Model Checking for Biological Systems: Languages, Algorithms, and Applications

Ph.D. Thesis Proposal

Qinsi Wang

March 28, 2016

Computer Science Department
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Professor Edmund M. Clarke, Carnegie Mellon University, Chair
Professor Stephen Brookes, Carnegie Mellon University
Professor Jasmin Fisher, University of Cambridge and Microsoft Research Cambridge
Professor Marta Zofia Kwiatkowska, University of Oxford
Professor Frank Pfenning, Carnegie Mellon University

Abstract

Formal methods hold great promise in promoting further discovery and innovation for complicated biological systems. Models can be tested and adapted inexpensively in-silico to provide new insights. However, development of accurate and efficient modeling methodologies and analysis techniques is still an open challenge. This thesis proposal is focused on designing appropriate modeling formalisms and efficient analyzing algorithms for various biological systems in three different thrusts:

- **Modeling Formalisms**: we have designed a multi-scale hybrid rule-based modeling formalism (MSHR) to depict intra- and intercellular dynamics using discrete and continuous variables respectively. Its hybrid characteristic inherits advantages of logic and kinetic modeling approaches.
- **Formal Analyzing Algorithms**: 1) we have developed a LTL model checking algorithm for Qualitative Networks (QNs). It considers the unique feature of QNs and combines it with over-approximation to compute decreasing sequences of reachability set, resulting in a more scalable method. 2) We have developed a formal analyzing method to handle probabilistic bounded reachability problems for two kinds of stochastic hybrid systems considering uncertainty parameters and probabilistic jumps. Compared to standard simulation-based methods, it supports non-deterministic branching, increases the coverage of simulation, and avoids the zero-crossing problem. 3) We are designing a new framework, where formal methods and machine learning techniques take joint efforts to automate the model design of biological systems. Within this framework, model checking can also be used as a (sub)model selection method. 4) We will propose a model checking technique for general stochastic hybrid systems (GSHSs) where, besides probabilistic transitions, stochastic differential equations are used to capture continuous dynamics.
- **Applications:** To check the feasibility of our modeling language and algorithms, we have constructed and studied 1) Boolean network models of the signaling network within pancreatic cancer cells, 2) QN models describing cellular interactions during skin cells' differentiation, 3) a MSHR model of the pancreatic cancer micro-environment, 4) a hybrid automaton of our light-aided bacteria-killing process, 5) extended stochastic hybrid models for atrial fibrillation, prostate cancer treatment, and our bacteria-killing process, and 6) a GSHS model depicting population changes of different species within the algae-fish-bird freshwater ecosystem considering distinct doses of estrogen injected.

Contents

Chapter 1

Introduction

As biomedical research advances into more complicated systems, there is an increasing need to model and analyze these systems to better understand them. For decades, biologists have been using diagrammatic models to describe and understand the mechanisms and dynamics behind their experimental observations. Although these models are simple to be built and understood, they can only offer a rather static picture of the corresponding biological systems, and scalability is limited. Thus, there is an increasing need to develop formalisms into more dynamic forms that can capture time-dependent processes, together with increases in the models' scale and complexity. Formal specification and analyzing methods, such as model checking techniques, hold great promise in helping further discovery and innovation for these complicated biochemical systems. Domain experts from physicians to chemical engineers can use computational modeling and analysis tools to clarify and demystify complex systems. Models can be tested and adapted inexpensively in-silico providing new insights. However, development of accurate and efficient modeling methodologies and analysis techniques are still open challenges for biochemical systems. For model analysis, simulation is the most widely used verification technique. However, in the case of complex, asynchronous systems, these techniques can cover only a limited portion of possible behaviors. A complementary verification technique is Model Checking. In this approach, the verified system is modeled as a finite state transition system, and the specifications

are expressed in a propositional temporal logic. Then, by exhaustively exploring the state space of the state transition system, it is possible to check automatically if the specifications are satisfied. The termination of model checking is guaranteed by the finiteness of the model. One of the most important features of model checking is that, when a specification is found not to hold, a counterexample (i.e., a witness of the offending behavior of the system) is produced.

In this thesis proposal, we have been focusing on designing appropriate modeling formalisms and efficient analyzing algorithms for various biological systems in three different thrusts:

- **Modeling Formalisms**: In prior work, we designed a multi-scale hybrid rule-based modeling formalism, extended from the traditional rule-based language - BioNetGen, which is able to describe the intracellular reactions and intercellular interactions simultaneously. Furthermore, to depict intracellular reactions, its hybrid characteristic asks for less information about model parameters, such as reaction rates, than traditional rule-based languages. In a nutshell, our language can describe both discrete and continuous models using a unified rule-based representation. This results in a modeling framework that combines the advantages of logic and kinetic modeling approaches.
- **Formal Analyzing Algorithms**: In completed work, we 1) developed a model checking algorithm for Qualitative Networks (QNs), a formalism for modeling signal transduction networks in biology. One of the unique features of qualitative networks, due to their lacking initial states, is that of "reducing reachability sets". Our method considers this unique features of QNs and combines it with over-approximation to compute decreasing sequences of reachability set for QN models, which results in a more scalable model checking algorithm for QNs; and 2) developed a formal analyzing method to handle probabilistic bounded reachability problems for two kinds of stochastic hybrid systems - general hybrid systems with parametric uncertainty and probabilistic hybrid automata with additional randomness. Standard approaches to reachability problems for linear hybrid systems require numerical solutions for large optimization problems, and become infeasible for systems in-

volving both nonlinear dynamics over the reals and stochasticity. Our approach combines a SMT-based model checking technique with statistical tests in a sound manner. Compared to standard simulation-based methods, it supports non-deterministic branching, increases the coverage of simulation, and avoids the zero-crossing problem. In proposed work, we will design a model checking technique for general stochastic hybrid systems (GSHSs) where, besides probabilistic transitions, stochastic differential equations are used to capture continuous dynamics. Our approach introduces a new quantifier symbol for random variables and SDE constraints for stochastic processes. It will integrate a new SDE solver, which will make use of numerical solutions to SDEs and simulation-based methods estimating distributions of hitting times for stochastic processes, into our existing nonlinear SMT solver. It will be used to analyze the probabilistic bounded reachability problems for GSHSs. Moreover, the other part of the proposed work still to be completed will be developing a new framework, where formal methods and machine learning techniques take joint efforts to automate the model construction of biological and biomedical systems. Within this framework, model checking can also be used as a (sub)model selection method.

- **Applications:** To check the feasibility of our modeling language and analysis algorithms, previously,
  - we constructed Boolean Network models for the signaling network for single pancreatic cancer cell, and formulated important system dynamics with respect to cell fate, cell cycle, and oscillating behaviors into CTL formulas. Then, we used an existing symbolic model checker NuSMV to check against these CTL properties, and confirmed experimental observations and thus validated our model.
  - we built Qualitative Network models describing the cellular interactions during the development of the skin differentiation, and applied our improved bounded LTL model checking technique. By comparing our method with an existing model checking technique for Qualitative Networks, we showed that our method offered a signif-

icant acceleration especially when analyzing large and complex models.

- we developed a multi-scale hybrid rule-based model for the pancreatic cancer microenvironment, and employed statistical model checking to analyze it. The formal analysis results showed that our model could reproduce existing experimental findings with regard to the mutual promotion between pancreatic cancer and stellate cells. The results also explained how treatments latching onto different targets resulted in distinct outcomes. We then used our model to predict possible targets for drug discovery.
- we created a nonlinear hybrid model to depict a light-aided bacteria-killing process. Then, by using a recently promoted δ-complete decision procedure-based model checking technique, we found that 1) the earlier we turn on the light after adding IPTG, the quicker bacteria cells can be killed; 2) in order to kill bacteria cells, the light has to be turned on for at least 4 time units; 3) the time difference between removing the light and removing IPTG has few impact on the cell killing outcome; and 4) the range of the necessary concentration of SOX to kill bacteria cells might be broader than the one given by our collaborating biologist, which had been confirmed then.
- we extended hybrid models for atrial fibrillation, prostate cancer treatment, and our bacteria-killing process into stochastic hybrid models. We, then, applied our probabilistic bounded reachability analyzer SReach to demonstrate its feasibility in model falsification, parameter estimation, and sensitivity analysis.
- we constructed a GSHS model to track changes in population sizes of different species within the algae-fish-bird ecosystem with distinct doses of estrogen injected. We will use it as the case study model for our proposed model checking technique for GSHSs later.

Chapter 2

Completed Work: Pancreatic Cancer Single Cell Model as Boolean Network and Symbolic Model Checking

Signal transduction is a process for cellular communication where the cell receives (and responds to) external stimuli from other cells and from the environment. It affects most of the basic cell control mechanisms such as differentiation and apoptosis. The transduction process begins with the binding of an extracellular signaling molecule to a cell-surface receptor. The signal is then propagated and amplified inside the cell through signaling cascades that involve a series of trigger reactions such as protein phosphorylation. The output of these cascades is connected to gene regulation in order to control cell function. Signal transduction pathways are able to crosstalk, forming complex signaling networks.

In this chapter, we have investigated the functionality of six signaling pathways that have been shown to be genetically mutated in 100% during the progression of pancreatic cancer [46], within a pancreatic cancer cell, and constructed a in-silico Boolean network model considering the crosstalk among them [35, 36]. In our model, we have considered three important cell functions - proliferation, apoptosis, and cell cycle arrest. Given this model, we are interested in verifying

that sequences of signal activation will drive the network to a pre-specified state within a pre-specified time. Thus, we have applied symbolic model checking (SMC) to it, and shown that its behaviors are qualitatively consistent with experiments. We have demonstrated that SMC offers a powerful approach for studying logical models of relevant biological processes.

2.1 Pancreatic Cancer Cell Model

Genomic analyses [46] have identified six cellular signaling pathways that are genetically altered in 100% of pancreatic cancers: the KRAS, Hedgehog, Wnt/Notch, Apoptosis, TGFβ, and regulation of G1/S phase transition signaling pathways. Also, many in vitro and in vivo experiments with pancreatic cancer cells have found that several growth factors and cytokines including IGF/Insulin, EGF, Hedgehog, WNT, Notch ligands, HMGB1, TGFβ, and oncoprotein including RAS, NFκB, and SMAD7 are overexpressed [6]. We performed an extensive literature search and constructed a signaling network model composed by the EGF-PI3K-P53, Insulin/IGF-KRAS-ERK, SHH-GLI, HMGB1-NFκB, RB - E2F, WNTβ - Catenin, Notch, TGFβ - SMAD, and Apoptosis pathway. Our aim is to study the interplay between tumor growth, cell cycle arrest, and apoptosis in the pancreatic cancer cell. In Figure 2.1, we depict the crosstalk model of different signaling pathways in the pancreatic cancer cell. (See [36] about the details of these pathways within our model.)

