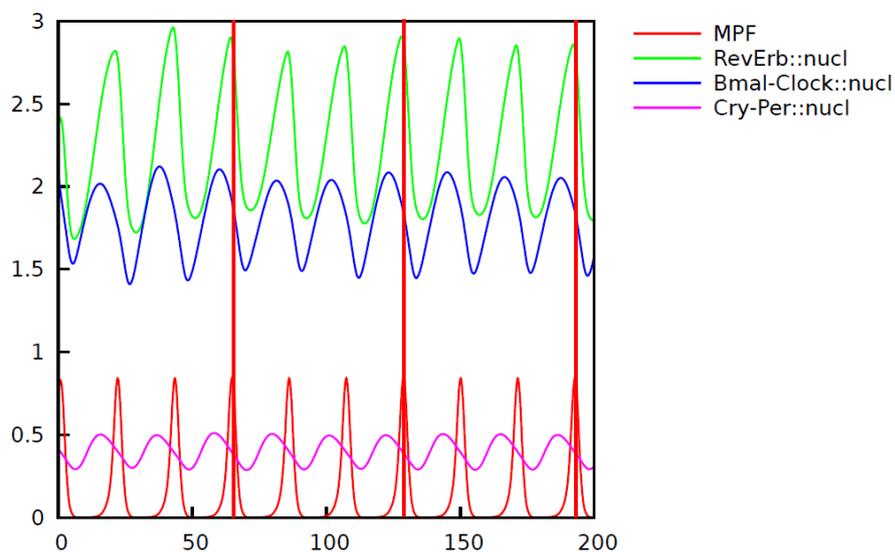


Figure 5.17: Entrainment to a period around 21.3h with $kampf = 3.75$, which corresponds to a milieu enriched with 10% FBS. Red vertical lines mark some peak times of MPF to facilitate the comparison with RevErb- α peaks.

In the circadian clock model of [Relógio et al., 2011], the phases of the different markers of the circadian clock have been precisely fitted to observations made in mouse suprachiasmatic

nucleus neurons, however without data about cell divisions. On the other hand, RevErb- α is the only marker on the circadian clock measured in the experiments, and no comparison is thus possible with the other data [Feillet et al., 2014].

5.6 Bidirectional coupling

We can also integrate in our model the influence classically studied between the cell cycle and the circadian clock: the control of the cell cycle by the circadian clock. Based on experimental observations [Nagoshi et al., 2004] and mathematical modeling [Gérard and Goldbeter, 2012], it has been hypothesized that this control implements a gating of the mitosis during preferential circadian phases.

Although it appears from the experimental data that the preponderant coupling in the fibroblasts is the control of the circadian clock by the cell cycle, investigated in the previous section, we study here the effect of the additional control of the cell cycle by the circadian clock. It results in a bidirectional coupling, represented on the schema in Figure 5.18.

Several links by which the circadian rhythm influences the cell cycle have been studied. For example, the protein RevErb- α can inhibit the transcription of the Cdk inhibitor p21 [Gréchez-Cassiau et al., 2008], or repress the gene *c-myc* that induces the expression of cyclin E [Perez-Roger, 1997]. However the dominant influence seems to be the link established between the cell and circadian cycles during the G2-M transition by the kinase Wee1: the *wee1* gene promoter is activated by the complex Clock/Bmal1 and inhibited by *Cry* [Matsuo et al., 2003].

In order to keep a simple model of the cell cycle and maintain reasonable simulation times, we focus here on the latter mechanism, implemented through the following reaction in Biocham (Listing 5.2), which defines a classic Hill function for the activation of the transcription of Wee1 mRNA, called mWee1, by the protein Bmal1 in the nucleus, called Bmal1_nucl in the model and which stands for the complex Clock/Bmal1.

```
vsw*[Bmal1_nucl]^nmw/(Kaw^nmw+[Bmal1_nucl]^nmw)
for _=[Bmal1_nucl]=>mWee1.
parameter(vsw,0.15).
parameter(Kaw,1).
parameter(nmw,1).
```

Listing 5.2: mWee1 synthesis inhibition by Bmal1_nucl implemented in Biocham

The landscapes in Figure 5.19 display the satisfaction degree scoring how the period of MPF from the cell cycle and RevErb- α from the circadian clock satisfy the objective of 24h. Increase of *kdie* (x -axis), accelerates the cell cycle. The parameter *vsw* (y -axis) characterizes the strength of the control that the circadian clock exerts on the cell cycle through the activation of Wee1.

First let us look at the bottom line of each landscape, corresponding to the particular condition $vsw = 0$: there is no control of the cell cycle by the circadian clock. Hence, the color at the bottom line of the top landscape indicates the free period of the cell cycle, and on the bottom landscape it indicates the period of the circadian clock controlled by the cell cycle.

On this line two domains can be distinguished: a region for $kdie < 0.05$ which is different for MPF and RevErb- α , and a region for $kdie > 0.05$ which is identical for both molecules, hence reflecting the entrainment of the circadian clock period to the same value as the cell cycle, which has already been studied in the previous section. In this condition, this period is equal to the objective value of 24h for $kdie = 0.1$. It is higher than 24h for $kdie \leq 0.1$ and smaller than 24h for $kdie \geq 0.1$. In contrast, for $kdie \leq 0.05$, the 24h-period of the circadian clock is restored. In this region, there is no entrainment of the circadian clock by the cell cycle.

The evolution of these two regions for positive values of *vsw* shows how the additional coupling impacts the entrainment of the circadian clock by the cell cycle observed with the unidirectional coupling, studied in the previous section. The period of entrainment of the circadian clock is changed not only by the value of *kdie*, which modulates the free period of the cell cycle, but also by the value of *vsw*, which is the strength of the coupling from the circadian clock to the cell

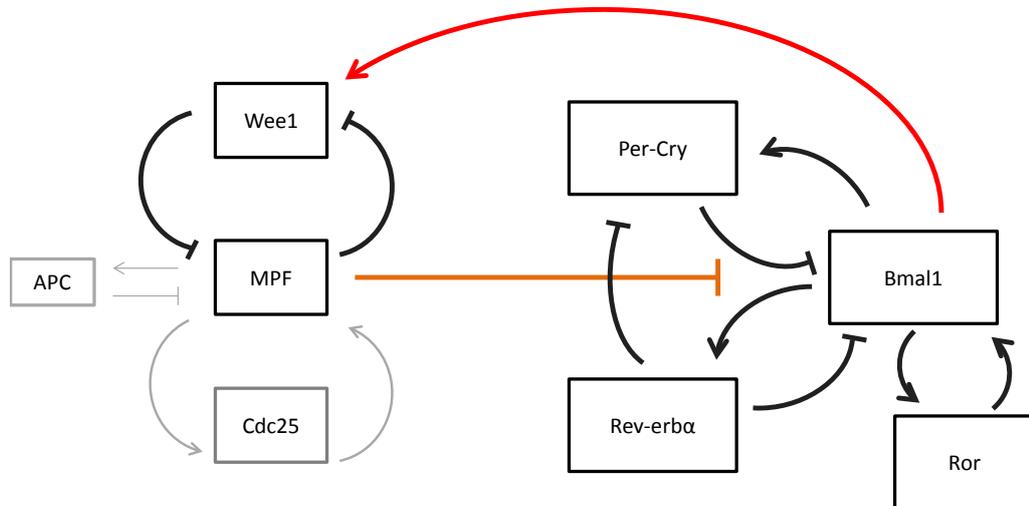


Figure 5.18: Schema of the bidirectional coupling between the cell cycle and the circadian clock. In red, the classic control of the cell cycle by the circadian clock. In orange, the reverse coupling through the inhibition of *bmal1* transcription during mitosis.

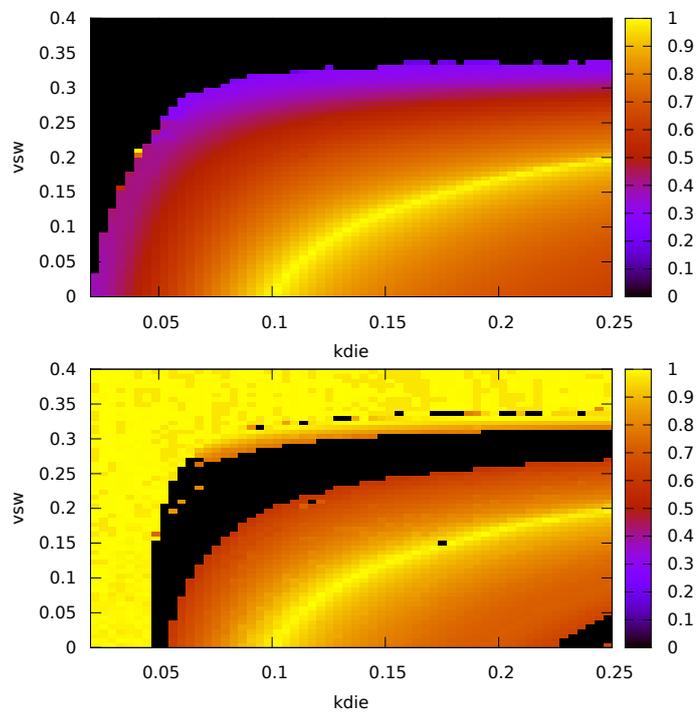


Figure 5.19: Landscape showing the satisfaction degree for the period of MPF (cell cycle, top) and RevErb- α (circadian clock, bottom).

cycle. In the entrainment region the period of the cell cycle changes identically to the circadian clock when the value of vsw changes. Therefore, there is a mutual entrainment of the cell cycle and the circadian clock, and increasing the value of vsw has the effect of slowing down both the cell cycle and the circadian clock. Two additional regions appear in black (null satisfaction degree), corresponding to conditions in which the oscillations of the cell cycle or the circadian clock are disrupted. The first one, close to the top and left borders of each panel, corresponds to either a slow cell cycle or a strong inhibition of *wee1* by *Bmal1_nucl*. In this region the circadian clock keeps its free period and disrupts the cell cycle. Between this region and the domain of mutual entrainment, another region stands out, where the cell cycle exhibits slow oscillations but is unable to fully entrain the circadian clock, resulting in a disruption of clock oscillations.

The domain of non-entrainment is wider for high values of vsw : the conditions of entrainment of the circadian clock are restricted to the conditions in which the free cell cycle is relatively fast.

5.7 Conclusion

By hypothesizing a selective inhibition of *bmal1* or a selective activation of *revErb- α* triggered by mitosis, we have been able to build a mechanistic dynamical model reproducing the somewhat surprising data reported in [Bieler et al., 2014, Feillet et al., 2014] about the acceleration of the circadian clock observed in dividing fibroblasts with high FBS concentrations. These observations suggest that the primary coupling between the cell division cycle and the circadian clock results from an influence of the cell cycle on the circadian clock, at least in fibroblasts.

While considering a uniform inhibition of the transcription during mitosis [Weisenberger and Scheer, 1995] was shown to be sufficient to fit the period data in [Traynard et al., 2015a], the phase data reported in [Feillet et al., 2014] seemed to be impossible to reproduce under that uniform inhibition hypothesis. The use of temporal logic patterns and our dedicated solvers, for searching transcription inhibition parameters satisfying the phase observations formalized in quantitative temporal logic, led us to the hypothesis that either the transcription of *bmal1* has to be strongly inhibited, while no inhibition should affect the transcription of *revErb- α* , or that *revErb- α* has to be strongly activated, but with different predictions concerning slow cell cycles under low levels of FBS.

Our model also postulates a different interpretation of some of the results presented in [Feillet et al., 2014] when cells are treated by a 2h pulse of Dexamethasone. Instead of different autonomous cycling regimes, the model predicts temporary perturbations leading to shorter or longer peak-to-peak distances, but returning to the previous entrainment regime after some time, longer than the horizon used in the experiments.