2.2 Results and Discussion

We used NuSMV [20], a Symbolic Model Checker to determine whether our in silico pancreatic cancer cell model satisfies certain properties written in a temporal logic. In our model, we set the initial values of ARF, INK4α, and SMAD4 to be OFF (0), while Cyclin D is set to be ON (1). These choices are motivated by the following observations. According to the genetic progression model of pancreatic adenocarcinoma, the malignant transformation from normal duct to pancreatic adenocarcinomas requires multiple genetic alterations in the progression of neoplastic growth, represented by Pancreatic intraepithelial neoplasia (PanINs)1A/B, PanIN-2, PanIN-3 [8]. The loss of the functions of CDKN2A, which encodes two tumor suppressors INK4A and

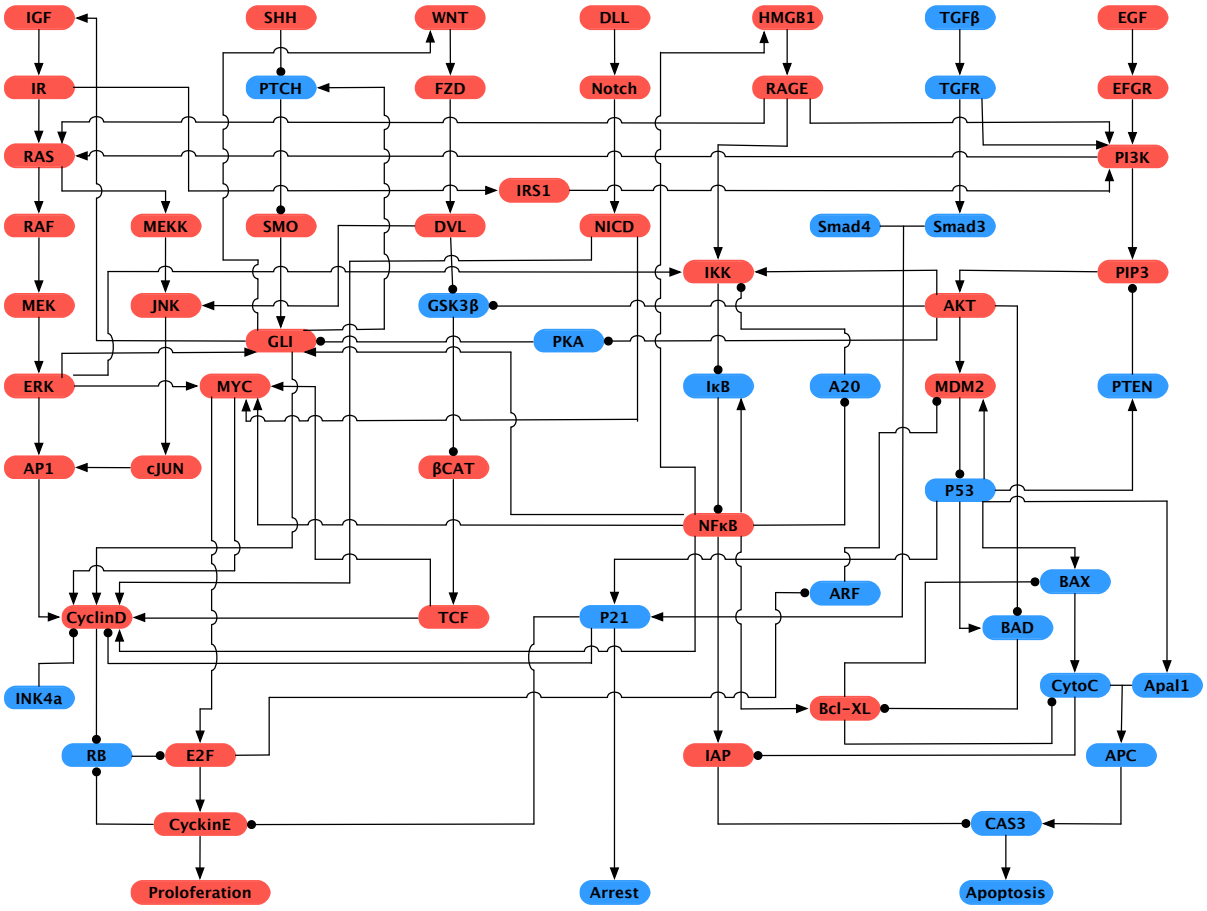

Figure 2.1: Schematic view of signal transduction in the pancreatic cancer model. Blue nodes represent tumor-suppressor proteins, red nodes represent oncoproteins/lipids. Arrow represents protein activation, circle-headed arrow represents deactivation.

ARF, occurs in 80 - 95% of sporadic pancreatic adenocarcinomas [60]. SMAD4 is a key component in the TGFβ pathway which can inhibit most normal epithelial cellular growth by blocking the G1-S phase transition in the cell cycle; and it is frequently lost or mutated in pancreatic adenocarcinoma [75]. Furthermore, it has been shown that the loss of SMAD4 can predict decreased survival in pancreatic adenocarcinoma [38]. Besides the loss of many tumor suppressors, the oncoprotein Cyclin D is frequently overexpressed in many human pancreatic endocrine tumors [19]. As shown in Table 2.1, we divide the properties that have been considered into three categories, according to their relationship with Cell Fate, Cell Cycle, and Oscillations.

property	verification result	discussion
Cell Fate		
$\mathbf{AF}\ Apoptosis \vee \mathbf{AF}\ Arrest$	False	the cell does not necessarily have to undergo apoptosis, and the cell cycle does not necessarily stop
$\mathbf{AF}\ Proliferate$	True	the cancer cell will necessarily proliferate
$\mathbf{AF}\ \mathbf{AG}\ Proliferate$	True	proliferation is eventually both unavoidable and permanent
$\mathbf{AF}\ !Apoptosis \wedge \mathbf{AF}\ !Arrest$	True	it is always possible for the cancer cell to reach states in which Apoptosis and Arrest are OFF, thereby making cell proliferation possible
$\mathbf{AF}\ (!Apoptosis \wedge !Arrest \wedge Proliferate)$	False	the model cannot always eventually reach a state in which apoptosis and cell cycle arrest are not inhibited and cell proliferation is active
$\mathbf{AF}\ \mathbf{AG}\ !Apoptosis \vee \mathbf{AF}\ \mathbf{AG}\ !Arrest$	False	inhibition of apoptosis and cell cycle arrest are not unavoidable and permanent
Cell Cycle		
$\mathbf{A}\ (!Proliferate\ \mathbf{U}\ CyclinD)$	True	it is always the case that cell proliferation does not occur until Cyclin D is expressed (or activated)
$\mathbf{AF}\ \mathbf{AG}\ CyclinD$	False	in our model the activation of Cyclin D is not a steady state
$!\mathbf{E}\ (!P53\ \mathbf{U}\ Apoptosis)$	False	apoptosis can be activated even when P53 is not
Oscillations		
$TGF\beta \rightarrow \mathbf{AG}\ ((!NF\kappa B \rightarrow \mathbf{AF}\ NF\kappa B) \wedge (NF\kappa B \rightarrow \mathbf{AF}\ !NF\kappa B)$	True	an initial overexpression of TGFβ always leads to oscillations in NFκB's expression level
$PIP3 \rightarrow \mathbf{AG}\ ((!NF\kappa B \rightarrow \mathbf{AF}\ NF\kappa B) \wedge (NF\kappa B \rightarrow \mathbf{AF}\ !NF\kappa B))$	True	PIP3 has the similar impact on NFκB's expression level
$\mathbf{AG}\ ((P53 \rightarrow \mathbf{AF}\ MDM2) \wedge (MDM2 \rightarrow \mathbf{AF}\ !P53))$	True	overexpression of P53 will always activate MDM2, which will in turn inhibit P53

Table 2.1: Model checking results.

Chapter 3

Completed Work: Biological Signaling Networks as Qualitative Networks and Improved Bounded Model Checking

One successful approach to the usage of abstraction in biology has been the usage of Boolean networks [69]. Boolean networks call for abstracting the status of each modeled substance as either active (on) or inactive (off). Although a very high level abstraction, it has been found useful to gain better understanding of certain biological systems [61, 64]. The appeal of this discrete approach along with the shortcomings of the very aggressive abstraction, led researchers to suggest various formalisms such as Qualitative Networks [62] and Gene Regulatory Networks [57] that allow to refine models when compared to the Boolean approach. In these formalisms, every substance can have one of a small discrete number of levels. Dependencies between substances become algebraic functions instead of Boolean functions. Dynamically, a state of the model corresponds to a valuation of each of the substances and changes in values of substances occur gradually based on these algebraic functions. Qualitative networks and similar formalisms (e.g., genetic regulatory networks [[69]) have proven to be a suitable formalism to model some biological systems [12, 61, 62, 69].

Here, we consider model checking of qualitative networks. One of the unique features of qualitative networks is that they have no initial states. That is, the set of initial states is the set of all states. Obviously, when searching for specific executions or when trying to prove a certain property we may want to restrict attention to certain initial states. However, the general lack of initial states suggests a unique approach towards model checking. It follows that if a state that is not visited after i steps will not be visited after i' steps for every $i' > i$. These "decreasing" sets of reachable states allow to create a more efficient symbolic representation of all the paths of a certain length. However, this observation alone is not enough to create an efficient model checking procedure. Indeed, accurately representing the set of reachable states at a certain time amounts to the original problem of model checking (for reachability), which does not scale. In order to address this we use an over-approximation of the set of states that are reachable by exactly n steps. We represent the over-approximation as a Cartesian product of the set of values that are reachable for each variable at every time point. The computation of this over-approximation never requires us to consider more than two adjacent states of the system. Thus, it can be computed quite efficiently. Then, using this over-approximation we create a much smaller encoding of the set of possible paths in the system. We test our method on many of the biological models developed using Qualitative Networks. The experimental results show that there is significant acceleration when considering the decreasing reachability property of qualitative networks. In many examples, in particular larger and more complicated biological models, this technique leads to considerable speedups. The technique scales well with increase of size of models and with increase in length of paths sought for.

3.1 Decreasing Reachability Sets

A notable difference between QNs and "normal" transition systems is that QNs do not specify initial states. For example, for the classical stability analysis all states are considered as initial states. It follows that if a state s of a QN is not reachable after i steps, it is not reachable after

i' steps for every $i' > i$. Thus, there is a decreasing sequence of sets $\Sigma_0 \supseteq \Sigma_1 \supseteq \cdots \supseteq \Sigma_l$ such that searching for runs of the network can be restricted to the set of runs of the form $\Sigma_0,\ \Sigma_1,\ \cdots,\ (\Sigma_l)^\omega$. Here we show how to take advantage of this fact in constructing a more scalable model checking algorithm for qualitative networks.

Consider a Qualitative Network $Q(V, T, N)$ with set of states $\Sigma : V \to \{0, \cdots, N\}$. We say that a state $s \in \Sigma$ is reachable by exactly i steps if there is some run $r = s_0, s_1, \cdots$ such that $s = s_i$. Dually, we say that s is not reachable by exactly i steps if for every run $r = s_0, s_1, \cdots$ we have $s_i \neq s$.

Lemma 1. If a state s is not reachable by exactly i steps then it is not reachable by exactly i' steps for every $i' > i$.

The algorithm 1 computes a decreasing sequence $\Sigma_0 \supset \Sigma_1 \supset \cdots \supset \Sigma_{j-1}$ such that all states that are reachable by exactly i steps are in Σ_i if $i < j$ and in Σ_{j-1} if $i \geq j$. We note that the definition of Σ_{j+1} in line 5 is equivalent to the standard $\Sigma_{j+1} = f(\Sigma_j)$, where function $f(\cdot)$ is used to compute the next reachable set. However, we choose to write it as in the algorithm below in order to stress that only states in Σ_j are candidates for inclusion in Σ_{j+1}. Given the sets $\Sigma_0, \cdots, \Sigma_{j-1}$, every run $r = s_0, s_1, \cdots$ of Q satisfies $s_i \in \Sigma_i$ for $i < j$ and $s_i \in \Sigma_{j-1}$ for $i \geq j$. In particular, if $Q \nvDash \varphi$ for some LTL formula φ, then the run witnessing the unsatisfaction of φ can be searched for in this smaller space of runs. Unfortunately, the algorithm 1 is not feasible. Indeed, it amounts to computing the exact reachability sets of the QN Q, which does not scale well [23].

Algorithm 1 Concrete Decreasing Reachability

1: $\Sigma_0 = \Sigma$;
2: $\Sigma_{-1} = \emptyset$;
3: $j = 0$;
4: **while** $\Sigma_{j-1} \neq \Sigma_j$ **do**
5: $\quad \Sigma_{j+1} = \Sigma_j \setminus \{s' \in \Sigma | \forall s \in \Sigma \cdot s' \neq f(s)\}$;
6: $\quad j++$;
7: **end while**
8: **return** $\Sigma_0, \cdots, \Sigma_{j-1}$

In order to effectively use Lemma 1 we combine it with over-approximation, which leads to a scalable algorithm. Specifically, instead of considering the set Σ_k of states reachable at step k, we identify for every variable $v_i \in V$ the domain $D_{i,k}$ of the set of values possible at time k for variable v_i. Just like the general set of states, when we consider the possible values of variable v_i we get that $D_{i,0} \supseteq D_{i,1} \supseteq \cdots \supseteq D_{i,l}$. The advantage is that the sets $D_{i,k}$ for all $v_i \in V$ and $k > 0$ can be constructed by induction by considering only the knowledge on previous ranges and the target function of one variable.

Consider the algorithm 2. For each variable, it initializes the set of possible values at time 0 as the set of all values. Then, based on the possible values at time j, it computes the possible values at time $j + 1$. The actual check can be either implemented explicitly if the number of inputs of all target functions is small (as in most cases) or symbolically (see [21]). Considering only variables (and values) that are required to decide the possible values of variable v_i at time j makes the problem much simpler than the general reachability problem. Notice that, again, only values that are possible at time j need be considered at time $j+1$. That is, $D_{i,j+1}$ starts as empty (line 6) and only values from $D_{i,j}$ are added to it (lines 7 - 10). As before, $D_{i,j+1}$ is the projection of $f(D_{1,j} \times \cdots \times D_{m,j})$ on v_i. The notation used in the algorithm above stresses that only states in $D_{i,j}$ are candidates for inclusion in $D_{i,j+1}$.

The algorithm produces very compact information that enables to follow with a search for runs of the QN. Namely, for every variable v_i and for every time point $0 \leq k < j$ we have a decreasing sequence of domains

$$D_{i,0} \supseteq D_{i,1} \supseteq \cdots \supseteq D_{i,k}.$$

Consider a Qualitative Network $Q(V, T, N)$, where $V = \{v_1, \cdots, v_n\}$ and a run $r = s_0, s_1, \cdots$. As before, every run $r = s_0, s_1, \cdots$ satisfies that for every i and for every t we have $s_t(v_i) \in D_{i,t}$ for $t < j$ and $s_t(v_i) \in D_{i,j-1}$ for $t \geq j$.

We look for paths that are in the form of a lasso, as we explain below. We say that r is *a*

Algorithm 2 Abstract Decreasing Reachability

$\forall v_i \in V \cdot D_{i,0} = \{0, 1, \cdots, N\}$;
$\forall v_i \in V \cdot D_{i,-1} = \emptyset$;
$j = 0$;
while $\exists v_i \in V \cdot D_{i,j} \neq D_{i,j-1}$ **do**
    for each $v_i \in V$ **do**
        $D_{i,j+1} = \emptyset$;
        for each $d \in D_{i,j}$ **do**
            if $\exists (d_1, \cdots, d_m) \in D_{1,j} \times \cdots \times D_{m,j} \cdot f_v(d_1, \cdots, d_m) = d$ **then**
                $D_{i,j+1} = D_{i,j+1} \cup \{d\}$;
            end if
        end for
    end for
end while
$j++$;
return $\forall v_i \in V, \forall j' \leq j \cdot D_{i,j'}$

loop of length l if for some $0 < k \leq l$ and for all $m \geq 0$ we have $s_{l+m} = s_{l+m-k}$. That is, the run r is obtained by considering a prefix of length $l - k$ of states and then a loop of k states that repeats forever. A search for a loop of length l that satisfies an LTL formula φ can be encoded as a bounded model checking query as follows. We encode the existence of l states $s_0, \cdots, s_{l-1}$. We use the decreasing reachability sets $D_{i,t}$ to force state s_t to be in $D_{0,t} \times \cdots \times D_{n,t}$. This leads to a smaller encoding of the states $s_0, \cdots, s_{l-1}$ and to smaller search space. We add constraints that enforce that for every $0 \leq t < l - 1$ we have $s_{t+1} = f(s_t)$. Furthermore, we encode the existence of a time $l - k$ such that $s_{l-k} = f(s_{l-1})$. We then search for a loop of length l that satisfies φ . It is well known that if there is a run of Q that satisfies φ then there is some l and a loop of length l that satisfies φ . We note that sometimes there is a mismatch between the length of loop sought for and length of sequence of sets (j) produced by the algorithm 2. Suppose that the algorithm returns the sets $D_{i,t}$ for $v_i \in V$ and $0 \leq t < j$. If $l > j$, we use the sets $D_{i,j-1}$ to "pad" the sequence. Thus, states $s_j, \cdots, s_{l-1}$ will also be sought in $\prod_i D_{i,j-1}$. If $l < j$, we use the sets $D_{i,0}, \cdots, D_{i,l-2}, D_{i,j-1}$ for $v_i \in V$. Thus, only the last state s_{l-1} is ensured to be in our "best" approximation $\prod_i D_{i,j-1}$. A detailed explanation of how we encode the decreasing reachability sets as a Boolean satisfiability problem is given in [21].

3.2 Results for Various Biological Models

We implemented this technique to work on models defined through our tool BMA [9]. Here, we present experimental results of running our implementation on a set of different biological models, including a total of 22 benchmark problems from various sources (skin cells differentiation models by ourselves, diabetes models from [12], models of cell fate determination during *C. elegans* vulval development, a Drosophila embryo development model from [61], Leukemia models constructed by ourselves, and a few additional examples constructed by ourselves). The number of variables in the models and the maximal range of variables is reported in Table 3.1.

Model name	#Vars	Range	Model name	#Vars	Range
2var_unstable	2	0..1	Bcr-Abl	57	0..2
Bcr-AblNoFeedbacks	54	0..2	BooleanLoop	2	0..1
NoLoopFound	5	0..4	Skin1D_TF_0	75	0..4
Skin1D_TF_1	75	0..4	Skin1D	75	0..4
Skin2D_3X2_0	90	0..4	Skin2D_3X2_1	90	0..4
Skin2D_3X2_2	90	0..4	Skin2D_5X2_TF	198	0..4
Skin2D_5X2	198	0..4	SmallTestCase	3	0..4
SSkin1D_TF_0	30	0..4	SSkin1D_TF_1	31	0..4
SSkin1D	30	0..4	SSkin2D_3X2	40	0..4
VerySmallTest	2	0..4	VPC_lin15ko	85	0..2
VPC_Non_stable	33	0..2	VPC_stable	43	0..2

Table 3.1: Number of variables in models and their ranges.

Our experiments compare two encodings. One encoding is explained in algorithm 2, referred to as "opt" (for optimized). the other considers l states $s_0, \cdots, s_l$ where $s_t(v_i) \in \{0, \cdots, N\}$ for every t and every i. That is, for every variable v_i and every time point $0 \leq t \leq l$ we consider the set $D_{i,t} = 0, \cdots, N$. This encoding is referred to as "naïve". In both cases we use the same encoding to a Boolean satisfiability problem. Further details about the exact encoding can be found in [21].

We perform two kinds of experiments. First, we search for loops of length 10, 20, $\cdots$, 50 on all the models for the optimized and naïve encodings. Second, we search for loops that satisfy a certain LTL property (either as a counterexample to model checking or as an example

run satisfying a given property). Again, this is performed for both the optimized and the naïve encodings. LTL properties are considered only for four biological models. The properties were suggested by our collaborators as interesting properties to check for these models. For both experiments, we report separately on the global time and the time spent in the SAT solver. All experiments were run on an Intel Xeon machine with CPU X7560@2.27GHz running Windows Server 2008 R2 Enterprise.

In Tables 3.2 and 3.3 we include experimental results for the search for loops. We compare the global run time of the optimized search vs the naïve search. The global run time for the optimized search includes the time it takes to compute the sequence of decreasing reachability sets. Accordingly, in some of the models, especially the smaller ones, the overhead of computing this additional information makes the optimized computation slower than the naïve one. For information we include also the net runtime spent in the SAT solver.

In Table 3.4 we include experimental results for the model checking experiment. As before, we include the results of running the search for counterexamples of lengths 10, 20, 30, 40, and 50. We include the total runtime of the optimized vs the naïve approaches as well as the time spent in the SAT solver. As before, the global runtime for the optimized search includes the computation of the decreasing reachability sets. The properties in the table are of the following form. Let I, $a \cdots d$ denote formulas that are Boolean combinations of propositions.

- $I \rightarrow (\neg a)\ \mathbf{U}\ b$: we check that the sequence of events when starting from the given initial states (I) satisfies the order that b happens before a.
- $I \wedge \mathbf{FG}\ a \wedge \mathbf{F}\ (b \wedge \mathbf{XF}\ c)$: we check that the model gets from some states (I) to a loop that satisfies the condition a and the path leading to the loop satisfies that b happens first and then c.
- $I \wedge \mathbf{FG}\ a \wedge \mathbf{F}\ (b \wedge \mathbf{XF}\ (c \wedge \mathbf{XF}\ d))$: we extend the previous property by checking the sequence a then b then c and then d.
- $I \wedge \mathbf{FG}\ a \wedge (\neg b)\ \mathbf{U}\ c$: we check that the model gets from some states (I) to a loop

Length of loop	10				20				30			
	Global Time (s)		Sat Time (s)		Global Time (s)		Sat Time (s)		Global Time (s)		Sat Time (s)	
Model name	Naïve	Opt	Naïve	Opt	Naïve	Opt	Naïve	Opt	Naïve	Opt	Naïve	Opt
2var_unstable	6.92	0.78	0.21	0	0.46	0.54	0	0	0.51	0.57	0	0
Bcr-Abl	67.76	9.32	28.92	1.46	196.68	9.49	142.41	1.31	281.27	10.29	108.14	1.85
Bcr-AblNoFeedbacks	66.52	6.77	29.58	0.71	201.59	6.71	101.69	0.56	307.60	6.60	219.72	0.62
BooleanLoop	0.49	0.51	0	0	0.48	0.57	0.01	0	0.53	0.59	0.01	0.01
NoLoopFound	0.78	0.74	0.06	0.01	1.14	0.93	0.09	0.03	1.45	1.04	0.10	0.06
Skin1D_TF_0	136.21	140.78	122.85	127.47	218.52	80.33	191.06	55.23	127.28	96.49	86.05	60.06
Skin1D_TF_1	167.32	173.03	154.00	159.55	698.47	445.32	670.77	419.24	883.35	572.03	842.06	536.04
Skin1D	90.92	68.82	77.63	54.54	45.67	23.21	17.55	8.77	133.72	23.46	92.36	8.13
Skin2D_3X2_0	567.31	640.71	545.49	618.44	238.28	205.15	192.28	162.14	164.79	218.77	93.45	153.11
Skin2D_3X2_1	910.08	553.27	891.70	535.02	82.04	117.48	44.70	82.79	122.77	219.04	64.96	167.65
Skin2D_3X2_2	315.20	169.92	293.45	151.64	121.12	36.58	74.49	18.74	188.78	39.36	114.81	20.15
Skin2D_5X2_TF	511.31	223.93	459.38	182.65	1466.90	391.96	1378.80	353.06	1275.30	73.77	1135.25	35.83
Skin2D_5X2	343.96	85.64	300.03	56.71	721.58	57.20	630.92	28.46	965.24	48.26	828.12	16.83
SmallTestCase	0.53	0.54	0.01	0	0.54	0.73	0.01	0	0.60	0.54	0.01	0
SSkin1D_TF_0	70.71	69.00	63.71	61.93	21.35	20.71	5.87	5.93	33.07	32.74	12.52	12.34
SSkin1D_TF_1	9.77	10.05	2.88	2.93	22.85	26.02	8.23	9.04	35.61	35.16	15.12	14.96
SSkin1D	145.28	146.74	138.61	139.76	32.00	33.38	18.29	18.51	33.89	33.80	13.57	13.49
SSkin2D_3X2	301.33	158.62	286.80	148.08	63.46	50.12	35.44	36.14	86.26	32.41	44.30	14.91
VerySmallTest	0.37	0.42	0	0	0.39	0.43	0.01	0	0.40	0.43	0.01	9
VPC_lin15ko	8.31	6.81	3.35	0.32	14.87	6.74	5.13	0.26	21.99	6.76	7.42	0.20
VPC_Non_stable	3.43	3.40	0.85	0.26	6.02	3.95	1.23	0.29	9.35	4.87	2.10	0.62
VPC_stable	3.31	4.79	0.74	0.14	5.84	4.79	0.99	0.18	9.10	4.67	1.92	0.14

Table 3.2: Searching for loops (10, 20, 30).