Furthermore, in our coupled model, the phases between some of the clock gene products are shifted when entrained by a fast cell cycle. We are able to quantify these phase shifts and we show that they concern mainly *RevErb- α* , whose mRNA peaks are advanced by the periodic activation during mitosis. Other clock mRNAs and proteins are slightly delayed compared to their activator *Bmal1*. Hence, a prediction of the model is that, in quickly dividing cells, these protein peaks are shifted with respect to quiescent cells where such a phenomenon should not be observed.

On the other hand, our model leaves completely open the molecular mechanisms responsible for the hypothesized selective inhibition of *bmal1* during mitosis, or activation of *revErb- α* , for which several mechanisms can be imagined [Loyer et al., 2005].

A possibility is a strong effect of the changes in the metabolism of the cell during and after mitosis on the circadian clock. Interplays between metabolism and the circadian clock, as well as the cell cycle, as been put in evidence [Eckel-Mahan and Sassone-Corsi, 2009]. However, precise relationships still remain to be identified. Further knowledge in this field will be necessary to confirm or invalidate our hypothesis. It is probable that novel mechanisms that link the circadian clock and the cell cycle through metabolism will be revealed.

Finally, we predict a mutual entrainment in period when considering a bidirectional coupling between the cell cycle and the circadian clock, with a possible slowing down of both the cell cycle

and the circadian clock, and a possible disruption of the cell cycle under a strong control by the circadian clock.

6. Conclusion, prospects

6.1 Summary

In this thesis, we have explored the use of temporal logic for model building in Systems Biology. The logical paradigm of biological systems abstracted as state-transition sequences and biological phenomenon formalized as logical constraints allows us to take advantage of analysis techniques from computer science and to infer mechanisms for properties emerging from complex regulatory networks. In this respect, oscillatory systems are particularly interesting study cases.

First, we have investigated the ability of Computation Tree Logic (CTL) to verify dynamical properties in asynchronous state transition graphs derived from logical models. Logical modeling provides a qualitative and potentially non-deterministic description of a biological system. This characteristic is useful to account for a variety of dynamical properties, observed in different conditions, together in a generic model. We developed an approach based on iterative property verification to assist the building and updating of a logical model of the mammalian cell cycle. In that case, model checking is efficiently used to analyze complex attractors and infer qualitative properties of the system. We successfully completed the model with additional nodes and interactions. We showed that, in addition to conserving previous perturbations properties, the updated model accounts for new perturbations and dynamical properties. Notably, the model can explain the sequential activity of cyclins in the G1/S transition.

Next, we have considered quantitative deterministic models. For such models, oscillatory properties such as pseudo-periods and pseudo-phases are formalized as quantitative constraints with First-Order Linear Time Logic (FO-LTL). A continuous model provides a precise description of the mechanisms governing a complex regulatory network, and a quantitative prediction of its dynamics. However, the classical difficulties associated with this approach are brought by numerous and often poorly characterized kinetic parameters. We address this challenge by implementing two complementary strategies to obtain efficient solving of dynamical constraints: the design of useful temporal logic formula patterns associated with dedicated constraint solvers, and some trace simplification rules to safely reduce the size of the traces to analyze.

We have applied these methods to a mammalian modular model coupling the circadian clock to the cell cycle. We showed how temporal logic constraints allow to calibrate high-dimensional models to quantitative single-cell data. We investigated several coupling hypotheses that reproduced the acceleration of the circadian clock observed in fast dividing cells, and discriminated them regarding their ability to reproduce the phase relationships between the cell cycle and the circadian clock. We selected an inhibition of the clock gene *bmal1* triggered by mitosis as the best hypotheses among those tested. We then used the model to predict the slowing down of

the circadian clock, and a modification of the phases between clock components in dividing cells. Adding the classic control of the cell cycle by the circadian clock, we further predict a reciprocal entrainment of the cell cycle and the circadian clock in some conditions. These new behaviors should be tested experimentally.

6.2 Contributions

Methodological contributions

The methodological contributions of this project include the formalization of all dynamical properties presented in this dissertation in temporal logic and the definition of trace simplifications keeping only local extrema or main local extrema of a trace.

In particular, the formalization of dynamical properties specific to oscillating traces with FO-LTL(\mathbb{R}_{lin}) and the definition of dedicated solvers have been published in a book chapter [Fages and Traynard, 2014]. Moreover, the definition of trace simplifications, with theorems stating on which conditions validity domains are preserved after trace simplification (discussed in Section 4.3) have been exposed in [Traynard et al., 2014].

A paper discussing the application of model checking to the analysis and updating of a logical model of the mammalian cell cycle, described in Chapter 2, has been submitted for publication [Traynard et al., 2015b].

Biological contributions

The investigation of the hypotheses of the control of the circadian clock by the cell cycle through the inhibition of transcription during mitosis, discussed in Chapter 5, has been published [Traynard et al., 2015a]. An extended version of this paper exploring the hypotheses of selective regulation of clock genes triggered by mitosis has been submitted.

Implementations

Contributions to the development of Biocham include the following points. All these additions have been documented in Biocham's user manual (cf. <http://lifeware.inria.fr/biocham>).

- We have completed the evaluation of FO-LTL formulae by Biocham with the generic algorithm, with the implementation of the missing operators \forall and \exists , as well as the completion operation.
- We have implemented the relations combining temporal logic macros and dedicated solvers defined in Section 4.2, with an optional argument that allows to neglect a user-specified time duration at the beginning of the trace. Apart from ensuring fast evaluation of specifications, this contribution also provides a user-friendly tool to exploit constraints applied on a model. Indeed, given the high complexity of temporal logic syntax, a certain expertise is required to use temporal logic formalism. In contrast, a rich list of built-in constraints can now be easily called. Although this contribution is too recent to estimate its level of acceptance and understanding by Biocham users, we believe that it should encourage users with little or no formal background to use constraints-based tools for biological modeling and analysis with Biocham.
- We have also implemented the trace simplifications, described in Sections 4.3 and 4.4, keeping only local extrema or main local extrema of a trace.
- Furthermore, we have completed or adapted several functions already implemented in Biocham: the parallel parameter search has been adapted to operate with a work queue

to distribute jobs dynamically, and to include a timeout option. We have parallelized the existing tool to compute landscapes for satisfaction degrees, and implemented a parallel tool to compute validity domains for response curves.

"Many tools providing implementations of model checking are experimental and academic in nature. This implies that users require certain expertise in under-lying techniques and formalisms in order to use model checking. The development of more reliable and user-friendly tools, as well as approaches that facilitate the creation of models, could "

Some contributions have also been implemented to easily manipulate logical models with GINsim through several scripts in Python. One of them automatically defines a list of perturbations in a GINsim model, simulates all perturbations and extract all attractors. Another script automatically verifies a list of dynamical properties, called with easy-to-use macros, on a logical model, and print the result in a HTML report. However these contributions are not part of GINsim and have not yet been documented or formalized.

6.3 Prospects

Logical model inference

CTL constraint solving is a formal method that allows us to verify exhaustively the satisfiability of a dynamical property on a logical model. We have shown in Chapter 2 a novel approach for applying this method to the analysis of cyclic attractors. We used this approach to assist the updating of a logical model by identifying the consequences caused on the dynamical properties of the model by each manual modification brought to the regulatory network or logical rules.

Recently, more powerful approaches have been explored to provide automatic model building solutions and free modelers from manual modifications. Formal methods can be used to produce a model from specifications, including perturbation experiments, based on satisfiability query and solved with SAT and SMT solvers.

Such problems remain costly to solve and rely on sufficient topological information and constraints. Statistical analysis can reduce this cost. This strategy is for example employed by [Thobe et al., 2014], who explore a statistical analysis of cross-talk model pools composed of boolean networks.

The integration of novel experimental data with single-cell resolution and temporal scales is particularly useful to solve synthesis problems. For instance, [Fisher et al., 2015] introduce an algorithm to reconstruct a boolean network from its state space, and apply this strategy to infer a boolean model from single-cell gene expression profiles, without a-priori information about the topology of the network.

Automatic property inference is also a problem of interest, addressed by the model checker ANTELOPE [Arellano et al., 2011], which extends CTL with hybrid-logic operators to support incompletely specified behavior.

Quantitative extension of qualitative modeling

Approaches to generate quantitative information about the dynamics in discrete state transition graphs could be further explored. Continuous-time Markov chains are a useful extension of logical models to analyze quantitatively nondeterministic dynamics. This is the approach used by MaBoSS [Stoll et al., 2012], which provides stochastic trajectories and state probability distributions. This formalism is used in Section 2.5 to generate mean trajectories in the asynchronous attractor.

MaBoSS could be further used to study the effect of coupling two oscillating systems such as the cell cycle and the circadian clock with logical modeling, thereby potentially producing results complementary to the continuous approach. Quantitative information such as probabilities of different types of behaviors, or the probability of the time spent in the cycle would be useful,

in particular to assess how they are affected by perturbations, similarly to a previous successful analysis [Calzone et al., 2010].

Moreover, model-checking techniques exist to analyze such models. For example, [Fisher and Piterman, 2014] extend CTL by replacing path quantification by probabilistic quantification. The resulting continuous stochastic logic (CSL) can be used to express properties of continuous-time Markov chains.

Formal verification for quantitative modeling

Our approach to handle continuous models by solving temporal constraints formalized as FO-LTL(\mathbb{R}_{lin}) formulae is able to infer parameter values to satisfy a set of properties with optimization procedures, but it relies heavily on numerical simulations. Although the access to high performance computing facilities allow extensive explorations of the parameter space, this exploration is costly and far from exhaustive, and the risk of neglecting a relevant optimum remains high. Moreover, failings to find a parameterization that satisfy some constraints suggest flaws in the structure of the model but are not sufficient to formally reject it.

In this respect, symbolic methods for hybrid discrete-continuous models are promising research axes to provide formal verification for quantitative modeling. SAT-modulo-theory (SMT) approaches take advantage of the powerful classical Boolean satisfiability (SAT) solving, which is able to verify Boolean properties exhaustively, and extends its expressiveness by generalizing the constraints to complex predicates. For instance, [Eggers et al., 2008] propose an SMT approach to interval-based numeric approximation of ODEs. This constraint solving of initial-value problems is able to narrow candidate sets of reachable states from a set of initial values, and vice versa. This SAT modulo ODEs strategy is extended to address more general specifications by [Gao et al., 2013], who define a δ -complete decision procedures for SMT constraints over the reals: a formula is δ -satisfiable if, under some δ -perturbations, a syntactic variant of the original formula is satisfiable.

Cell cycle and circadian clock modeling

Considering the coupling between the cell cycle and the circadian clock in mouse fibroblasts, our modeling approach leads us to propose a novel biological hypothesis: the inhibition of *bmal1* triggered by mitosis, which awaits experimental validation. We also propose an alternative hypothesis with the activation of *rev-erba*. Although it is very well possible that the inhibiting influence is not caused by a direct interaction, mathematical modeling at least strongly suggests a mechanism of selective inhibition.

Further analyses of this coupling could study the impact on various characteristics of interest, such as robustness measures for circadian oscillations and cell cycle progression, or sensitivity to external stimuli. Validation of such findings would require specific experimental data.

Ultimately, assessing the role of the coupling between the cell cycle and the circadian clock in cancer cells is particularly interesting for future applications. To this end, studies should investigate how this coupling is affected in perturbed conditions. Although experimental studies of circadian clock genes mutations have been performed in mice [Li et al., 2013], time and spatial resolution of such data is still sparse.

The increasing study of model organisms, exhibiting conserved coupled cell cycle and circadian clock machineries relatively to mammals, and providing easier genetic mutants, will contribute to new perturbation data. In this respect, *Neurospora crassa* has recently been shown to be a good model organism to investigate molecular mechanisms and emerging behavior of the coupled networks driving the cell cycle and circadian rhythms [Zámborszky et al., 2014].