Length of loop	40				50			
	Global Time (s)		Sat Time (s)		Global Time (s)		Sat Time (s)	
Model name	Naïve	Opt	Naïve	Opt	Naïve	Opt	Naïve	Opt
2var_unstable	0.54	0.60	0.01	0	1.05	0.64	0.01	0.01
Bcr-Abl	667.22	11.54	552.90	2.74	1019.68	11.94	869.56	2.76
Bcr-AblNoFeedbacks	574.61	6.79	316.07	0.64	857.17	6.90	719.21	0.69
BooleanLoop	0.54	0.60	0.01	0.01	0.59	0.66	0.01	0.01
NoLoopFound	1.90	1.15	0.23	0.04	2.23	1.34	0.22	0.05
Skin1D_TF_0	126.13	153.85	68.31	104.11	224.38	247.93	149.55	182.58
Skin1D_TF_1	108.84	160.72	52.01	112.33	167.86	290.97	91.13	228.46
Skin1D	122.73	29.39	64.99	12.84	259.75	34.04	182.50	16.09
Skin2D_3X2_0	391.08	325.43	293.83	237.89	470.89	663.87	341.24	545.49
Skin2D_3X2_1	196.99	271.98	118.22	202.01	476.94	557.09	366.61	464.88
Skin2D_3X2_2	413.13	44.06	314.95	23.75	445.78	47.51	308.71	25.18
Skin2D_5X2_TF	3067.08	93.15	2649.01	48.12	5135.87	82.38	3956.13	34.25
Skin2D_5X2	2403.53	47.69	2149.43	14.87	4025.83	56.86	3254.90	18.18
SmallTestCase	0.96	0.57	0.02	0	0.77	0.58	0.02	0
SSkin1D_TF_0	44.81	42.03	13.52	13.37	58.09	57.45	22.64	21.91
SSkin1D_TF_1	43.97	46.26	15.88	16.13	60.46	60.52	22.77	24.35
SSkin1D	41.13	41.49	12.48	12.59	60.77	61.34	22.82	22.87
SSkin2D_3X2	117.64	42.86	50.82	20.36	157.07	51.19	80.54	22.95
VerySmallTestCase	0.48	0.44	0	0	0.81	0.67	0.01	0
VPC_lin15ko	27.04	6.94	7.34	0.20	45.70	7.14	20.78	0.23
VPC_Non_stable	14.58	5.64	2.36	0.65	16.21	6.50	4.10	1.07
VPC_stable	13.13	6.66	3.44	0.12	17.07	4.99	5.07	0.20

Table 3.3: Searching for loops (40, 50).

that satisfies the condition a and the path leading to the loop satisfies that b cannot happen before c.

- $\mathbf{GF}\ a\ \wedge\ \mathbf{GF}\ b$: we check for the existence of loops that exhibit a form of instability by having states that satisfy both a and b.

When considering the path search, on many of the smaller models the new technique does not offer a significant advantage. However, on larger models, and in particular the two dimensional skin model (Skin2D_5X2 from [62]) and the Leukemia model (Bcr_Abl) the new technique is an order of magnitude faster. Furthermore, when increasing the length of the path it scales a lot better than the naïve approach. When model checking is considered, the combination of the decreasing reachability sets accelerates model checking considerably. While the naïve search increases considerably to the order of tens of minutes, the optimized search remains within the order of 10s, which affords a "real-time" response to users.

Model name	Global Time (s)		Sat Time (s)		Ratio		
	Naïve	Opt	Naïve	Opt	Global	Sat	
Bcr-Abl1	69.30	9.04	26.67	0.90	7.66	29.61	sat
Bcr-Abl1	188.13	12.21	87.70	1.42	15.40	61.47	sat
Bcr-Abl1	380.24	13.12	292.21	2.01	28.96	145.02	sat
Bcr-Abl1	648.02	12.37	349.70	2.30	52.38	151.87	sat
Bcr-Abl1	1005.37	11.52	588.34	2.17	87.19	270.93	sat
Bcr-Abl2	47.04	10.97	9.94	0.72	4.28	13.76	Unsat
Bcr-Abl2	136.48	8.62	41.04	0.75	15.82	54.66	Unsat
Bcr-Abl2	285.28	11.28	112.35	0.77	25.28	144.58	Unsat
Bcr-Abl2	561.65	9.29	443.91	0.80	60.41	553.83	Unsat
Bcr-Abl2	781.64	12.03	408.55	0.87	64.96	465.55	Unsat
Bcr-Abl3	48.64	8.47	9.54	0.83	5.74	11.45	Unsat
Bcr-Abl3	133.83	9.10	38.68	1.11	14.69	34.81	Unsat
Bcr-Abl3	283.73	9.45	106.61	1.16	30.01	91.28	Unsat
Bcr-Abl3	596.50	9.50	466.01	1.18	62.78	394.48	Unsat
Bcr-Abl3	853.53	10.05	480.77	1.36	84.89	351.99	Unsat
Bcr-Abl4	75.27	9.19	44.50	0.80	8.18	55.31	sat
Bcr-Abl4	202.06	9.95	143.49	1.53	20.30	93.50	sat
Bcr-Abl4	296.02	11.35	116.24	2.54	26.07	45.75	sat
Bcr-Abl4	740.39	11.00	116.24	2.54	26.07	45.74	sat
Bcr-Abl4	975.97	10.42	823.53	1.10	93.63	747.14	sat
Bcr-AblNoFeedbacks1	42.98	6.25	7.94	0.40	6.87	19.51	Unsat
Bcr-AblNoFeedbacks1	163.33	8.18	95.43	0.77	19.95	123.90	Unsat
Bcr-AblNoFeedbacks1	302.17	6.41	122.25	0.46	47.07	260.90	Unsat
Bcr-AblNoFeedbacks1	493.28	6.41	314.24	0.45	76.92	686.28	Unsat
Bcr-AblNoFeedbacks1	809.97	6.45	680.70	0.46	125.51	1461.69	Unsat
Bcr-AblNoFeedbacks2	44.88	6.39	6.59	0.40	7.01	16.27	Unsat
Bcr-AblNoFeedbacks2	117.96	6.34	20.98	0.39	18.58	53.61	Unsat
Bcr-AblNoFeedbacks2	312.73	7.59	231.87	0.46	41.18	500.00	Unsat
Bcr-AblNoFeedbacks2	527.40	6.31	423.61	0.39	83.46	1084.74	Unsat
Bcr-AblNoFeedbacks2	751.45	6.83	362.09	0.44	109.87	806.35	Unsat
Bcr-AblNoFeedbacks3	60.99	6.95	20.45	0.64	8.77	31.64	sat
Bcr-AblNoFeedbacks3	204.66	7.06	144.58	0.61	28.97	233.95	sat
Bcr-AblNoFeedbacks3	356.33	8.81	267.48	0.49	40.42	539.32	sat
Bcr-AblNoFeedbacks3	Time out	7.06	Time out	0.42	N/A	N/A	sat
VPC_non_stable1	30.14	10.83	4.83	0.69	2.78	6.93	Unsat
VPC_non_stable2	17.42	9.85	3.59	1.11	1.76	3.24	sat
VPC_non_stable3	52.01	11.91	26.69	1.48	4.36	17.93	Unsat
VPC_non_stable4	19.53	8.31	7.08	0.60	2.34	11.77	Unsat
VPC_stable1	3.75	5.11	0.31	0.07	0.73	3.99	Unsat
VPC_stable2	5.53	5.32	0.86	0.11	1.04	7.41	sat

Table 3.4: Model checking results.

Chapter 4

Completed Work: Phage-based Bacteria Killing as A Nonlinear Hybrid Automaton and δ-complete Decision-based Bounded Model Checking

Due to the widespread misuse and overuse of antibiotics, drug resistant bacteria now pose significant risks to health, agriculture and the environment. Therefore, we were interested in an alternative to conventional antibiotics, a phage therapy. Phages, or bacteriophages, are viruses that infect bacteria and have evolved to manipulate the bacterial cells and genome, making resistance to bacteriophages difficult to achieve. However, many phages are temperate, meaning that they can enter a lysogenic phase and therefore not lyse and kill the host bacteria. The addition of a phototoxic protein - KillerRed [59] - to the system offers a second method of killing those bacteria targeted by a lysogenic phage. In this chapter, we constructed a hybrid model of a bacteria killing procedure that mimics the stages through which bacteria change when phage therapy is adopted. Our model was designed according to an experimental procedure to engineer a temperate phage, Lambda (λ), and then kill bacteria via light-activated production of superoxide. We

applied δ-complete decision based bounded model checking [33] to our model and the results show that such an approach can speed up evaluation of the system, which would be impractical or possibly not even feasible to study in a wet lab.

4.1 The KillerRed Model

We have modeled synthesis and action of KillerRed that occurs over three main phases of a typical photobleaching experiment: induction at 37°C, storage at 4°C to allow for protein maturation, and photobleaching at room temperature. Within these phases, we identify several stages of interest in KillerRed synthesis and activity as follows.

- mRNA synthesis and degradation
- KillerRed synthesis, maturation, and degradation
- KillerRed states: singlet (S), singlet excited (S^*), triplet excited (T^*), and deactivated (Da)
- Superoxide production (by KillerRed)
- Superoxide elimination (by superoxide dismutase)

We implemented these system stages with distinct model states, and outlined them in Figure 4.1, together with state variables (values are included if variables are fixed within a state), transitions between states, and events that trigger state transitions. In Table 4.1 we list the model states that are used to describe the stages of the system. (See [74] for the details about equations that we derived for each stage and choices of system parameters.)

4.2 Results and Discussion

Effect of delay in turning light ON

First, we have studied the relation between the time to turn ON the light after adding IPTG that is a molecular biology reagent used to induce protein expression ($t_{lightON}$), and the total time needed until the bacteria cells being killed (t_{total}). We fixed the values of several other parameters as follows.

- SOX_{thres} = 5e-4m - threshold for the concentration level of SOX which is sufficient to kill the

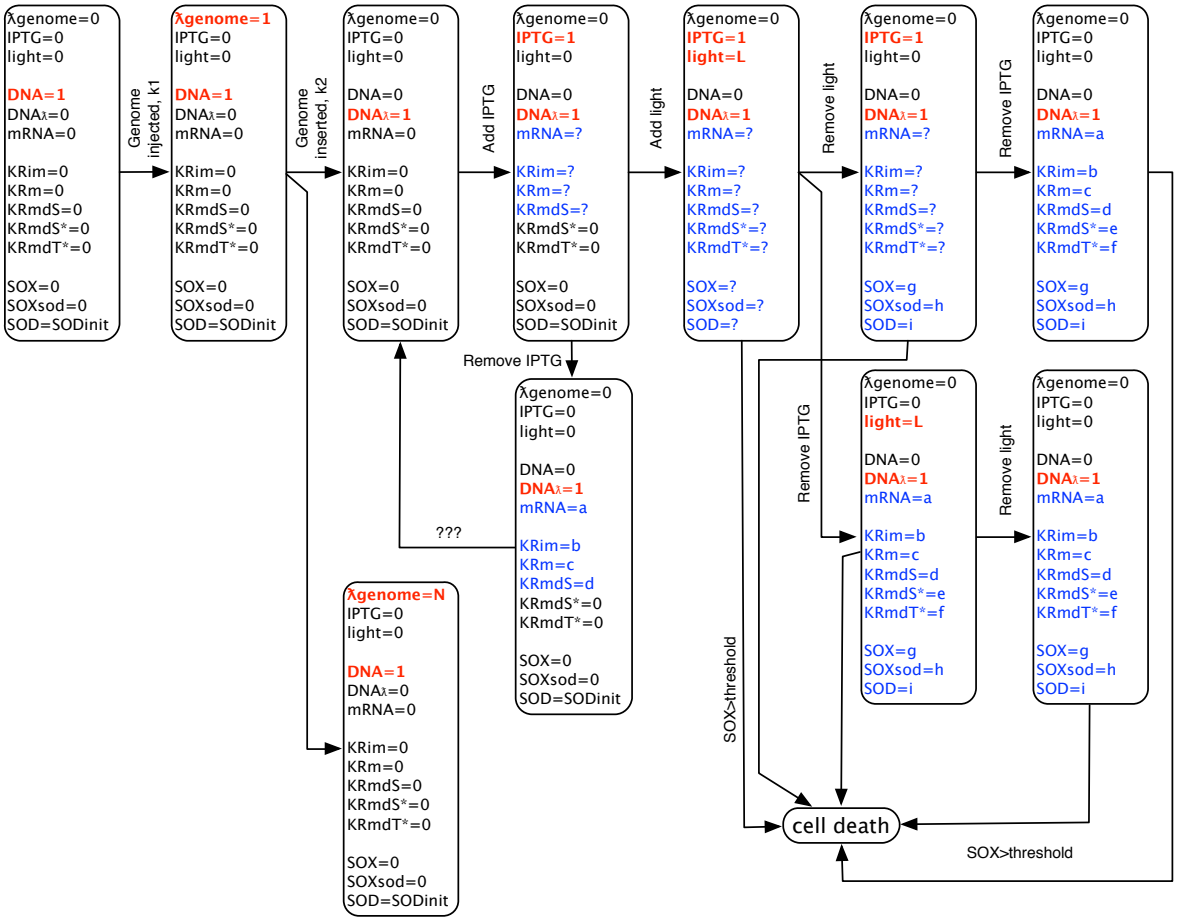

Figure 4.1: Hybrid automaton for our KillerRed model

bacteria cells

- $t_{lightOFF_1}$ = 2 hours (hrs) - time to turn the light OFF after turning it ON

- $t_{lightOFF_2}$ = 2 hrs - time to turn the light OFF after removing IPTG

- t_1 = 1 hr - time to inject genome

- t_2 = 1 hr - time to insert genome into DNA after injecting it into bacteria cell

- $t_{addIPTG_3}$ = 1 hr - time to add IPTG after inserting phage genome into bacteria DNA

As shown in the first two rows of Table 4.2, the earlier we turn on the light after adding IPTG, the quicker the bacteria cells will be killed.

Lower bound for the duration of exposure to light

The δ-decisions technique has also been adopted to analyze the impact of the time duration

State	State description	Input	Next state(s)
S_0	Initial system state, bacteria cell, without phage	n/a	S_1 (ex.)
S_1	Phage genome injected	λ-phage genome	S_2 (in.), S_3 (in.)
S_2	Phage genome replication (lytic cycle)	Genome replication	n/a
S_3	Phage genome within bacterial DNA (lysogenic cycle)	Genome insertion	S_4 (ex.)
S_4	Gene transcription, translation	Addition of IPTG	S_5 (ex.), S_6 (ex.)
S_5	Gene transcription decrease	Removal of IPTG	S_3 (in.)
S_6	Activation of KillerRed	Light turned ON	S_7 (ex.), S_8 (ex.), S_{11} (in.)
S_7	Mixture of KillerRed forms, no activation	Light turned OFF	S_9 (ex.), S_{11} (in.)
S_8	Mixture of KillerRed forms, transcription decrease	Removal of IPTG	S_{10} (ex.), S_{11} (in.)
S_9	Mixture of KillerRed forms, no activation, transcription decrease	Removal of IPTG	S_{11} (in.)
S_{10}	Mixture of KillerRed forms, transcription decrease, no activation	Light turned OFF	S_{11} (in.)
S_{11}	Cell death	SOX>threshold	n/a

Table 4.1: List of modeled system states, their description, inputs and next state(s) with indication whether transition was triggered by external input (ex.) or by internal variable (in.) reaching some specified value.

that the cells are exposed to light ($t_{lightOFF_1}$) on the system, and estimate an appropriate range for $t_{lightOFF_1}$ which leads to the successful killing of bacteria cells by KillerRed. By setting SOX_{thres}, $t_{lightOFF_2}$, t_1, t_2, and $t_{addIPTG_3}$ with the same values in Section 4.2, and assigning 2 hr to $t_{lightON}$ (time to turn the light OFF after turning it ON), we have found that, in order to kill bacteria cells, the system has to keep the light ON for at least 4 hours (see row 3-4 of Table 4.2).In addition, we have also found that the bacteria cells can be killed within 100 hours when light is ON for 4 hours.

Time to remove IPTG as an insensitive role

The sensitivity of the time difference between removing the light and removing IPTG (t_{rmIPTG_3}) with regard to the successful killing of bacteria cells has also been studied. We have noticed that

$t_{lightON}$ (hr)	1	2	3	4	5	6	7	8	9	10
t_{total} (hr)	16	17.2	18.5	20	21.3	22.7	23.5	24.1	25	30
$t_{lightOFF_1}$ (hr)	1	2	3	4	5	6	7	8	9	10
killed bacteria cells	failed	failed	failed	succ	succ	succ	succ	succ	succ	succ
t_{rmIPTG_3} (hr)	1	2	3	4	5	6	7	8	9	10
killed bacteria cells	succ	succ	succ	succ	succ	succ	succ	succ	succ	succ
SOX_{thres} (M)	1e-4	2e-4	3e-4	4e-4	5e-4	6e-4	7e-4	8e-4	9e-4	1e-3
t_{total} (hr)	5.1	5.2	5.4	17	19	48	61	71	36	42

Table 4.2: Formal analysis results for our KillerRed hybrid model

t_{rmIPTG_3} has insignificant impacts on the cell killing outcome (see row 5-6 of Table 4.2). This is in accordance with our understanding of this system, since any additional KillerRed that will be synthesized will not be activated in the absence of light. Note that, for other involved system parameters, we used the same values for SOX_{thres}, $t_{lightON}$, $t_{lightOFF_2}$, t_1, t_2, and $t_{addIPTG_3}$ as in Section 4.2, and set $t_{lightOFF_1}$ as 4 hours.

Necessary level of superoxide

Finally, we have used the δ-decisions to discuss the correctness of our hybrid model by considering various values of SOX_{thres} within the suggested range - [100uM, 1mM]. We have used the same values for variables SOX_{thres}, $t_{lightON}$, $t_{lightOFF_1}$, $t_{lightOFF_2}$, t_1, t_2, and $t_{addIPTG_3}$ as in Section 4.2. As we can see from row 7-8 of Table 4.2, the bacteria cells can be killed in reasonable time for all 10 point values of SOX_{thres}, which was uniformly chosen from [100uM, 1mM]. Furthermore, we have also found a broader range for SOX_{thres} up to 0.6667M, with which bacteria cells can be killed by KillerRed.

Chapter 5

Completed Work: Biological Systems as Stochastic Hybrid Models and *SReach*

Stochastic hybrid systems (SHSs) are dynamical systems exhibiting discrete, continuous, and stochastic dynamics. Due to the generality, they have been widely used in various areas, including biological systems, financial decision problems, and cyber-physical systems [15, 22]. One elementary question for the quantitative analysis of SHSs is the probabilistic reachability problem, considering that many verification problems can be reduced to reachability problems. It is to compute the probability of reaching a certain set of states. The set may represent certain unsafe states which should be avoided or visited only with some small probability, or dually, good states which should be visited frequently. This problem is no longer a decision problem, as it generalizes that by asking what is the probability that the system reaches the target region. For SHSs with both stochastic and non-deterministic behavior, the problem results in general in a range of probabilities, thereby becoming an optimization problem. To describe stochastic dynamics, uncertainties have been added to hybrid systems in various ways, resulting in different stochastic hybrid model classes.

In this chapter, we describe our tool *SReach* which supports probabilistic bounded δ-reachability analysis for two model classes: hybrid automata (HAs) [39] with parametric uncertainty, and

probabilistic hybrid automata (PHAs) [67] with additional randomness. (Note that, in the following, we use notations - HA_p and PHA_r - for these two model classes respectively.) Our method combines the recently proposed δ-complete bounded reachability analysis technique [34] with statistical testing techniques. *SReach* saves the virtues of the Satisfiability Modulo Theories (SMT) based Bounded Model Checking (BMC) for HAs [24, 70], namely the fully symbolic treatment of hybrid state spaces, while advancing the reasoning power to probabilistic models. Furthermore, by utilizing the δ-complete analysis method, the full non-determinism of models will be considered. The coverage of simulation will be increased, as the δ-complete analysis method results in an over-approximation of the reachable set, whereas simulation is only an under-approximation of it. The zero-crossing problem can be avoided as, if a zero-crossing point exists, it will always return an interval containing it. By using statistical tests, *SReach* can place controllable error bounds on the estimated probabilities. We discuss three biological models - an atrial fibrillation model, a prostate cancer treatment model, and our synthesized Killerred biological model - to show that *SReach* can answer questions including model validation/falsification, parameter synthesis, and sensitivity analysis.

5.1 Stochastic Hybrid Models

Before introducing the algorithm implemented by *SReach* and the problems that it can handle, we first define two model classes that *SReach* considers formally. For HA_ps, we follow the definition of HAs in [39], and extend it to consider probabilistic parameters in the following way.

Definition 5.1.1 (HA_p) *A hybrid automaton with parametric uncertainty is a tuple* $H_p = \langle (Q, E), V, RV, \mathsf{Init}, \mathsf{Flow}, \mathsf{Inv}, \mathsf{Jump}, \Sigma \rangle$*, where*

- *The vertices* $Q = \{q_1, \cdots, q_m\}$ *is a finite set of discrete modes, and edges in* E *are control switches.*
- $V = \{v_1, \cdots, v_n\}$ *denotes a finite set of real-valued system variables. We write* $\dot{V}$ *to represent the first derivatives of variables during the continuous change, and write* V' *to denote values of variables at the conclusion of the discrete change.*

- $RV = \{w_1, \cdots, w_k\}$ *is a finite set of independent random variables, where the distribution of* w_i *is denoted by* P_i*.*
- Init, Flow*, and* Inv *are labeling functions over* Q*. For each mode* $q \in Q$*, the initial condition* $Init(q)$ *and invariant condition* $Inv(q)$ *are predicates whose free variables are from* $V \cup RV$*, and the flow condition* $Flow(q)$ *is a predicate whose free variables are from* $V \cup \dot{V} \cup RV$*.*
- Jump *is a transition labeling function that assigns to each transition* $e \in E$ *a predicate whose free variables are from* $V \cup V' \cup RV$*.*
- Σ *is a finite set of events, and an edge labeling function* $event : E \to \Sigma$ *assigns to each control switch an event.*

Another class is PHA$_r$s, which extend HAs with discrete probability transitions and additional randomness for transition probabilities and variable resets.

Definition 5.1.2 (PHA$_r$) *A probabilistic hybrid automaton with additional randomness* H_r *consists of* $Q,\ E,\ V,\ RV,$ Init, Flow, Inv, Σ *as in Definition 5.1.1, and* $Cmds$*, which is a finite set of probabilistic guarded commands of the form:*

$g \ \to \ p_1 : u_1 \ + \ \cdots \ + \ p_m : u_m$*,*

where g *is a predicate representing a transition guard with free variables from* V*,* p_i *is the transition probability for the* i*th probabilistic choice which can be expressed by an equation involving random variable(s) in* RV *and the* p_i*'s satisfy* $\sum_{i=1}^{m} p_i = 1$*, and* u_i *is the corresponding transition updating function for the* i*th probabilistic choice, whose free variables are from* $V \cup V' \cup RV$*.*

To illustrate the additional randomness allowed for transition probabilities and variable resets, an example probabilistic guarded command is $x \geq 5 \ \ \to \ \ p_1 : (x' = sin(x)) + (1 - p_1) : (x' = p_x)$, where x is a system variable, p_1 has a Uniform distribution $U(0.2, 0.9)$, and p_x has a Bernoulli distribution $B(0.85)$. This means that, the probability to choose the first transition is not a fixed value, but a random one having a Uniform distribution. Also, after taking the second transition, x can be assigned to either 1 with probability 0.85, or 0 with 0.15. In general,

for an individual probabilistic guarded command, the transition probabilities can be expressed by equations of one or more new random variables, as long as values of all transition probabilities are within $[0, 1]$, and their sum is 1. Currently, all four primary arithmetic operations are supported. Note that, to preserve the Markov property, only unused random variables can be used, so that no dependence between the current probabilistic jump and previous transitions will be introduced.

5.2 The *SReach* Algorithm

A recently proposed δ-complete decision procedure [34] relaxes the reachability problem for HAs in a sound manner: it verifies a conservative approximation of the system behavior, so that bugs will always be detected. The over-approximation can be tight (tunable by an arbitrarily small rational parameter δ), and a false alarm with a small δ may indicate that the system is fragile, thereby providing valuable information to the system designer. We now define the probabilistic bounded δ-reachability problem based on the bounded δ-reachability problem defined in [34] .