Bibliography

- [Abou-Jaoudé et al., 2015] Abou-Jaoudé, W., Monteiro, P. T., Naldi, A., Grandclaudon, M., Soumelis, V., Chaouiya, C., and Thieffry, D. (2015). Model checking to assess T-helper cell plasticity. *Frontiers in Bioengineering and Biotechnology*, 2(Suppl. 1):86.
- [Albert and Thakar, 2014] Albert, R. and Thakar, J. (2014). Boolean modeling: a logic-based dynamic approach for understanding signaling and regulatory networks and for making useful predictions. *Wiley Interdiscip Rev Syst Biol Med*, 6(5):353–369.
- [Altman et al., 2015] Altman, B. J., Hsieh, A. L., Sengupta, A., Krishnanaiah, S. Y., Stine, Z. E., Walton, Z. E., Gouw, A. M., Venkataraman, A., Li, B., Goraksha-Hicks, P., et al. (2015). MYC disrupts the circadian clock and metabolism in cancer cells. *Cell metabolism*, 22(16):1009–1019.
- [Arellano et al., 2011] Arellano, G., Argil, J., Azpeitia, E., Benítez, M., Carrillo, M., Góngora, P., Rosenblueth, D. A., and Álvarez-Buylla, E. R. (2011). "Antelope": a hybrid-logic model checker for branching-time Boolean GRN analysis. *BMC Bioinformatics*, 12(1):490.
- [Assoian and Yung, 2008] Assoian, R. K. and Yung, Y. (2008). A reciprocal relationship between Rb and Skp2: implications for restriction point control, signal transduction to the cell cycle and cancer. *Cell Cycle*, 7(1):24–27.
- [Bagnara et al., 2008] Bagnara, R., Hill, P. M., and Zaffanella, E. (2008). The Parma Polyhedra Library: Toward a complete set of numerical abstractions for the analysis and verification of hardware and software systems. *Science of Computer Programming*, 72(1–2):3–21.
- [Ballesta et al., 2011] Ballesta, A., Dulong, S., Abbara, C., Cohen, B., Okyar, A., Clairambault, J., and Levi, F. (2011). A combined experimental and mathematical approach for molecular-based optimization of irinotecan circadian delivery. *PLOS Computational Biology*, 7(9).
- [Banks et al., 2015] Banks, C. J., Seaton, D. D., and Stark, I. (2015). *Computational Methods in Systems Biology: 13th International Conference, CMSB 2015, Nantes, France, September 16-18, 2015, Proceedings*, chapter Analysis of a Post-translational Oscillator Using Process Algebra and Spatio-Temporal Logic, pages 222–238. Springer International Publishing, Cham.
- [Barnes et al., 2003] Barnes, J. W., Tischkau, S. A., Barnes, J. A., Mitchell, J. W., Burgoon, P. W., Hickok, J. R., and Gillette, M. U. (2003). Requirement of mammalian timeless for circadian rhythmicity. *Science*, 302(5644):439–442.
- [Batt et al., 2010] Batt, G., Page, M., Cantone, I., Goessler, G., Monteiro, P., and de Jong, H. (2010). Efficient parameter search for qualitative models of regulatory networks using symbolic model checking. *Bioinformatics*, 26(18):i603–i610.
- [Batt et al., 2005] Batt, G., Ropers, D., de Jong, H., Geiselman, J., Mateescu, R., Page, M., and Schneider, D. (2005). Validation of qualitative models of genetic regulatory networks by model checking : Analysis of the nutritional stress response in *Escherichia coli*. *Bioinformatics*, 21(Suppl.1):i19–i28.

- [Bérengruier et al., 2013] Bérengruier, D., Chaouiya, C., Monteiro, P. T., Naldi, A., Remy, E., Thieffry, D., and Tichit, L. (2013). Dynamical modeling and analysis of large cellular regulatory networks. *Chaos*, 23(2):025114.
- [Bernot et al., 2004] Bernot, G., Comet, J.-P., Richard, A., and Guespin, J. (2004). A fruitful application of formal methods to biological regulatory networks: Extending Thomas' asynchronous logical approach with temporal logic. *Journal of Theoretical Biology*, 229(3):339–347.
- [Bieler et al., 2014] Bieler, J., Cannavo, R., Gustafson, K., Gobet, C., Gatfield, D., and Naef, F. (2014). Robust synchronization of coupled circadian and cell cycle oscillators in single mammalian cells. *Molecular systems biology*, 10(7):739.
- [Billy et al., 2014] Billy, F., Clairambault, J., Fercoq, O., Gaubert, S., Lepoutre, T., Ouillon, T., and Saito, S. (2014). Synchronisation and control of proliferation in cycling cell population models with age structure. *Mathematics and Computers in Simulation*, 96:66–94.
- [Binné et al., 2007] Binné, U. K., Classon, M. K., Dick, F. A., Wei, W., Rape, M., Kaelin, W. G., Naar, A. M., and Dyson, N. J. (2007). Retinoblastoma protein and anaphase-promoting complex physically interact and functionally cooperate during cell-cycle exit. *Nat. Cell Biol.*, 9(2):225–232.
- [Bollinger and Schibler, 2014] Bollinger, T. and Schibler, U. (2014). Circadian rhythms - from genes to physiology and disease. *Swiss Med Wkly*, 144:w13984.
- [Bradley, 2011] Bradley, A. R. (2011). Sat-based model checking without unrolling. In *Proceedings of the 12th International Conference on Verification, Model Checking, and Abstract Interpretation*, Lecture Notes in Computer Science. Springer-Verlag.
- [Brim et al., 2014] Brim, L., Dluhoš, P., Šafránek, D., and Vejpustek, T. (2014). Stl*: Extending signal temporal logic with signal-value freezing operator. *Information and Computation*, 236:52 – 67. Special Issue on Hybrid Systems and Biology.
- [Calzone et al., 2006] Calzone, L., Fages, F., and Soliman, S. (2006). BIOCHAM: An environment for modeling biological systems and formalizing experimental knowledge. *Bioinformatics*, 22(14):1805–1807.
- [Calzone and Soliman, 2006] Calzone, L. and Soliman, S. (2006). Coupling the cell cycle and the circadian cycle. Research Report 5835, INRIA.
- [Calzone et al., 2010] Calzone, L., Tournier, L., Fourquet, S., Thieffry, D., Zhivotovsky, B., Barillot, E., and Zinovyev, A. (2010). Mathematical modelling of cell-fate decision in response to death receptor engagement. *PLoS computational biology*, 6(3):e1000702.
- [Cavada et al., 2002] Cavada, R., Cimatti, A., Olivetti, E., Pistore, M., and Roveri, M. (1998–2002). *NuSMV 2.1 User Manual*. CMU and ITC-irst, IRST - Via Sommarive 18, 38055 Povo (Trento) - Italy.
- [Chabrier and Fages, 2003] Chabrier, N. and Fages, F. (2003). Symbolic model checking of biochemical networks. In Priami, C., editor, *CMSB'03: Proceedings of the first workshop on Computational Methods in Systems Biology*, volume 2602 of *Lecture Notes in Computer Science*, pages 149–162, Rovereto, Italy. Springer-Verlag.
- [Chabrier-Rivier et al., 2004] Chabrier-Rivier, N., Chiaverini, M., Danos, V., Fages, F., and Schächter, V. (2004). Modeling and querying biochemical interaction networks. *Theoretical Computer Science*, 325(1):25–44.
- [Chaouiya, 2007] Chaouiya, C. (2007). Petri net modelling of biological networks. *Briefings in Bioinformatics*.

- [Chaouiya et al., 2012] Chaouiya, C., Naldi, A., and Thieffry, D. (2012). Logical modelling of gene regulatory networks with ginsim. *Methods Mol Biol*, 804:463–79.
- [Chaouiya and Remy, 2013] Chaouiya, C. and Remy, E. (2013). Logical modelling of regulatory networks, methods and applications. *Bull. Math. Biol.*, 75(6):891–895.
- [Chaouiya et al., 2003] Chaouiya, C., Remy, E., Mossé, B., and Thieffry, D. (2003). Qualitative analysis of regulatory graphs: a computational tool based on a discrete formal framework. In *Positive Systems*, pages 119–126. Springer.
- [Chiang et al., 2015] Chiang, H.-J., Fages, F., Jiang, J.-H., and Soliman, S. (2015). Hybrid simulations of heterogeneous biochemical models in SBML. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 25(2):14:1–14:22.
- [Cimatti et al., 2002] Cimatti, A., Clarke, E., Enrico Giunchiglia, F. G., Pistore, M., Roveri, M., Sebastiani, R., and Tacchella, A. (2002). Nusmv 2: An opensource tool for symbolic model checking. In *Proceedings of the International Conference on Computer-Aided Verification, CAV'02*, Copenhagen, Denmark.
- [Clarke et al., 2001] Clarke, E., Biere, A., Raimi, R., and Zhu, Y. (2001). Bounded Model Checking using SAT Solving. *Journal of Formal Methods in System Design*, 19(1):7–34.
- [Clarke et al., 1999] Clarke, E. M., Grumberg, O., and Peled, D. A. (1999). *Model Checking*. MIT Press.
- [Comet et al., 2012] Comet, J.-P., Bernot, G., Das, A., Diener, F., Massot, C., and Cessieux, A. (2012). Simplified Models for the Mammalian Circadian Clock. *Procedia Computer Science*, 11:127–138. Proceedings of the 3rd International Conference on Computational Systems-Biology and Bioinformatics.
- [Cousot and Cousot, 1977] Cousot, P. and Cousot, R. (1977). Abstract interpretation: A unified lattice model for static analysis of programs by construction or approximation of fixpoints. In *POPL'77: Proceedings of the 6th ACM Symposium on Principles of Programming Languages*, pages 238–252, New York. ACM Press. Los Angeles.
- [Csikász-Nagy et al., 2006] Csikász-Nagy, A., Battogtokh, D., Chen, K. C., Novák, B., and Tyson, J. J. (2006). Analysis of a generic model of eukaryotic cell-cycle regulation. *Biophysical journal*, 90(12):4361–79.
- [Dai et al., 2013] Dai, L., Liu, Y., Liu, J., Wen, X., Xu, Z., Wang, Z., Sun, H., Tang, S., Maguire, A. R., Quan, J., Zhang, H., and Ye, T. (2013). A novel CyclinE/CyclinA-CDK Inhibitor targets p27Kip1 degradation, cell cycle progression and cell survival: Implications in cancer therapy. *Cancer Letters*, 333(1):103–112.
- [Davidich and Bornholdt, 2008] Davidich, M. I. and Bornholdt, S. (2008). Boolean network model predicts cell cycle sequence of fission yeast. *PloS one*, 3(2):e1672.
- [de Jong et al., 2004] de Jong, H., Gouzé, J.-L., Hernandez, C., Page, M., Sari, T., and Geiselmann, J. (2004). Qualitative simulation of genetic regulatory networks using piecewise-linear models. *Bulletin of Mathematical Biology*, 66(2):301–340.
- [De Maria et al., 2011] De Maria, E., Fages, F., Rizk, A., and Soliman, S. (2011). Design, optimization, and predictions of a coupled model of the cell cycle, circadian clock, dna repair system, irinotecan metabolism and exposure control under temporal logic constraints. *Theoretical Computer Science*, 412(21):2108–2127.
- [Dick and Rubin, 2013] Dick, F. a. and Rubin, S. M. (2013). Molecular mechanisms underlying RB protein function. *Nature reviews. Molecular cell biology*, 14(5):297–306.