Definition 5.2.1 *The probabilistic bounded k step δ-reachability for a HA$_p$ H_p is to compute the probability that H_p reaches the target region T in k steps. Given the set of independent random variables* $\mathbf{r}$*, $Pr(\mathbf{r})$ a probability measure over* $\mathbf{r}$*, and Ω the sample space of* $\mathbf{r}$*, the reachability probability is $\int_\Omega I_T(\mathbf{r})dPr(\mathbf{r})$, where $I_T(\mathbf{r})$ is the indicator function which is 1 if H_p with* $\mathbf{r}$ *reaches T in k steps.*

Definition 5.2.2 *For a PHA$_r$ H_r, the probabilistic bounded k step δ-reachability estimated by SReach is the maximal probability that H_r reaches the target region T in k steps:*
$max_{\sigma \in E} Pr^k_{H_r,\sigma,T}(i)$, where E is the set of possible executions of H starting from the initial state i, and σ is an execution in the set E.

After encoding uncertainties using random variables, *SReach* samples them according to the given distributions. For each sample, a corresponding intermediate HA is generated by replacing random variables with their assigned values. Then, the δ-complete analyzer *dReach* is utilized to analyze each intermediate HA M_i, together with the desired precision δ and unfolding depth

Algorithm 3 SReach

```
1: function SReach(MP, ST, δ, k)
2:     if MP is a HA_p then
3:         MP ← EncRM_1(MP)                          ▷ encode uncertain system parameters
4:     else                                          ▷ otherwise a PHA_r
5:         MP ← EncRM_2(MP)          ▷ encode probabilistic jumps and extra randomness
6:     end if
7:     Succ, N ← 0                   ▷ number of δ-sat samples and total samples
8:     Assgn ← ∅          ▷ record unique sampling assignments and dReach results
9:     RV ← ExtractRV(MP)            ▷ get the RVs from the probabilistic model
10:    repeat in parallel
11:        S_i ← Sim(RV)                                  ▷ sample the parameters
12:        if S_i ∈ Assgn.sample then
13:            Res ← Assgn(S_i).res                       ▷ no need to call dReach
14:        else
15:            M_i ← Gen(MP, S_i)                         ▷ generate a dReach model
16:            Res ← dReach(M_i, δ, k)   ▷ call dReach to solve k-step δ-reachability
17:        end if
18:        if Res = δ-sat then Succ ← Succ + 1
19:        end if
20:        N ← N + 1
21:    until ST.done(Succ, N)                             ▷ perform statistical test
22:    return ST.output
23: end function
```

k. The analyzer returns either unsat or δ-sat for M_i. This information is then used by a chosen statistical testing procedure to decide whether to stop or to repeat the procedure, and to return the estimated probability. The full procedure is illustrated in Algorithm 3, where MP is a given stochastic model, and ST indicates which statistical testing method will be used. Note that, for a PHA$_r$, sampling and fixing the choices of all the probabilistic transitions in advance results in an over-approximation of the original PHA$_r$, where safety properties are preserved. To promise a tight over-approximation and correctness of estimated probabilities, *SReach* supports PHA$_r$s with no or subtle non-determinism. That is, in order to offer a reasonable estimation, for PHA$_r$s, *SReach* is supposed to be used on models with no or few non-deterministic transitions, or where dynamic interleaving between non-deterministic and probabilistic choices are not important.

To improve the performance of *SReach*, each sampled assignment and its corresponding *dReach* result are recorded for avoiding redundant calls to *dReach*. This significantly reduces

the total calls for PHA$_r$s, as the size of the sample space involving random variables describing probabilistic jumps is comparatively small. Furthermore, a parallel version of *SReach* has been implemented using OpenMP, where multiple samples and corresponding HAs are generated, and passed to *dReach* simultaneously.

Currently, *SReach* supports a number of hypothesis testing methods - Lai's test [50], Bayes factor test [47], Bayes factor test with indifference region [76], and Sequential probability ratio test (SPRT)[73], and statistical estimation techniques - Chernoff-Hoeffding bound [42], Bayesian Interval Estimation with Beta prior[77], and Direct Sampling. All methods produce answers that are correct up to a precision that can be set arbitrarily by the user.

With these hypothesis testing methods, *SReach* can answer qualitative questions, such as "Does the model satisfy a given reachability property in k steps with probability greater than a certain threshold?" With the above statistical estimation techniques, *SReach* can offer answers to quantitative problems. For instance, "What is the probability that the model satisfies a given reachability property in k steps?" *SReach* can also handle additional types of interesting problems by encoding them as probabilistic bounded reachability problems. The **model validation/falsification** problem with prior knowledge can be encoded as a probabilistic bounded reachability question. After expressing prior knowledge about the given model as reachability properties, is there any number of steps k in which the model satisfies a given property with a desirable probability? If none exists, the model is incorrect regarding the given prior knowledge. The **parameter synthesis** problem can also be encoded as a probabilistic k-step reachability problem. Does there exist a parameter combination for which the model reaches the given goal region in k steps with a desirable probability? If so, this parameter combination is potentially a good estimation for the system parameters. The goal here is to find a combination with which all the given goal regions can be reached in a bounded number of steps. Moreover, **sensitivity analysis** can be conducted by a set of probabilistic bounded reachability queries as well: Are the results of reachability analysis the same for different possible values of a certain system parameter? If so, the model is insensitive to this parameter with regard to the given prior knowledge.

5.3 Case Studies

Both sequential and parallel versions of *SReach* are available on `https://github.com/dreal/SReach` Experiments for the following three biological models were conducted on a server with 2* AMD Opteron(tm) Processor 6172 and 32GB RAM (12 cores were used), running on Ubuntu 14.0.1 LTS. In our experiments we used 0.001 as the precision for the δ-decision problem, and Bayesian sequential estimation with 0.01 as the estimation error bound, coverage probability 0.99, and a uniform prior ($\alpha = \beta = 1$). All the details (including discrete modes, continuous dynamics that described by ODEs, non-determinism, and stochasticity) of models in the following case studies and additional benchmarks can be found on the tool website.

Atrial Fibrillation. The minimum resistor model reproduces experimentally measured characteristics of human ventricular cell dynamics [18]. It reduces the complexity of existing models by representing channel gates of different ions with one fast channel and two slow gates. However, due to this reduction, for most model parameters, it becomes impossible to obtain their values through measurements. After adding parametric uncertainty into the original hybrid model, we show that *SReach* can be adapted to synthesize parameters for this stochastic model, i.e., identifying appropriate ranges and distributions for model parameters. We chose two system parameters - EPI_TO1 and EPI_TO2, and varied their distributions to see which ones allow the model to present the desired patterns. As in Table 5.1, when EPI_TO1 is either close to 400, or between 0.0061 and 0.007, and EPI_TO2 is close to 6, the model can satisfy the given bounded reachability property with a probability very close to 1.

Model	#RVs	EPI_TO1	EPI_TO2	#S_S	#T_S	Est_P	A_T(s)	T_T(s)
Cd_to1_s	1	U(6.1e-3, 7e-3)	6	240	240	0.996	0.270	64.80
Cd_to1_uns	1	U(5.5e-3, 5.9e-3)	6	0	240	0.004	0.042	10.08
Cd_to2_s	1	400	U(0.131, 6)	240	240	0.996	0.231	55.36
Cd_to2_uns	1	400	U(0.1, 0.129)	0	240	0.004	0.038	9.15
Cd_to12_s	2	N(400, 1e-4)	N(6, 1e-4)	240	240	0.996	0.091	21.87
Cd_to12_uns	2	N(5.5e-3, 10e-6)	N(0.11, 10e-5)	0	240	0.004	0.037	8.90

Table 5.1: Results for the 4-mode atrial fibrillation model ($k = 3$). For each sample generated, *SReach* analyzed systems with 62 variables and 24 ODEs in the unfolded SMT formulae. #RVs = number of random variables in the model, #S_S = number of δ-sat samples, #T_S = total number of samples, Est_P = estimated probability of property, A_T(s) = average CPU time of each sample in seconds, and T_T(s) = total CPU time for all samples in seconds. Note that, we use the same notations in the remaining tables.

Prostate cancer treatment. This model is a nonlinear hybrid automaton with parametric uncertainty. We modified the model of the intermittent androgen suppression (IAS) therapy in [68] by adding parametric uncertainty. The IAS therapy switches between treatment-on, and treatment-off with respect to the serum level thresholds of prostate-specific antigen (PSA), namely r_0 and r_1. As suggested by the clinical trials [16], an effective IAS therapy highly depends on the individual patient. Thus, we modified the model by taking parametric variation caused by personalized differences into account. In detail, according to clinical data from hundreds of patients [17], we replaced six system parameters with random variables having appropriate (continuous) distributions, including α_x (the proliferation rate of androgen-dependent (AD) cells), α_y (the proliferation rate of androgen-independent (AI) cells), β_x (the apoptosis rate of AD cells), β_y (the apoptosis rate of AI cells), m_1 (the mutation rate from AD to AI cells), and z_0 (the normal androgen level). To describe the variations due to individual differences, we assigned α_x to be $U(0.0193, 0.0214)$, α_y to be $U(0.0230, 0.0254)$, β_x to be $U(0.0072, 0.0079)$, β_y to be $U(0.0160, 0.0176)$, m_1 to be $U(0.0000475, 0.0000525)$, and z_0 to be $N(30.0, 0.001)$. We used *SReach* to estimate the probabilities of preventing the relapse of prostate cancer with three distinct pairs of treatment thresholds (*i.e.*, combinations of r_0 and r_1). As shown in Table 5.2, the model with thresholds $r_0 = 10$ and $r_1 = 15$ has a maximum posterior probability that approaches 1, indicating that these thresholds may be considered for the general treatment.

Model	#RVs	r_0	r_1	Est_P	#S_S	#T_S	A_T(s)	T_T(s)
PCT1	6	5.0	10.0	0.496	8226	16584	0.596	9892
PCT2	6	7.0	11.0	0.994	335	336	54.307	18247
PCT3	6	10.0	15.0	0.996	240	240	506.5	121560

Table 5.2: Results for the 2-mode prostate cancer treatment model ($k = 2$). For each sample generated, *SReach* analyzed systems with 41 variables and 10 ODEs in the unfolded SMT formulae.

Synthesized Stochastic KillerRed Model. One approach to antibiotic resistance is to engineer a temperate phage λ with light-activated production of superoxide (SOX). The incorporated Killerred protein is phototoxic and provides another level of controlled bacteria killing [54]. A PHA_r with subtle non-determinism for our synthesized Killerred model (as shown in Figure 5.1) has been constructed. Considering individual differences of bacterial cells and distinct exper-