- [Donzé et al., 2013] Donzé, A., Ferrère, T., and Maler, O. (2013). Efficient robust monitoring for STL. In *25th International Conference on Computer Aided Verification, CAV'13*, Berkeley, USA.
- [Donzé and Maler, 2010] Donzé, A. and Maler, O. (2010). Robust satisfaction of temporal logic over real-valued signals. In *FORMATS 2010*, volume 6246 of *Lecture Notes in Computer Science*, pages 92–106. Springer-Verlag.
- [Dunlap et al., 1999] Dunlap, J. C., Loros, J. J., Liu, Y., and Crosthwaite, S. K. (1999). Eukaryotic circadian systems: cycles in common. *Genes Cells*, 4(1):1–10.
- [Eckel-Mahan and Sassone-Corsi, 2009] Eckel-Mahan, K. and Sassone-Corsi, P. (2009). Metabolism control by the circadian clock and vice versa. *Nature structural & molecular biology*, 16(5):462–7.
- [Eggers et al., 2008] Eggers, A., Fränzle, M., and Herde, C. (2008). *Automated Technology for Verification and Analysis: 6th International Symposium, ATVA 2008, Seoul, Korea, October 20-23, 2008. Proceedings*, chapter SAT Modulo ODE: A Direct SAT Approach to Hybrid Systems, pages 171–185. Springer Berlin Heidelberg, Berlin, Heidelberg.
- [El Cheikh et al., 2014] El Cheikh, R., Bernard, S., and El Khatib, N. (2014). Modeling circadian clock-cell cycle interaction effects on cell population growth rates. *J. Theor. Biol.*, 363:318–331.
- [Fages, 2005] Fages, F. (2005). Temporal logic constraints in the biochemical abstract machine BIOCHAM (invited talk). In Springer-Verlag, editor, *Proceedings of Logic Based Program Synthesis and Transformation, LOPSTR'05*, number 3901 in *Lecture Notes in Computer Science*, London, UK.
- [Fages et al., 2014] Fages, F., Floch, F.-M., Gay, S., Jovanovska, D., Rizk, A., Soliman, S., and Traynard, P. (2014). *BIOCHAM v3.7.3 Reference Manual*. INRIA.
- [Fages and Rizk, 2008] Fages, F. and Rizk, A. (2008). On temporal logic constraint solving for the analysis of numerical data time series. *Theoretical Computer Science*, 408(1):55–65.
- [Fages and Soliman, 2008] Fages, F. and Soliman, S. (2008). Abstract interpretation and types for systems biology. *Theoretical Computer Science*, 403(1):52–70.
- [Fages et al., 2004] Fages, F., Soliman, S., and Chabrier-Rivier, N. (2004). Modelling and querying interaction networks in the biochemical abstract machine BIOCHAM. *Journal of Biological Physics and Chemistry*, 4(2):64–73.
- [Fages and Traynard, 2014] Fages, F. and Traynard, P. (2014). Temporal logic modeling of dynamical behaviors: First-order patterns and solvers. In del Cerro, L. F. and Inoue, K., editors, *Logical Modeling of Biological Systems*, chapter 8, pages 291–323. John Wiley & Sons, Inc.
- [Fauré et al., 2006] Fauré, A., Naldi, A., Chaouiya, C., and Thieffry, D. (2006). Dynamical analysis of a generic Boolean model for the control of the mammalian cell cycle. *Bioinformatics (Oxford, England)*, 22(14):e124–31.
- [Fauré et al., 2009] Fauré, A., Naldi, A., Lopez, F., Chaouiya, C., Ciliberto, A., and Thieffry, D. (2009). Modular logical modelling of the budding yeast cell cycle. *Molecular Biosystems*, 5:1787–1796.
- [Feillet et al., 2014] Feillet, C., Krusche, P., Tamanini, F., Janssens, R. C., Downey, M. J., Martin, P., Teboul, M., Saito, S., Lévi, F., Bretschneider, T., van der Horst, G. T. J., Delaunay, F., and Rand, D. A. (2014). Phase locking and multiple oscillating attractors for the coupled mammalian clock and cell cycle. *Proceedings of the National Academy of Sciences of the United States of America*, 111(27):9928–9833.
- [Feillet et al., 2015] Feillet, C., van der Horst, G. T. J., Levi, F., Rand, D. A., and Delaunay, F. (2015). Coupling between the circadian clock and cell cycle oscillators: Implication for healthy cells and malignant growth. *Frontiers in Neurology*, 6(May):1–7.

- [Ferrell et al., 2011] Ferrell, J. E., Tsai, T. Y.-C., and Yang, Q. (2011). Modeling the cell cycle: why do certain circuits oscillate? *Cell*, 144(6):874–885.
- [Fisher and Henzinger, 2007] Fisher, J. and Henzinger, T. A. (2007). Executable cell biology. *Nature biotechnology*, 25(11):1239–49.
- [Fisher et al., 2015] Fisher, J., Köksal, A. S., Piterman, N., and Woodhouse, S. (2015). *Computer Aided Verification: 27th International Conference, CAV 2015, San Francisco, CA, USA, July 18–24, 2015, Proceedings, Part I*, chapter Synthesising Executable Gene Regulatory Networks from Single-Cell Gene Expression Data, pages 544–560. Springer International Publishing, Cham.
- [Fisher and Piterman, 2014] Fisher, J. and Piterman, N. (2014). *A Systems Theoretic Approach to Systems and Synthetic Biology I: Models and System Characterizations*, chapter Model Checking in Biology, pages 255–279. Springer Netherlands, Dordrecht.
- [Flobak et al., 2015] Flobak, Å., Baudot, A., Remy, E., Thommesen, L., Thieffry, D., Kuiper, M., and Lægreid, A. (2015). Discovery of drug synergies in gastric cancer cells predicted by logical modeling. *PLoS Computational Biology*, 11(8):e1004426.
- [Franke et al., 2010] Franke, R., Theis, F. J., and Klamt, S. (2010). From binary to multivalued to continuous models: the lac operon as a case study. *Journal of integrative bioinformatics*, 7(1):1–19.
- [Funahashi et al., 2008] Funahashi, A., Matsuoka, Y., Jouraku, A., Morohashi, M., Kikuchi, N., and Kitano, H. (2008). CellDesigner 3.5: A versatile modeling tool for biochemical networks. *Proceedings of the IEEE*, 96(8):1254–1265.
- [Gao et al., 2013] Gao, S., Kong, S., and Clarke, E. (2013). Satisfiability Modulo ODEs. In *FMCAD’13: Proc. of Formal Methods in Computer-Aided Design*, Lecture Notes in Computer Science. Springer-Verlag.
- [Garg et al., 2008] Garg, A., Di Cara, A., Xenarios, I., Mendoza, L., and De Micheli, G. (2008). Synchronous versus asynchronous modeling of gene regulatory networks. *Bioinformatics*, 24(17):1917–25.
- [Geng et al., 2003] Geng, Y., Yu, Q., Sicinska, E., Das, M., Schneider, J. E., Bhattacharya, S., Rideout, W. M., Bronson, R. T., Gardner, H., and Sicinski, P. (2003). Cyclin E ablation in the mouse. *Cell*, 114(4):431–443.
- [Gérard and Goldbeter, 2009] Gérard, C. and Goldbeter, A. (2009). Temporal self-organization of the cyclin/cdk network driving the mammalian cell cycle. *Proceedings of the National Academy of Sciences*, 106(51):21643–21648.
- [Gérard and Goldbeter, 2012] Gérard, C. and Goldbeter, A. (2012). Entrainment of the mammalian cell cycle by the circadian clock: Modeling two coupled cellular rhythms. *PLoS Comput Biol*, 8(21):e1002516.
- [Gérard et al., 2012] Gérard, C., Gonze, D., and Goldbeter, A. (2012). Effect of positive feedback loops on the robustness of oscillations in the network of cyclin-dependent kinases driving the mammalian cell cycle. *FEBS Journal*, 279(18):3411–3431.
- [Gery et al., 2006] Gery, S., Komatsu, N., Baldjyan, L., Yu, A., Koo, D., and Koeffler, H. (2006). The circadian gene *per1* plays an important role in cell growth and dna damage control in human cancer cells. *Mol Cell*, 22:375–382.
- [Gillespie, 1977] Gillespie, D. T. (1977). Exact stochastic simulation of coupled chemical reactions. *Journal of Physical Chemistry*, 81(25):2340–2361.
- [Glass and Kauffman, 1973] Glass, L. and Kauffman, S. A. (1973). The logical analysis of continuous, non-linear biochemical control networks. *J. Theor. Biol.*, 39(1):103–129.

- [Gong et al., 2011] Gong, H., Zuliani, P., Wang, Q., and Clarke, E. M. (2011). Formal analysis for logical models of pancreatic cancer. In *IEEE Conference on Decision and Control and European Control Conference*, pages 4855–4860. IEEE.
- [Gréchez-Cassiau et al., 2008] Gréchez-Cassiau, A., Rayet, B., Guillaumond, F., Teboul, M., , and Delaunay, F. (2008). The circadian clock component *bmal1* is a critical regulator of p21WAF1/CIP1 expression and hepatocyte proliferation. *J Biol Chem*, 283:4535–4542.
- [Grieco et al., 2013] Grieco, L., Calzone, L., Bernard-Pierrot, I., Radvanyi, F., Kahn-Perles, B., and Thieffry, D. (2013). Integrative modelling of the influence of MAPK network on cancer cell fate decision. *PLoS Comput. Biol.*, 9(10):e1003286.
- [Hansen and Ostermeier, 2001] Hansen, N. and Ostermeier, A. (2001). Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195.
- [Heitzler et al., 2012] Heitzler, D., Durand, G., Gallay, N., Rizk, A., Ahn, S., Kim, J., Violin, J. D., Dupuy, L., Gauthier, C., Piketty, V., Crépieux, P., Poupon, A., Clément, F., Fages, F., Lefkowitz, R. J., and Reiter, E. (2012). Competing G protein-coupled receptor kinases balance G protein and β -arrestin signaling. *Molecular Systems Biology*, 8(590).
- [Henley and Dick, 2012] Henley, S. a. and Dick, F. a. (2012). The retinoblastoma family of proteins and their regulatory functions in the mammalian cell division cycle. *Cell division*, 7(1):10.
- [Hong et al., 2014] Hong, C. I., Zamborszky, J., Baek, M., Labiscsak, L., Ju, K., Lee, H., Larrondo, L. F., Goity, A., Chong, H. S., Belden, W. J., and Csikasz-Nagy, A. (2014). Circadian rhythms synchronize mitosis in *Neurospora crassa*. *Proceedings of the National Academy of Sciences*, 111(4):1397–1402.
- [Irons, 2009] Irons, D. (2009). Logical analysis of the budding yeast cell cycle. *Journal of theoretical biology*, 257(4):543–559.
- [Ji et al., 2004] Ji, P., Jiang, H., Rekhman, K., Bloom, J., Ichetovkin, M., Pagano, M., and Zhu, L. (2004). An Rb-Skp2-p27 pathway mediates acute cell cycle inhibition by Rb and is retained in a partial-penetrance Rb mutant. *Molecular cell*, 16(1):47–58.
- [Ji and Zhu, 2005] Ji, P. and Zhu, L. (2005). Using Kinetic Studies to Uncover New Rb Functions in Inhibiting Cell Progression. *Cell cycle*, (March):373–375.
- [Kang et al., 2008] Kang, B., Li, Y.-Y., Chang, X., Liu, L., and Li, Y.-X. (2008). Modeling the effects of cell cycle m-phase transcriptional inhibition on circadian oscillation. *PLoS Comput Biol*, 4(3):e1000019.
- [Kauffman, 1969] Kauffman, S. A. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. *J Theor Biol*, 22(3):437–67.
- [Korenčič et al., 2014] Korenčič, A., Košir, R., Bordyugov, G., Lehmann, R., Rozman, D., and Herzog, H. (2014). Timing of circadian genes in mammalian tissues. *Scientific reports*, 4:5782.
- [Kotoshiba et al., 2014] Kotoshiba, S., Gopinathan, L., Pfeifferberger, E., Rahim, A., Vardy, L. A., Nakayama, K., Nakayama, K. I., and Kaldis, P. (2014). p27 is regulated independently of Skp2 in the absence of Cdk2. *Biochimica et biophysica acta*, 1843(2):436–45.
- [Kotoshibai et al., 2005] Kotoshibai, S., Kamura, T., Hara, T., Ishida, N., and Nakayama, K. I. (2005). Molecular dissection of the interaction between p27 and Kip1 ubiquitylation-promoting complex, the ubiquitin ligase that regulates proteolysis of p27 in G1 phase. *Journal of Biological Chemistry*, 280(18):17694–17700.
- [Krek, 1994] Krek, W. (1994). Negative regulation of the growth-promoting transcription factor E2F-1 by a stably bound cyclin A-dependent protein kinase. *Cell*, 78(1):161–172.

- [Le Novere, 2015] Le Novere, N. (2015). Quantitative and logic modelling of molecular and gene networks. *Nat. Rev. Genet.*, 16(3):146–158.
- [Lees et al., 1992] Lees, E., Faha, B., Dulic, V., Reed, S. I., and Harlow, E. (1992). Cyclin E/cdk2 and cyclin A/cdk2 kinases associate with p107 and E2F in a temporally distinct manner. *Genes and Development*, 6(Reed 1991):1874–1885.
- [Leloup and Goldbeter, 2003] Leloup, J.-C. and Goldbeter, A. (2003). Toward a detailed computational model for the mammalian circadian clock. *Proceedings of the National Academy of Sciences*, 100:7051–7056.
- [Levi et al., 2007] Levi, F., Focan, C., Karaboue, A., de la Valette, V., Focan-Hennard, D., Baron, B., Kreutz, F., and Giacchetti, S. (2007). Implications of circadian clocks for the rhythmic delivery of cancer therapeutics. *Adv. Drug Deliv. Rev.*, 59(9-10):1015–1035.
- [Lévi and Schibler, 2007] Lévi, F. and Schibler, U. (2007). Circadian rhythms: mechanisms and therapeutic implications. *Annual Review of Pharmacology and Toxicology*, 47:593–628.
- [Li et al., 2004] Li, F., Long, T., Lu, Y., Ouyang, Q., and Tang, C. (2004). The yeast cell-cycle network is robustly designed. *Proceedings of the National Academy of Sciences of the United States of America*, 101(14):4781–4786.
- [Li et al., 2013] Li, X.-M., Mohammad-Djafari, A., Dumitru, M., Dulong, S., Filipski, E., Siffroi-Fernandez, S., Mteyrek, A., Scaglione, F., Guettier, C., Delaunay, F., and Lévi, F. (2013). A Circadian Clock Transcription Model for the Personalization of Cancer Chronotherapy. *Cancer research*, pages 0008–5472.CAN–13–1528–.
- [Lincoln and Tiwari, 2004] Lincoln, P. and Tiwari, A. (2004). Symbolic systems biology: Hybrid modeling and analysis of biological networks. In Alur, R. and Pappas, G., editors, *Hybrid Systems: Computation and Control HSCC*, volume 2993 of *LNCS*, pages 660–672. Springer.
- [Liu et al., 2008] Liu, Y., Perdreau, S. a., Chatterjee, P., Wang, L., Kuan, S.-F., and Duensing, A. (2008). Imatinib mesylate induces quiescence in gastrointestinal stromal tumor cells through the CDH1-SKP2-p27Kip1 signaling axis. *Cancer research*, 68(21):9015–23.
- [Loyer et al., 2005] Loyer, P., Trembley, J. H., Katona, R., Kidd, V. J., and Lahti, J. M. (2005). Role of CDK/cyclin complexes in transcription and RNA splicing. *Cellular Signalling*, 17(9):1033–1051.
- [Markey and Schnoebelen, 2003] Markey, N. and Schnoebelen, P. (2003). Model checking a path. In *CONCUR 2003*, volume 2761 of *Lecture Notes in Computer Science*, pages 251–265. Springer-Verlag.
- [Matsuo et al., 2003] Matsuo, T., Yamaguchi, S., Mitsui, S., Emi, A., Shimoda, F., and Okamura, H. (2003). Control mechanism of the circadian clock for timing of cell division in vivo. *Science*, 302(5643):255–259.
- [Meyer and Rape, 2011] Meyer, H. J. and Rape, M. (2011). Processive ubiquitin chain formation by the anaphase-promoting complex. *Seminars in Cell and Developmental Biology*, 22(6):544–550.
- [Mombach et al., 2014] Mombach, J. C. M., Bugs, C. A., and Chaouiya, C. (2014). Modelling the onset of senescence at the g1/s cell cycle checkpoint. *BMC Genomics*, 15 Suppl 7:S7.
- [Montagnoli et al., 1999] Montagnoli, a., Fiore, F., Eytan, E., Carrano, a. C., Draetta, G. F., Hershko, a., and Pagano, M. (1999). Ubiquitination of p27 is regulated by CDK dependent phosphorylation and trimeric complex formation. *Genes & Dev*, 13:1181–1189.
- [Monteiro and Chaouiya, 2012] Monteiro, P. and Chaouiya, C. (2012). Efficient verification for logical models of regulatory networks. In Rocha, M. P., Luscombe, N., Fdez-Riverola, F., and Rodríguez, J. M. C., editors, *6th International Conference on Practical Applications of Computational Biology & Bioinformatics*, volume 154 of *Advances in Intelligent and Soft Computing*, pages 259–267. Springer Berlin Heidelberg.

- [Monteiro et al., 2014] Monteiro, P. T., Abou-Jaoude, W., Thieffry, D., and Chaouiya, C. (2014). Model checking logical regulatory networks. In *Discrete Event Systems*, volume 12, pages 170–175.
- [Monteiro et al., 2008] Monteiro, P. T., Ropers, D., Mateescu, R., Freitas, A. T., and de Jong, H. (2008). Temporal logic patterns for querying dynamic models of cellular interaction networks. *Bioinformatics*, 24(16):i227–i233.
- [Nagoshi et al., 2004] Nagoshi, E., Saini, C., Bauer, C., Laroche, T., Naef, F., and Schibler, U. (2004). Circadian gene expression in individual fibroblasts: cell-autonomous and self-sustained oscillators pass time to daughter cells. *Cell*, 119:693–705.
- [Nakayama et al., 2000] Nakayama, K., Nagahama, H., Minamishima, Y. A., Matsumoto, M., Nakamichi, I., Kitagawa, K., Shirane, M., Tsunematsu, R., Tsukiyama, T., Ishida, N., Kitagawa, M., and Hatakeyama, S. (2000). Targeted disruption of Skp2 results in accumulation of cyclin E and p27(Kip1), polyploidy and centrosome overduplication. *The EMBO journal*, 19(9):2069–81.
- [Naldi et al., 2009] Naldi, A., Lopez, F., Thieffry, D., and Chaouiya, C. (2009). Qualitative modelling and analysis of biological regulatory networks with GINsim 2.3. *Biosystems*, 97:134–139.
- [Naldi et al., 2012] Naldi, A., Monteiro, P. T., and Chaouiya, C. (2012). Efficient handling of large signalling-regulatory networks by focusing on their core control. In *Proceedings of the Computational Methods in Systems Biology - 10th International Conference, CMSB*, pages 288–306.
- [Naldi et al., 2015] Naldi, A., Monteiro, P. T., Müssel, C., the Consortium for Logical Models, Tools, Kestler, H. A., Thieffry, D., Xenarios, I., Saez-Rodriguez, J., Helikar, T., and Chaouiya, C. (2015). Cooperative development of logical modelling standards and tools with colomoto. *Bioinformatics*.
- [Naldi et al., 2011] Naldi, A., Remy, E., Thieffry, D., and Chaouiya, C. (2011). Dynamically consistent reduction of logical regulatory graphs. *Theoretical Computer Science*, 412(21):2207–2218. Selected Papers from the 7th International Conference on Computational Methods in Systems Biology.
- [Narasimha et al., 2014] Narasimha, A. M., Kaulich, M., Shapiro, G. S., Choi, Y. J., Sicinski, P., and Dowdy, S. F. (2014). Cyclin D activates the Rb tumor suppressor by mono-phosphorylation. *eLife*, 3:e02872.
- [Novák and Tyson, 2004] Novák, B. and Tyson, J. J. (2004). A model for restriction point control of the mammalian cell cycle. *Journal of Theoretical Biology*, 230:1383–1388.
- [Novák and Tyson, 2008] Novák, B. and Tyson, J. J. (2008). Design principles of biochemical oscillators. *Nature reviews. Molecular cell biology*, 9(12):981–91.
- [Ohtsubo et al., 1995] Ohtsubo, M., Theodoras, a. M., Schumacher, J., Roberts, J. M., and Pagano, M. (1995). Human cyclin E, a nuclear protein essential for the G1-to-S phase transition. *Molecular and cellular biology*, 15(5):2612–2624.
- [Perez-Roger, 1997] Perez-Roger, I. (1997). Myc activation of cyclin e/cdk2 kinase involves induction of cyclin e gene transcription and inhibition of p27(kip1) binding to newly formed complexes. *Oncogene*, 14(20):2373–81.
- [Pittendrigh, 1960] Pittendrigh, C. S. (1960). Circadian rhythms and the circadian organization of living systems. *Cold Spring Harb. Symp. Quant. Biol.*, 25:159–184.
- [Qu et al., 2003] Qu, Z., MacLellan, W. R., and Weiss, J. N. (2003). Dynamics of the cell cycle: checkpoints, sizers, and timers. *Biophysics Journal*, 85(6):3600–3611.