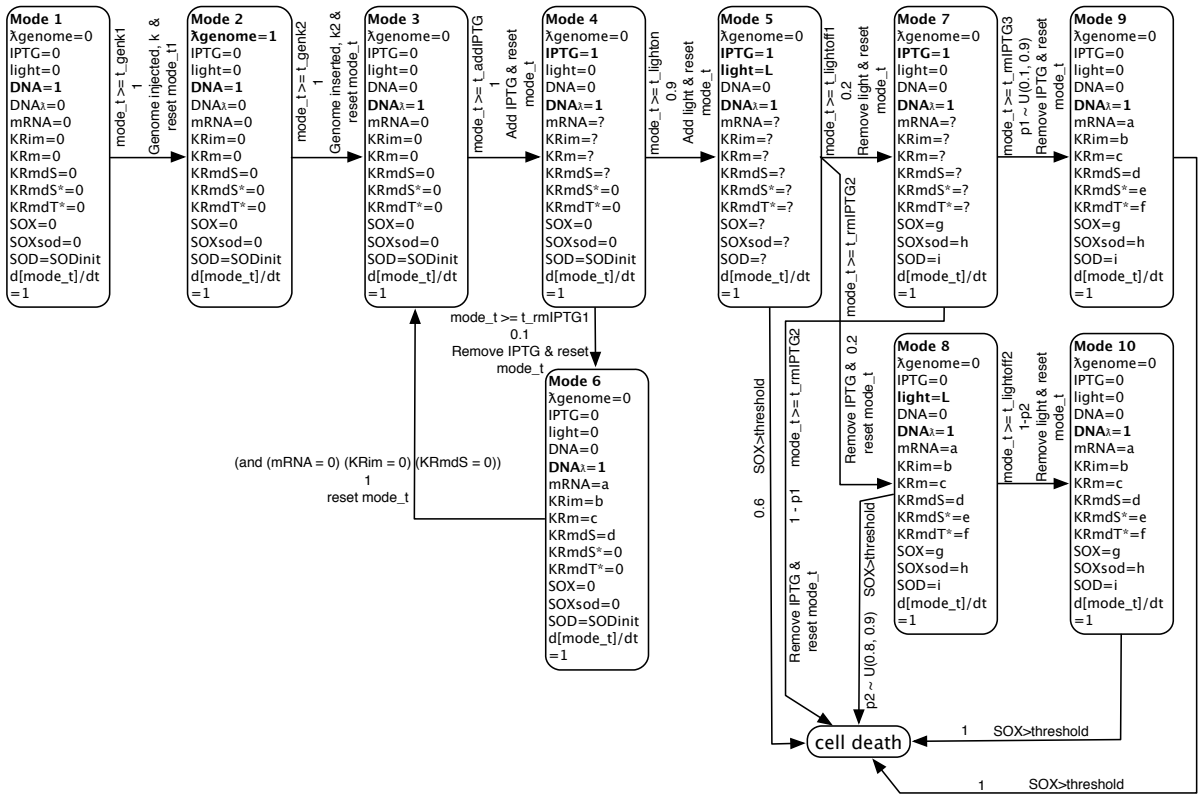

Figure 5.1: A probabilistic hybrid automaton for synthesized phage-based therapy model

imental environments, additional randomness on transition probabilities have been considered. *SReach* was used to validate this model by estimating the probabilities of killing bacterial cells with different ks (see Table 5.3). We noticed that the probabilities of paths going through mode 6 to mode 11 are close to 0. To exclude the effect from sampling of rare events, we increase the probability of entering mode 6, but this situation remains. We conclude that it is impossible for this model to enter mode 6. This remains even after increasing the probability of entering mode 6, indicating that it is impossible for this model to enter mode 6.

k	Est_P	#S_S	#T_S	A_T(s)	T_T(s)	k	Est_P	#S_S	#T_S	A_T(s)	T_T(s)
5	0.544	8951	16452	0.074	1219.38	8	0.004	0	240	0.004	0.88
6	0.247	3045	12336	0.969	11957.12	9	0.004	0	240	0.012	2.97
7	0.096	559	5808	5.470	31770.36	10	0.004	0	240	0.013	3.18

Table 5.3: Results for the 11-mode killerred model.

Chapter 6

Completed Work: Pancreatic Cancer Microenvironment Model as A Multiscale Hybrid Rule-based Model and Statistical Model Checking

As mentioned in chapter 2, the poor prognosis for Pancreatic cancer (PC) remains largely unchanged. To turn this tide, the research focus of pancreatic cancer has been shifted from solely looking into pancreatic cancer cells towards investigating the microenvironment of the pancreatic cancer. Biologists have recently noticed that one contributing factor to the failure of systemic therapies may be the abundant tumor micro-environment. As a characteristic feature of PC, the microenvironment includes pancreatic stellate cells (PSCs), endothelial cells, nerve cells, immune cells, lymphocytes, dendritic cells, the extracellular matrix, and other molecules surrounding PCCs [48]. Over the past decade, evidence has been accumulated to demonstrate the potentially critical functions of these cells in regulating the growth, invasion, and metastasis of PC [29, 31, 32, 48]. Among these cells, PSCs and cancer-associated macrophages play primary roles during the development of PC [48]. Studies have confirmed that PSCs are the primary

cells producing the stromal reaction [5, 7]. In a healthy pancreas, PSCs exist quiescently in the periacinar, perivascular, and periductal space. While, in the diseased state, PSCs will be activated by growth factors, cytokines, and oxidant stress secreted or induced by PCCs. Activated PSCs will then transform from the quiescent state to the myofibroblast phenotype. This results in their losinlipid droplets, actively proliferating, migrating, producing large amounts of extracellular matrix, and expressing cytokines, chemokines, and cell adhesion molecules. In return, the activated PSCs promote the growth of PCCs.

we construct a multicellular model to study the microenvironment of PC. The model consists of intracellular signaling networks of pancreatic cancer cells and stellate cells respectively, and intercellular interactions among them as well. To perform formal analysis, we propose a multiscale hybrid rule-based modeling formalism by extending the rule-based language BioNetGen [30]. The latter one was designed to model reactions happening among molecules within a single cell. By using the extended modeling language, we represent the intercellular level dynamics in the pancreatic cancer microenviroment as continuous, and intracellular ones as discrete considering that it is very difficult to obtain reaction rates for complex signaling networks via experimental measurements. We then apply statistical model checking (StatMC) to analyze properties of the system. The formal analysis results show that our model reproduces existing experimental findings with regard to the mutual promotion between pancreatic cancer and stellate cells. The model also explains how treatments latching onto different targets may result in distinct outcomes. We then use our model to predict possible targets for drug discovery.

6.1 Multiscale Hybrid Rule-based Modeling Language

Cell signaling embraces cellular processes that molecules outside of the cell bind to cognate receptors on the cell membrane, resulting in complex series of protein binding and biochemical events, which ultimately leads to the activation or deactivation of proteins that regulate gene expression or other cellular processes [3]. A typical signaling protein has multiple interaction sites

with activities that can be modified by direct chemical modification or by the effects of modification or interaction at other sites. This complexity at the protein level leads to a combinatorial explosion in the number of possible species and reactions at the level of signaling networks [41], which then poses a major barrier to the development of detailed, mechanistic models of biochemical systems. Rule-based modeling [13, 25, 26, 30] is a modeling paradigm that was proposed to alleviate this problem. It provides a rich yet concise description of signaling proteins and their interactions by representing interacting molecules as structured objects and by using pattern-based rules to encode their interactions. (See [26, 30, 63] for overviews of rule-based languages.)

The traditional rule-based modeling aims at representing molecules as structured objects and molecular interactions as rules for transforming the attributes of these objects. It is used to specify protein-to-protein reactions within cells and track concentrations of different proteins. One widely used rule-based modeling formalism is the BioNetGen language [30]. Its semantics includes three components: basic building blocks, patterns, and rules. For the BioNetGen, basic building blocks are molecules that may be assembled into complexes through bonds that link components of different molecules, patterns selects particular attributes of molecules in species, and rules specify the biochemical transformations that can take place in the system and be used to build up a network of species and reactions. In this paper, in order to model the dynamics of multiple cells, interactions among cells, and intracellular reactions in the mean time, we have extended it into multiscale hybrid rule-based modeling in the following way.

The basic building blocks

For the new language, the fundamental blocks can be either cells or extracellular molecules. In detail, a cell is treated as a fundamental block with subunits representing all components constructing its intracellular signaling network, which includes intracellular species and cell functions. While, each extracellular molecule is treated as a fundamental block without any subunits within it. For each subunit, it can take discrete values. Note that, as in our microenvironment model, subunits take boolean values, we will consider boolean values in the following explana-

tions and instances. All of these can be extended for discrete values in a straightforward way.

The boolean values - True (T) and False (F) - can have different biological meanings for distinct types of components within the cell. For each subunit representing a cell function or a secretion, “T” means the cell function/secretion is triggered, and “F” not triggered. For a receptor, “T” means the receptor is bounded with the corresponding ligand, and “F” means it is free. While, for other molecules within a cell, “T” indicates the high concentration of this molecule, and “F” indicates that the concentration level of this molecule is below the value to regulate (activate or inhibit) the downstream targets.

Patterns

As the second component for the modeling language, patterns are used to identify a set of species that share a set of features. Their behavior is illustrated in Figure 6.1. The semantics of patterns used in here are the same as the original one for BioNetGen.

Rules

The original BioNetGen has specified three types of rules - binding/unbinding, phosphorylation, and dephosphorylation. In order to be able to describe cellular actions and human/treatment interventions, we have extended usable rules in the following way.

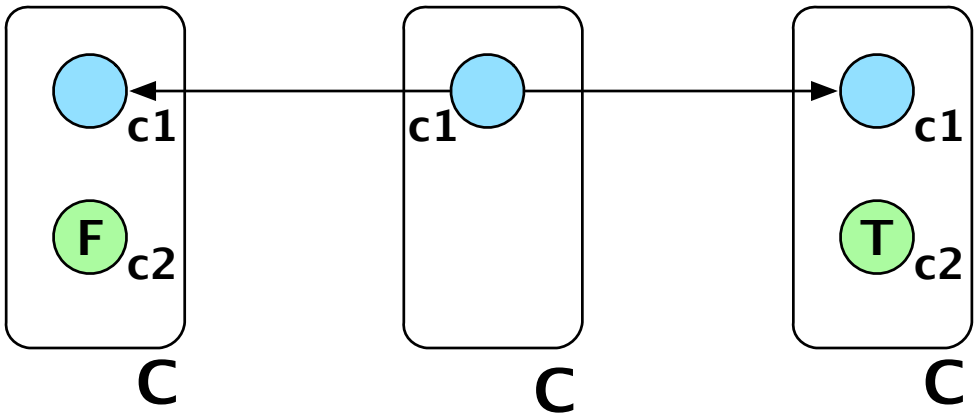

Figure 6.1: Patterns in rule-based modeling. In this example, the pattern C(c1) matches C(c1, c2∼T) or C(c1, c2∼F)

Rule 1: Ligand-receptor binding

$$Lig + Cell(Rec \sim F) \rightarrow Cell(Rec \sim T) \quad brate$$

Explanation: On the left hand, the "F" value of "Rec" in this cell indicates that the receptor is free and unbound. When the ligand has bound with this receptor, the reduction of number of extracellular molecule "Lig" is represented by the elimination of this "Lig". In the meanwhile, "Rec∼T", on the right side, indicates that this receptor is not free any more. The binding rate "brate" is decided according to affinity and whether the ligands are endogenous. Note that, the multiple receptors on the surface of a cell can be modeled by setting a comparatively high rate on the following downstream regulating rules, which indicates the rapid "releasing" of bound receptors.

Rule 2: Mutated receptors form a heterodimer

$$Cell(Rec_1 \sim F, Rec_2 \sim F) \rightarrow Cell(Rec_1 \sim T, Rec_2 \sim T) \quad frate$$

Explanation: The unbounded receptors can bind together and form a heterodimer. For example, mutated HER2 receptor activates the downstream signaling pathways of EGFR by binding with it and forming a heterodimer. That is, HER2 can be "Rec_1" and EGFR can be "Rec_1" in this rule.

Rule 3: Downstream regulation

Rule 3.1 (Single parent) Positive regulation (activation, phosphorylation, etc.)

$$Cell(Mol_1 \sim T, Mol_2 \sim F) \rightarrow Cell(Mol_1 \sim T, Mol_2 \sim T) \quad trate$$

Rule 3.2 (Single parent) Negative regulation (inhibition, dephosphorylation, etc.)

$$Cell(Mol_1 \sim T, Mol_2 \sim T) \rightarrow Cell(Mol_1 \sim T, Mol_2 \sim F) \quad trate$$

Rule 3.3 (Multiple parents) Downstream regulation

$$Cell(Mol_1 \sim F, Mol_2 \sim T, Mol_3 \sim F) \rightarrow$$

$$Cell(Mol_1 \sim F, Mol_2 \sim T, Mol_3 \sim T) \quad trate$$

$$Cell(Mol_1 \sim T, Mol_3 \sim T) \rightarrow Cell(Mol_1 \sim T, Mol_3 \sim F) \quad trate$$

Explanation: Downsteam regulation rules are used to describe the logical updating funtions. For instance, Rule 3.1 is consistent with the logical updating funtion for "Mol_2": $Mol_2^{(t+1)} = Mol_1^{(t)} + Mol_2^{(t)}$, where "$Mol_1$" is the single activator of "Mol_2". Rule 3.2 describes the funtion $Mol_2^{(t+1)} = \neg Mol_1^{(t)} \times Mol_2^{(t)}$, where "$Mol_1$" is the single inhibitor of "Mol_2". Rule 3.3 presents the updating funtion $Mol_3^{(t+1)} = \neg Mol_1^{(t)} \times (Mol_2^{(t)} + Mol_3^{(t)})$, where "$Mol_1$" is the inhibitor, and "Mol_2" is the activator. In this way, rules can be easily written for more complex cases where there are multiple regulating parents. Note that, in our model, we follow the biological assumption that inhibitors hold higher priorities than activators with regard to impacts on the regulating target.