- [Rape and Kirschner, 2004] Rape, M. and Kirschner, M. W. (2004). Autonomous regulation of the anaphase-promoting complex couples mitosis to S-phase entry. *Nature*, 432(7017):588–95.
- [Rape et al., 2006] Rape, M., Reddy, S. K., and Kirschner, M. W. (2006). The Processivity of Multiubiquitination by the APC Determines the Order of Substrate Degradation. *Cell*, 124(1):89–103.
- [Reddy et al., 1996] Reddy, V. N., Liebman, M. N., and Mavrovouniotis, M. L. (1996). Qualitative analysis of biochemical reaction systems. *Computers in Biology and Medicine*, 26(1):9–24.
- [Regev et al., 2001] Regev, A., Silverman, W., and Shapiro, E. Y. (2001). Representation and simulation of biochemical processes using the pi-calculus process algebra. In *Proceedings of the sixth Pacific Symposium of Biocomputing*, pages 459–470.
- [Relógio et al., 2014] Relógio, A., Thomas, P., Medina-Pérez, P., Reischl, S., Bervoets, S., Gloc, E., Riemer, P., Mang-Fatehi, S., Maier, B., Schäfer, R., Leser, U., Herzog, H., Kramer, A., and Sers, C. (2014). Ras-Mediated Dereglulation of the Circadian Clock in Cancer. *PLoS Genetics*, 10(5).
- [Relógio et al., 2011] Relógio, A., Westermarck, P. O., Wallach, T., Schellenberg, K., Kramer, A., and Herzog, H. (2011). Tuning the mammalian circadian clock: robust synergy of two loops. *PLoS computational biology*, 7(12):e1002309.
- [Rivard et al., 1996] Rivard, N., L’Allemain, G., Bartek, J., and Pouyssegur, J. (1996). Abrogation of p27Kip1 by cDNA antisense suppresses quiescence (G0 state) in fibroblasts. *J. Biol. Chem.*, 271(31):18337–18341.
- [Rizk, 2011] Rizk, A. (2011). *Résolution de contraintes temporelles pour l’analyse de systèmes biologiques*. PhD thesis, Université Denis Diderot, Paris.
- [Rizk et al., 2008] Rizk, A., Batt, G., Fages, F., and Soliman, S. (2008). On a continuous degree of satisfaction of temporal logic formulae with applications to systems biology. In Heiner, M. and Uhrmacher, A., editors, *CMSB’08: Proceedings of the fourth international conference on Computational Methods in Systems Biology*, volume 5307 of *Lecture Notes in Computer Science*, pages 251–268. Springer-Verlag.
- [Rizk et al., 2009] Rizk, A., Batt, G., Fages, F., and Soliman, S. (2009). A general computational method for robustness analysis with applications to synthetic gene networks. *Bioinformatics*, 12(25):il69–il78.
- [Rizk et al., 2011] Rizk, A., Batt, G., Fages, F., and Soliman, S. (2011). Continuous valuations of temporal logic specifications with applications to parameter optimization and robustness measures. *Theoretical Computer Science*, 412(26):2827–2839.
- [Roşu and Havelund, 2005] Roşu, G. and Havelund, K. (2005). Rewriting-based techniques for runtime verification. *Automated Software Engineering*, 12(2):151–197.
- [Seth et al., 1993] Seth, A., Gupta, S., and Davis, R. J. (1993). Cell cycle regulation of the c-Myc transcriptional activation domain. *Molecular and cellular biology*, 13(7):4125–4136.
- [Srivastava, 1993] Srivastava, D. (1993). Subsumption and indexing in constraint query languages with linear arithmetic constraints. *Annals of Mathematics and Artificial Intelligence*, 8:315–343.
- [Stal and Krumbein, 1987] Stal, L. J. and Krumbein, W. E. (1987). Temporal separation of nitrogen fixation and photosynthesis in the filamentous, non-heterocystous cyanobacterium *oscillatoria* sp. *Archives of Microbiology*, 149(1):76–80.
- [Stoll et al., 2012] Stoll, G., Viara, E., Barillot, E., and Calzone, L. (2012). Continuous time boolean modeling for biological signaling: application of gillespie algorithm. *BMC Systems Biology*, 6:116.

- [Stoma et al., 2013] Stoma, S., Donzé, A., Bertaux, F., Maler, O., and Batt, G. (2013). STL-based analysis of TRAIL-induced apoptosis challenges the notion of type I/type II cell line classification. *PLoS Computational Biology*, 9(5):e1003056.
- [Sugita, 1963] Sugita, M. (1963). Functional analysis of chemical systems in vivo using a logical circuit equivalent. ii. the idea of a molecular automation. *J Theor Biol*, 4(2):179–92.
- [Thobe et al., 2014] Thobe, K., Streck, A., Klarner, H., and Siebert, H. (2014). *Computational Methods in Systems Biology: 12th International Conference, CMSB 2014, Manchester, UK, November 17-19, 2014, Proceedings*, chapter Model Integration and Crosstalk Analysis of Logical Regulatory Networks, pages 32–44. Springer International Publishing, Cham.
- [Thomas, 1973] Thomas, R. (1973). Boolean formalization of genetic control circuits. *J Theor Biol*, 42(3):563–85.
- [Thomas, 1991] Thomas, R. (1991). Regulatory networks seen as asynchronous automata : a logical description. *Journal of Theoretical Biology*, 153:1–23.
- [Thomas and D’Ari, 1990] Thomas, R. and D’Ari, R. (1990). *Biological feedback*. CRC Press, Boca Raton.
- [Thomas et al., 1995] Thomas, R., Thieffry, D., and Kaufman, M. (1995). Dynamical behaviour of biological regulatory networks-i. biological role of feedback loops and practical use of the concept of the loop-characteristic state. *Bulletin of mathematical biology*, 57(2):247–276.
- [Traynard et al., 2014] Traynard, P., Fages, F., and Soliman, S. (2014). Trace simplifications preserving temporal logic formulae with case study in a coupled model of the cell cycle and the circadian clock (best student paper award). In *CMSB’14: Proceedings of the twelfth international conference on Computational Methods in Systems Biology*, number 8859 in Lecture Notes in Bioinformatics, pages 114–128. Springer-Verlag.
- [Traynard et al., 2015a] Traynard, P., Fages, F., and Soliman, S. (2015a). Model-based investigation of the effect of the cell cycle on the circadian clock through transcription inhibition during mitosis. In Roux, O. and Bourdon, J., editors, *CMSB’15: Proceedings of the thirteenth international conference on Computational Methods in Systems Biology*, volume 9308 of *Lecture Notes in Bioinformatics*, pages 208–221. Springer-Verlag.
- [Traynard et al., 2015b] Traynard, P., Fauré, A., Fages, F., and Thieffry, D. (2015b). Logical modeling of the mammalian cell cycle. viXra:1512.0337.
- [Tyson and Novák, 2015] Tyson, J. J. and Novák, B. (2015). Models in biology: lessons from modeling regulation of the eukaryotic cell cycle. *BMC Biology*, 13:46.
- [Ünsal-Kaçmaz et al., 2005] Ünsal-Kaçmaz, K., Mullen, T. E., Kaufmann, W. K., and Sancar, A. (2005). Coupling of human circadian and cell cycles by the timeless protein. *Molecular and Cellular Biology*, 25(8):3109–3116.
- [Weinberg, 1995] Weinberg, R. A. (1995). The retinoblastoma protein and cell cycle control. *Cell*, 81(3):323–330.
- [Weisenberger and Scheer, 1995] Weisenberger, D. and Scheer, U. (1995). A possible mechanism for the inhibition of ribosomal rna gene transcription during mitosis. *Journal of Cell Biology*.
- [Wilkinson, 2009] Wilkinson, D. J. (2009). Stochastic modelling for quantitative description of heterogeneous biological systems. *Nature reviews. Genetics*, 10(2):122–133.
- [Wolkenhauer, 2014] Wolkenhauer, O. (2014). Why model? *Frontiers in physiology*, 5:21.
- [Wong et al., 2014] Wong, J. V., Dong, P., Nevins, J. R., Mathey-Prevot, B., and You, L. (2014). Control of E2F dynamics in cell cycle entry. *Cell Cycle*, 10(18):3086–3094.

- [Yoo et al., 2004] Yoo, S., Yamazaki, S., Lowrey, P., Shimomura, K., Ko, C., Buhr, E., Siepk, S., Hong, H., Oh, W., Yoo, O., Menaker, M., and Takahashi, J. (2004). PERIOD2::LUCIFERASE real-time reporting of circadian dynamics reveals persistent circadian oscillations in mouse peripheral tissues. *PNAS*, pages 5339–5346.
- [Zámborszky et al., 2014] Zámborszky, J., Csikász-Nagy, A., and Hong, C. I. (2014). *Neurospora crassa* as a model organism to explore the interconnected network of the cell cycle and the circadian clock. *Fungal genetics and biology : FG & B*, 71:52–7.

A. Appendix

A.1 Perturbation properties and CTL formulae for the logical model of the cell cycle

Experiment	Phenotype	Fauré et al. (2006)	Updated model
p27 KO CycE E1	cycle in absence of growth factor Viable; cycle in absence of growth factor; less serum-dependent.	Disagreement For CycD=0, cycle or steady state. Questionable	Disagreement Cycle for CycD=1, steady state for CycD=0. Questionable
CycE KO CycE KO CycA KO	Cell cycle arrest before S phase Cell cycle arrest before S phase	Disagreement Disagreement	OK OK
p27 E1	Cell cycle arrest in presence of growth factors.	OK	OK
p27 and CycA E1	Cell cycle arrest in presence of growth factors.	OK	OK
p27 and CycE E1	Cell cycle arrest in presence of growth factors.	OK	OK
p27 and E2F E1	Cell cycle arrest in presence of growth factors.	OK	OK
Rb E2	G1 arrest + p27 accumulation	OK	OK
Cdh1 KO	Viable cell cycle in presence or absence of growth factors.	OK	OK
UbcH10 KO	Cell cycle arrested in M phase, destruction of cyclin A and B inhibited.	Disagreement	OK
E2F overexpression in Rb-/- E2F E1 Rb KO	Viable cell cycle in presence or absence of growth factors	OK	OK
E2F overexpression in Rb+/+	Targets of E2F are expressed even in absence of growth factor but the cell cycle is arrested.	Cyclic attractor with CycD=0 or CycD=1 Disagreement	OK
Loss of Rb Rb KO	Viable cell cycle in presence or absence of growth factors	OK	OK
Rb induction (early kinetics) RbR661W	Cell cycle arrest Cell cycle in presence of growth factors, cell cycle arrest in absence Cyclic attractor with CycD=0 or CycD=1	Disagreement Disagreement	OK OK
p27 KO (early kinetics)	Viable cell cycle in presence or absence of growth factors	OK	OK
Skp2 KO	Cell cycle arrest or endoreplication with Cyclin E and p27 accumulation.	-	OK
Skp2 KO p27 KO	Cell cycle in presence of growth factors, cycle arrest in absence	-	OK
Skp2 overexpression and Rb induction (early kinetics)	Viable cell cycle in presence or absence of growth factors.	-	OK

Table A.1: Perturbation properties and simulation results for the initial model of the mammalian cell cycle [Fauré et al., 2006] and the updated model.

Phenotypes	Sequences of events verified by model checking	Fauré et al. (2006)	Updated model
Feedforward loop that ensures that cells advance to S-phase irreversibly: phosphorylation of Rb by CycE to release E2F and synthesize more CycE, and phosphorylation of p27.	1. Correct sequence: CycE=0 & CycA=0 & p27=1 -> CycE=1 & CycA=0 & p27=1 -> CycE=1 & CycA=0 & p27=0 -> CycE=1 & CycA=1 & p27=0 -> CycE=0 & CycA=1 & p27=0	No	OK
	2. Incorrect sequence starting from CycB=0 & Cdh1=1 (G1) or Rb=1 & CycB=0 & Cdh1=1 for the updated model: CycE=0 & CycA=0 & p27=1 -> CycE=0 & CycA=0 & p27=0 -> CycE=1 & CycA=0 & p27=0 -> CycE=1 & CycA=1 & p27=0 -> CycE=0 & CycA=1 & p27=0	No	OK
	3. Incorrect sequence starting from CycB=0 & Cdh1=1 (G1) or Rb=1 & CycB=0 & Cdh1=1 for the updated model: CycE=0 & CycA=0 & p27=1 -> CycE=1 & CycA=0 & p27=1 -> CycE=1 & CycA=1 & p27=1 -> CycE=1 & CycA=1 & p27=0 -> CycE=0 & CycA=1 & p27=0	No	OK
	4. Incorrect sequence starting from CycB=0 & Cdh1=1 (G1) or Rb=1 & CycB=0 & Cdh1=1 for the updated model: CycE=0 & CycA=0 & p27=1 -> CycE=1 & CycA=0 & p27=1 -> CycE=1 & CycA=0 & p27=1 -> CycE=1 & CycA=1 & p27=1 -> CycE=1 & CycA=1 & p27=1 -> CycE=0 & CycA=1 & p27=1	OK	OK
As long as E2F is inactivated, the cell remains in the G1 phase.	5. Incorrect sequence starting from CycB=0 & Cdh1=1 (G1) or Rb=1 & CycB=0 & Cdh1=1 for the updated model: CycE=0 & CycA=0 & p27=1 & E2F=0 -> CycE=1 & CycA=0 & p27=1 & E2F=0 -> CycE=1 & CycA=0 & p27=0 & E2F=0 -> CycE=1 & CycA=0 & p27=0 & E2F=0 -> CycE=0 & CycA=1 & p27=0 & E2F=0	OK	OK
Cyclin A expression is delayed relative to E2F and cyclin E.	6. Incorrect sequence starting from CycB=0 & Cdh1=1 (G1) or Rb=1 & CycB=0 & Cdh1=1 for the updated model: CycE=0 & CycA=0 & E2F=0 -> CycE=0 & CycA=0 & E2F=1 -> CycE=1 & CycA=0 & E2F=1 -> CycE=1 & CycA=0 & E2F=0 -> CycE=1 & CycA=0 & E2F=0 -> CycE=1 & CycA=1 & E2F=0	OK	OK
Sequential activation of CycE and CycA.	7. Incorrect sequence starting from CycB=0 & Cdh1=1 (G1) or Rb=1 & CycB=0 & Cdh1=1 for the updated model: CycE=0 & CycA=0 -> CycE=0 & CycA=1 -> CycE=1 & CycA=1	No	OK
Degradation of B cyclins by the APC/C complex is required for mitosis to take place.	8. Correct sequence from Rb=0 & Cdh1=0 & p27=0 & Skp2=1 & CycE=0 (G2): CycA=1 & CycB=0 & Cdc20=0 -> CycA=1 & CycB=1 & Cdc20=0 -> CycA=1 & CycB=1 & Cdc20=1 -> CycA=0 & CycB=1 & Cdc20=1 -> CycA=0 & CycB=0 & Cdc20=1 -> CycA=0 & CycB=0 & Cdc20=0 9. Incorrect sequence from Rb=0 & Cdh1=0 & p27=0 & Skp2=1 & CycE=0 (G2): CycA=1 & CycB=0 & Cdc20=0 -> CycA=1 & CycB=1 & Cdc20=0 -> CycA=1 & CycB=1 & Cdc20=1 -> CycA=0 & CycB=1 & Cdc20=1 -> CycA=0 & CycB=1 & Cdc20=0 -> CycA=0 & CycB=0 & Cdc20=0 10. Incorrect sequence from Rb=0 & Cdh1=0 & p27=0 & Skp2=1 & CycE=0 (G2): CycA=1 & CycB=0 & Cdc20=0 -> CycA=1 & CycB=1 & Cdc20=0 -> CycA=0 & CycB=1 & Cdc20=0 -> CycA=0 & CycB=1 & Cdc20=0 -> CycA=0 & CycB=0 & Cdc20=0	OK	OK

Table A.2: Sequence properties and simulation results for the initial model of the mammalian cell cycle [Fauré et al., 2006] and the updated model.

A.1. Perturbation properties and CTL formulae for the logical model of the cell cycle

Phenotypes	Sequences of events verified by model checking	Fauré et al. (2006)	Updated model
Cdh1 is not required for mitotic exit.	11. Correct sequence from Rb=0 & Cdh1=0 & p27=0 & Skp2=1 & CycE=0 (G2): CycA=1 & CycB=0 & Cdc20=0 & Cdh1=0 -> CycA=1 & CycB=1 & Cdc20=0 & Cdh1=0 -> CycA=1 & CycB=1 & Cdc20=1 & Cdh1=0 -> CycA=0 & CycB=1 & Cdc20=1 & Cdh1=0 -> CycA=0 & CycB=0 & Cdc20=1 & Cdh1=0 -> CycA=0 & CycB=0 & Cdc20=0 & Cdh1=0	OK	OK
UBEC2 is required for the destruction of mitotic cyclins and other mitosis-related substrates.	12. Incorrect sequence from Rb=0 & Cdh1=0 & p27=0 & Skp2=1 & CycE=0 (G2): CycA=1 & CycB=0 & Cdc20=0 & UbcH10=0 -> CycA=1 & CycB=1 & Cdc20=0 & UbcH10=0 -> CycA=1 & CycB=1 & Cdc20=1 & UbcH10=0 -> CycA=0 & CycB=1 & Cdc20=1 & UbcH10=0 -> CycA=0 & CycB=0 & Cdc20=1 & UbcH10=0 -> CycA=0 & CycB=0 & Cdc20=0 & UbcH10=0	No	OK
Degradation of CycE in S phase, activation of CycB in G2 phase	13. Incorrect sequence starting from CycB=0 & Cdh1=1 (G1) or Rb=1 & CycB=0 & Cdh1=1 for the updated model: CycE=0 & CycA=0 & CycB=0 -> CycE=1 & CycA=0 & CycB=0 -> CycE=1 & CycA=1 & CycB=0 -> CycE=1 & CycA=1 & CycB=1 -> CycE=0 & CycA=1 & CycB=1 -> CycE=0 & CycA=0 & CycB=1 -> CycE=0 & CycA=0 & CycB=0	No	No
Synthesis of CycA in in early S phase, activation of CycB in G2 phase.	14. Incorrect sequence starting from CycB=0 & Cdh1=1 (G1) or Rb=1 & CycB=0 & Cdh1=1 for the updated model: CycE=0 & CycA=0 & CycB=0 -> CycE=1 & CycA=0 & CycB=0 -> CycE=1 & CycA=0 & CycB=1 -> CycE=1 & CycA=1 & CycB=1	OK	OK
Degradation of CycA in early mitosis, degradation of CycB in late mitosis	15. Incorrect sequence starting from CycB=0 & Cdh1=1 (G1) or Rb=1 & CycB=0 & Cdh1=1 for the updated model: CycE=0 & CycA=0 & CycB=0 -> CycE=1 & CycA=0 & CycB=0 -> CycE=1 & CycA=1 & CycB=0 -> CycE=0 & CycA=1 & CycB=0 -> CycE=0 & CycA=1 & CycB=1 -> CycE=0 & CycA=1 & CycB=0 -> CycE=0 & CycA=0 & CycB=0	No	No
Complete removal of cyclin B1 is essential to prevent the return of the spindle checkpoint following sister chromatid disjunction.	16. Incorrect sequence starting from CycB=0 & Cdh1=1 (G1) or Rb=1 & CycB=0 & Cdh1=1 for the updated model: CycE=0 & CycA=0 & CycB=0 -> CycE=1 & CycA=0 & CycB=0 -> CycE=1 & CycA=1 & CycB=0 -> CycE=0 & CycA=1 & CycB=0 -> CycE=0 & CycA=1 & CycB=1 -> CycE=0 & CycA=0 & CycB=1 -> CycE=1 & CycA=0 & CycB=1	OK	No
Experimentally observed expressions of each component	17. Correct sequence defined in Table 9	OK for a sequence without p27, Rb and Skp2	OK

Table A.3: Sequence properties and simulation results for the initial model of the mammalian cell cycle [Fauré et al., 2006] and the updated model.

A.2 Toy oscillator model

This toy model reproduces the dynamics of a self-inhibited gene. It is a simple multi-component differential oscillator described in [Novák and Tyson, 2008] and schematized in Figure 3.7. Based on a delayed negative-feedback loop, it is composed of only three molecular species represented with three ODEs with nine kinetic parameters. The molecules X , Y_{cyto} and Y_{nucl} represent the gene mRNA, protein in the cytoplasm and protein in the nucleus, respectively. Y_{nucl} down-regulates the expression of X (allosteric binding with Michaelis-Menten kinetics). X activates Y_{cyto} and Y_{cyto} is translocated into Y_{nucl} . All these reactions, along with the degradations, follow mass-action kinetics, except the degradation of Y_{cyto} which follows a Hill function. The transport of the protein from the cytoplasm to the nucleus provides the necessary delay enabling the rise of oscillations. It is defined by the following set of ODEs:

$$\begin{aligned}\frac{d[X]}{dt} &= \frac{a \cdot k1 \cdot Kd^p}{Kd^p + [Y_{nucl}]^p} - a \cdot kdx \cdot [X] \\ \frac{d[Y_{cyto}]}{dt} &= a \cdot ksy \cdot [X] + a \cdot kn \cdot [Y_{nucl}] - a \cdot kc \cdot [Y_{cyto}] - \frac{a \cdot k2 \cdot [Y_{cyto}]}{Km + [Y_{cyto}]} \\ \frac{d[Y_{nucl}]}{dt} &= a \cdot kc \cdot [Y_{cyto}] - a \cdot kn + a \cdot kd \cdot [Y_{nucl}]\end{aligned}$$

The initial concentrations are not calibrated and therefore kept null for each compound. The kinetic parameters values are listed in the following table:

Parameter	Value
p	5
Km	0.2
Kd	1
$k1$	0.5
kdx	0.05
ksy	0.8
$k2$	1
kc	0.1
kn	0.05
kd	0.1
a	1.63

A.3 Models studied in Chapter 5

This appendix provides the Biocham files encoding the models used in Chapter 5: the cell cycle model from [Calzone and Soliman, 2006], the circadian clock model from [Relógio et al., 2011], and a model combining the two with an additional coupling.

Cell cycle model

```
% Cell cycle adapted from Qu et al, Biophysical journal , 2003
% Initial concentrations
present(preMPF,0.279939269).
present(MPF,0.747404087).
present(C25,0.629659946).
present(C25P,0.370340054).
present(Wee1,0.00223979517).
present(Wee1~{p},0.132059534).
present(APC,0.0303504407).
present(IE,0.41856651).
present(mWee1,0.141190132).
present(APCi,0.973282491).
```

```

% Kinetic parameters
parameter(kd25,1).
parameter(kdmpfp,0.25).
parameter(kampf,2.2).
parameter(kaapc,1).
parameter(ksweemp,0.001).
parameter(kdweem,1).
parameter(Kweem,0.2).
parameter(kwpcn,5).
parameter(Vac,1).
parameter(Kac25,0.01).
parameter(Kic25,0.01).
parameter(kdmpf,2).
parameter(Vaw,1).
parameter(Vapc,0.1).
parameter(Kaw,0.005).
parameter(Kiw,0.005).
parameter(Vapw,0.1).
parameter(Kapc,0.01).
parameter(ks25,1).
%parameter(kswee,0.5).
parameter(kswee,1.5).
%parameter(kimpf,1.5).
parameter(kimpf,2.5).
parameter(Vic,0.2).
parameter(kiapc,0.5).
parameter(kdwee,1).
parameter(Viw,0.2).
parameter(ksmpf,0.35).
parameter(kdie,0.1).
parameter(ksiep,0).
parameter(ksie,0.3).
parameter(kaapcp,0).
% Reaction rules
% MPF – preMPF
ksmpf for _=>preMPF.
kdmpfp*[preMPF] for preMPF=>_..
kdmpfp*[MPF] for MPF=>_..
kdmpf*[APC]*[MPF] for MPF=[APC]=>_..
kdmpf*[APC]*[preMPF] for preMPF=[APC]=>_..
kampf*[C25P]*[preMPF] for preMPF=[C25P]=>MPF.
kimpf*[Wee1]*[MPF] for MPF=[Wee1]=>preMPF.
% Wee1
Viw*[Wee1~{p}]/(Kiw+[Wee1~{p}]) for Wee1~{p}=>Wee1.
(Vapw+Vaw*[MPF])*[Wee1]/(Kaw+[Wee1]) for Wee1=[MPF]=>Wee1~{p}.
ksweemp for _=> mWee1.
kdweem*[mWee1] for mWee1=>_..
kswee*[mWee1] for _=[mWee1]=>Wee1.
kdwee*[Wee1] for Wee1=>_..
kdwee*[Wee1~{p}] for Wee1~{p}=>_..
% Cdc25
Vic*[C25P]/(Kic25+[C25P]) for C25P=>C25.
(Vapc+Vac*[MPF])*[C25]/(Kac25+[C25]) for C25=[MPF]=>C25P.
ks25 for _=>C25.
kd25*[C25] for C25=>_..
kd25*[C25P] for C25P=>_..
% APC
ksiep for _=>IE.
ksie*[MPF] for _=[MPF]=>IE.
kdie*[IE] for IE=>_..
(kaapcp+kaapc*[IE])*[APCi]/(Kapc+[APCi]) for APCi=[IE]=>APC.
kiapc*[APC]/(Kapc+[APC]) for APC=>APCi.

```

Circadian clock model

```

% Parameters
parameter(dx1,0.08).
parameter(dx2,0.06).
parameter(dx3,0.09).
parameter(dx5,0.17).
parameter(dx6,0.12).
parameter(dx7,0.15).
parameter(dy1,0.3).
parameter(dy2,0.2).
parameter(dy3,2).
parameter(dy4,0.2).
parameter(dy5,1.6).

```

A. APPENDIX

```
parameter(dz1,0.23).
parameter(dz2,0.25).
parameter(dz3,0.6).
parameter(dz4,0.2).
parameter(dz5,0.2).
parameter(dz6,0.31).
parameter(dz7,0.3).
parameter(dz8,0.73).
parameter(kfx1,2.3).
parameter(kdx1,0.01).
parameter(kfz4,1).
parameter(kdz4,1).
parameter(kfz5,1).
parameter(kdz5,1).
parameter(kphz2,2).
parameter(kdphz3,0.05).
parameter(V1maxp,1).
parameter(V2maxp,2.92).
parameter(V3maxp,1.9).
parameter(V4maxp,10.9).
parameter(V5maxp,1).
macro(V1max,V1maxp).
macro(V2max,V2maxp).
macro(V3max,V3maxp).
macro(V4max,V4maxp).
macro(V5max,V5maxp).
parameter(kt1,3).
parameter(ki1,0.9).
parameter(kt2,2.4).
parameter(ki2,0.7).
parameter(ki21,5.2).
parameter(kt3,2.07).
parameter(ki3,3.3).
parameter(kt4,0.9).
parameter(ki4,0.4).
parameter(kt5,8.35).
parameter(ki5,1.94).
parameter(a,12).
parameter(d,12).
parameter(g,5).
parameter(h,5).
parameter(i,12).
parameter(kp1,0.4).
parameter(kp2,0.26).
parameter(kp3,0.37).
parameter(kp4,0.76).
parameter(kp5,1.21).
parameter(kiz4,0.2).
parameter(kiz5,0.1).
parameter(kiz6,0.5).
parameter(kiz7,0.1).
parameter(kiz8,0.1).
parameter(kex2,0.02).
parameter(kex3,0.02).
parameter(b,5).
parameter(c,7).
parameter(e,6).
parameter(f,4).
parameter(f1,1).
parameter(v,6).
parameter(w,2).
parameter(p,6).
parameter(q,3).
parameter(n,2).
parameter(m,5).
parameter(y10,0).
parameter(y20,0).
parameter(y30,0).
parameter(y40,0).
parameter(y50,0).
parameter(coefclock,1.05).
macro(PC,[Cry-Per~{p}::nucl]+[Cry-Per_nucl]).

% Initial concentrations
present(Bmal-Clock_nucl,2.10424393).
present(Cry-Per~{p}::nucl,1.33350922).
present(Cry-Per_nucl,0.416825761).
present(RevErb_nucl,2.34469812).
present(Ror_nucl,5.63477788).
present(mPER,0.487066085).
present(mCRY,3.55456577).
present(mREVERB,2.51320081).
```

```

present(mROR,2.79852328).
present(mBMAL,1.14062417).
present(Cry_cyto,3.99608062).
present(Per_cyto,0.0704055565).
present(Per~{p}::cyto,0.103699981).
present(Cry-Per~{p}::cyto,0.336017237).
present(Cry-Per_cyto,0.238146903).
present(RevErb_cyto,1.06505599).
present(Ror_cyto,5.42605629).

% Reaction rules
MA(kiz5*coefclock) for Cry-Per_cyto => Cry-Per_nucl.
MA(kex3*coefclock) for Cry-Per_nucl => Cry-Per_cyto.
MA(dx3*coefclock) for Cry-Per_nucl => _.
MA(kiz4*coefclock) for Cry-Per~{p}::cyto => Cry-Per~{p}::nucl.
MA(kex2*coefclock) for Cry-Per~{p}::nucl => Cry-Per~{p}::cyto.
MA(dx2*coefclock) for Cry-Per~{p}::nucl => _.
MA(kfz5*coefclock) for Per_cyto + Cry_cyto => Cry-Per_cyto.
MA(kdz5*coefclock) for Cry-Per_cyto => Per_cyto+Cry_cyto.
MA(dz5*coefclock) for Cry-Per_cyto => _.
MA(kfz4*coefclock) for Per~{p}::cyto + Cry_cyto => Cry-Per~{p}::cyto.
MA(kdz4*coefclock) for Cry-Per~{p}::cyto => Per~{p}::cyto+Cry_cyto.
MA(dz4*coefclock) for Cry-Per~{p}::cyto => _.
MA(kphz2*coefclock) for Per_cyto => Per~{p}::cyto.
MA(kdphz3*coefclock) for Per~{p}::cyto => Per_cyto.
MA(dz3*coefclock) for Per~{p}::cyto => _.
MA(kp1*coefclock) for _=[mPER] => Per_cyto.
kp1*y10*coefclock for _ => Per_cyto.
MA(dz2*coefclock) for Per_cyto => _.
MA(kp2*coefclock) for _=[mCRY] => Cry_cyto.
kp2*y20*coefclock for _ => Cry_cyto.
MA(dz1*coefclock) for Cry_cyto => _.
% Activation of gene Cry : synthesis of mCry (mCRY)
V2max*(1+d*([Bmal-Clock_nucl]/kt2)^e)/((1+([RevErb_nucl]/ki21)^f1)*(1+([Bmal-Clock_nucl]/kt2)^e*(PC/ki2)^f+([Bmal-
Clock_nucl]/kt2)^e))
for RevErb_nucl + Cry-Per_nucl + Cry-Per~{p}::nucl + Bmal-Clock_nucl => RevErb_nucl + Cry-Per_nucl + Cry-Per~{p}::nucl +
Bmal-Clock_nucl + mCRY.
MA(dy2) for mCRY => _.
% Activation of gene Per : synthesis of mPER (mPER)
coefclock*V1max*(1+a*([Bmal-Clock_nucl]/kt1)^b)/(1+([Bmal-Clock_nucl]/kt1)^b*(PC/ki1)^c+([Bmal-Clock_nucl]/kt1)^b)
for Cry-Per_nucl + Cry-Per~{p}::nucl + Bmal-Clock_nucl => Cry-Per_nucl + Cry-Per~{p}::nucl + Bmal-Clock_nucl + mPER.
MA(coefclock*dy1) for mPER => _.
MA(coefclock*kiz8) for Bmal_cyto => Bmal_nucl.
MA(coefclock*kdx1) for Bmal-Clock_nucl => Bmal_nucl.
MA(coefclock*kfx1) for Bmal_nucl => Bmal-Clock_nucl.
MA(coefclock*dx7) for Bmal_nucl => _.
MA(coefclock*kp5) for _=[mBMAL] => Bmal_cyto.
coefclock*kp5*y50 for _ => Bmal_cyto.
MA(coefclock*dz8) for Bmal_cyto => _.
% Activation of gene Bmal : synthesis of mBmal (mBMAL)
coefclock*V5max*(1+i*([Ror_nucl]/kt5)^n)/(1+([RevErb_nucl]/ki5)^m+([Ror_nucl]/kt5)^n)
for Ror_nucl + RevErb_nucl => Ror_nucl + RevErb_nucl + mBMAL.
MA(coefclock*dy5) for mBMAL => _.
MA(coefclock*kiz7) for Ror_cyto => Ror_nucl.
MA(coefclock*dx6) for Ror_nucl => _.
MA(coefclock*kiz6) for RevErb_cyto => RevErb_nucl.
MA(coefclock*dx5) for RevErb_nucl => _.
MA(coefclock*kp4) for _=[mROR] => Ror_cyto.
coefclock*kp4*y40 for _ => Ror_cyto.
MA(coefclock*dz7) for Ror_cyto => _.
MA(coefclock*kp3) for _=[mREVERB] => RevErb_cyto.
coefclock*kp3*y30 for _ => RevErb_cyto.
MA(coefclock*dz6) for RevErb_cyto => _.
% Activation of gene Ror : synthesis of mRor (mROR)
coefclock*V4max*(1+h*([Bmal-Clock_nucl]/kt4)^p)/(1+([Bmal-Clock_nucl]/kt4)^p*(PC/ki4)^q+([Bmal-Clock_nucl]/kt4)^p)
for Cry-Per_nucl + Cry-Per~{p}::nucl + Bmal-Clock_nucl => Cry-Per_nucl + Cry-Per~{p}::nucl + Bmal-Clock_nucl + mROR.
MA(coefclock*dy4) for mROR => _.
% Activation of gene Rev : synthesis of mRev (mREVERB)
coefclock*V3max*(1+g*([Bmal-Clock_nucl]/kt3)^v)/(1+([Bmal-Clock_nucl]/kt3)^v*(PC/ki3)^w+([Bmal-Clock_nucl]/kt3)^v)
for Cry-Per_nucl + Cry-Per~{p}::nucl + Bmal-Clock_nucl => Cry-Per_nucl + Cry-Per~{p}::nucl + Bmal-Clock_nucl + mREVERB

MA(coefclock*dy3) for mREVERB => _.
% Degradation of BC_nucl (Bmal-Clock_nucl)
MA(coefclock*dx1) for Bmal-Clock_nucl => _.
```

Coupling the two models

A. APPENDIX

```
load_biocham('../QuCellCycle.bc').
add_biocham('../ReligioClock.bc').

%% Link circadian clock -> cell cycle
%parameter(ksweem,0.25).
%ksweem=max(0, ([Bmal-Clock_nucl] - 0.5)) for _=[Bmal-Clock_nucl]=>mWee1.
%parameter(vsw,2).
%parameter(vsw_0,0).
%parameter(Kaw2,1).
%parameter(nmw,1).

% Link cell clock -> circadian cycle
%parameter(nbis,4).
%parameter(J, 1).
%macro(I, J^nbis/(J^nbis+[APC]^nbis)).
parameter(I1,1).
parameter(I2,1).
parameter(I3,1).
parameter(I4,1).
parameter(I5,1).
parameter(coefsynth1,1). %mPER
parameter(coefsynth2,1). %mCRY
parameter(coefsynth3,1). %mREVERB
parameter(coefsynth4,1). %mROR
parameter(coefsynth5,0). %mBMAL
parameter(timeinhib,3).
parameter(endMitosis,0).
parameter(ti,1).
parameter(ti_0,0).
add_event([MPF]<0.5,endMitosis,Time+timeinhib).
add_event([MPF]<0.5, I1, coefsynth1).
add_event([MPF]<0.5, I2, coefsynth2).
add_event([MPF]<0.5, I3, coefsynth3).
add_event([MPF]<0.5, I4, coefsynth4).
add_event([MPF]<0.5, I5, coefsynth5).
add_event(Time>=endMitosis, I1, 1).
add_event(Time>=endMitosis, I2, 1).
add_event(Time>=endMitosis, I3, 1).
add_event(Time>=endMitosis, I4, 1).
add_event(Time>=endMitosis, I5, 1).
macro(V1max, (ti*I1 +1 -ti) * V1maxp). % new parameter for mPER synthesis
macro(V2max, (ti*I2 +1 -ti) * V2maxp). % new parameter for mCRY synthesis
macro(V3max, (ti*I3 +1 -ti) * V3maxp). % new parameter for mREVERB synthesis
macro(V4max, (ti*I4 +1 -ti) * V4maxp). % new parameter for mROR synthesis
macro(V5max, (ti*I5 +1 -ti) * V5maxp). % new parameter for mBMAL synthesis
```