Rule 4: Cell functions

For different cell functions, we specify distinct rules as follows.

Rule 4.1 Proliferation

$$Cell(Pro \sim T) \rightarrow Cell(Pro \sim F) + Cell(Pro \sim F, \cdots) \quad prate$$

Explanation: When a cell proliferates, we keep the current values of subunits for the cell that initiates the proliferation, and set the default values to subunits of the new cell. The "$\cdots$" in the rule denotes the remaining subunits with their default values in this cell.

Rule 4.2 Apoptosis

$$Cell(Apo \sim T) \rightarrow Null() \quad aprate$$

Explanation: We declare a type "Null()" to represent dead cells or degradated molecules.

Rule 4.3 Autophagy

$$Cell(Aut \sim T) \rightarrow Mol_1 + \cdots \quad aurate$$

Explanation: The molecules on the right side of this type of rules, which will be released into the microenvironment due to the happening of autophagy, are decided according to what molecules are currently expressed inside this cell.

Rule 5: Secretion

$$Cell_1(secMol \sim T) \rightarrow Cel_1(secMol \sim F) + Mol_{cell_1} \quad srate1$$

$$Cell_2(secMol \sim T) \rightarrow Cel_2(secMol \sim F) + Mol_{cell_2} \quad srate2$$

Explanation: When the secretion of "Mol" has been triggered, the number of "Mol" in the microenvironment will be added by 1. Note that, the reason to label the secreted "Mol" with cell's name is to differentiate the endogenous and exogenous molecules. The binding rates are different for these two cases. We use this way to take the locations of secreted molecules in the microenvironment into consideration.

Rule 6: Mutation

$$Cell(Mol \sim F(/T)) \rightarrow Cell(Mol \sim T(/F)) \quad mrate$$

Explanation: The key idea of modeling mutations is to set a very high value to the mutation rate "mrate". In this way, we can almost keep the value of the mutated molesule as "T(/F)".

Rule 7: Constantly over-expressed extracellular molecules

$$CancerEvn \rightarrow CancerEvn + Mol \quad secrate$$

Explanation: With this rule, we can mimic the situation when the concentration of an over-

expressed extracellular molecule stays in a high level constantly.

Rule 8: Degradation of extracellular molecules

$$Mol \rightarrow Null() \quad degrate$$

Explanation: As mentioned in Rule 4.2, we declare a type "Null()" to represent dead cells or degradated molecules.

Rule 9: Human/treatment intervention

$$Cell(Mol \sim T(/F)) \rightarrow Cell(Mol \sim F(/T)) \quad intrate$$

Explanation: Given a validated model, with intervention rules, we can predict whether a therapy targeting at certain molecule(s) can obtain effective outcomes. Also, the well-tuned value of the intervention rate can, more or less, give indications when deciding the dose of medicine used in this therapy, based on the Law of Mass Action.

We extend the rule-based BioNetGen language by redefining its three components - basic building blocks, patterns, and rules. These redefined components allow us to be able to model not only the signaling network within a single cell, but also interactions among multiple cells. Cell populations can also be tracked. Moreover, by choosing to describe the intracellular dynamics to be discrete, we can overcome the difficulty of obtaining values of a large amount of system parameters from wet laboratory, which is a key issue that is faced by traditional rule-based languages.

6.2 The MICROENVIRONMENT Model

Accumulating evidence indicates that PSCs may play an important role during the progression of pancreatic cancer [5, 7, 28, 29, 31, 32, 44, 48, 72]. This motivates our interest in modeling and

analyzing the molecular functions with respects to PCCs and PSCs, and the interplay between these two types of cells. Our multicellular and multilevel model is visualized in Figure 6.2. This model has three parts with different colors - green, blue, and purple. The green part depicts the intracellular signaling network of PCCs. The blue part represents the intracellular signaling network of PSCs. The purple nodes in the middle are extracellular signaling molecules (such as growth factors and cytokines) existing in the microenvironment. They can trigger signaling pathways both in cancer cells and in stellate cells by binding to the corresponding receptors. In the following sections, where we will discuss different parts of this model in detail, we will use $\rightarrow$ to denote activation or promotion, and $\dashv$ to represent inhibition or repression.

6.2.1 Intracellular signaling network of PCCs

Pathways regulating proliferation

K-RAS mutation enhances proliferation [8]. Mutations of the K-RAS oncogene occur in the precancerous stages and in over 90% of the pancreatic carcinomas. The RAS signaling pathway is crucial in the transmission of the proliferation-promoting signals. Mutation of the K-RAS gene can lead to its continuous activation of the RAS protein. Then, RAS constantly triggers the RAF $\rightarrow$ MEK cascade, and promotes the proliferation of PCCs through both ERK and JNK.

HER2/neu mutation also intensifies proliferation [8]. HER2/neu is another oncogene frequently mutated in the initial formation of pancreatic cancers. The HER2 protein is a receptor tyrosine kinase that binds to the cell membrane surface. Mutated HER2 can bind with EGFR to form a heterodimer and thus activate the downstream signaling pathways of EGFR. Overexpressed HER2 can also induce the production of VEGF stimulating angiogenesis during the development of pancreatic cancer.

EGF activates proliferation and enhances it through an autocrine signaling [56]. EGF and EGF receptors (EGFR) are expressed in $\sim$95% of pancreatic cancers. EGF promotes proliferation through the RAS$\rightarrow$RAF$\rightarrow$MEK$\rightarrow$JNK cascade. It can also trigger the RAS$\rightarrow$RAF$\rightarrow$MEK

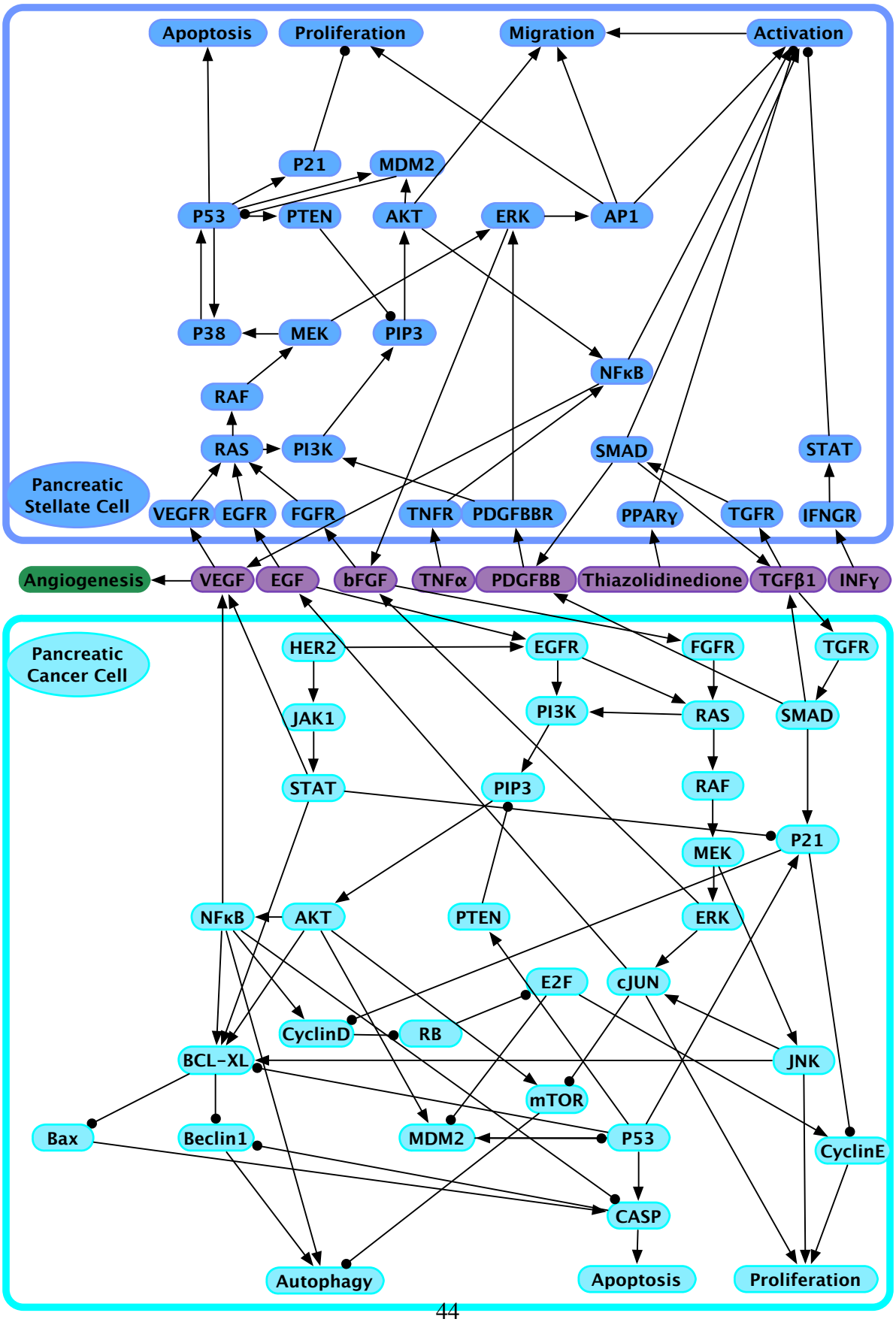

Figure 6.2: The Pancreatic Cancer Microenvironment Model

$\rightarrow$ERK$\rightarrow$cJUN cascade to secrete EGF molecules. These endogenous EGF molecules can then quickly bind to overexpressed EGFR again to promote the proliferation of pancreatic cancer cells. This autocrine provides one possible explanation of the devastating nature of pancreatic cancer.

bFGF promotes proliferation [10]. bFGF is a mitogenic polypeptide. Proliferation is activated by bFGF through both RAF$\rightarrow$MEK$\rightarrow$ERK and RAF$\rightarrow$MEK$\rightarrow$JNK cascades. In addition, bFGF molecules are released through RAF$\rightarrow$MEK$\rightarrow$ERK pathway to form another autocrine signaling pathway in the development of pancreatic cancer.

Pathways regulating apoptosis

Apoptosis is a regulated cell death mechanism. It is the most common mode of programmed cell death and is executed by caspase proteases that can be activated by either the death receptor or the mitochondrial pathways.

TGFβ1 signaling initiates apoptosis [65]. The TGFβ1 signaling mechanism, in PSCs, begins with TGFβ1 ligands binding to TGFβ1 receptors. Phosphorylated receptors further activate receptor-regulated SAMDs. The receptor-regulated SAMDs hetero-oligomerize with the common SAMD, and SAMD4. Then, the complex translocate to the nucleus, where it regulates gene expression, and is responsible for initiating apoptosis in the early stage of the pancreatic cancer development. Also, it contributes to the secretion of the TGFβ1 and PDGFBBs that are major molecules in activating PSCs. Although the TGFβ1 signaling system is a tumor suppressor pathway in the early stages of cancer progression, mutations and epigenetic dysregulation of TGFβ1 signaling mechanisms will occur later in the progression of pancreatic cancer [65]. Then, increased expression of TGFβ1 will promote the frequency of metastasis. It was also reported that this signaling is associated with poor patient prognosis of pancreatic cancer [2]. As our model describes the early stage of pancreatic cancer, TGFβ1 signaling pathway is treated as a proliferation inhibited pathway.

Mutated oncogenes inhibit apoptosis. Mutated RAS and HER2 can inhibit apoptosis by

downregulating CASP through PI3K→AKT→NFκB cascade and by inhibiting Bax (and indirectly CASP) via PI3K→PIP3→AKT→$\cdots$→BCL-XL pathways.

Pathways regulating autophagy

Autophagy is a catabolic process involving the degradation of a cell's own components through the lysosomal machinery. It is a tightly regulated process that plays a normal part in cell growth, development, and homeostasis. Autophagy helps to maintain a balance between the synthesis, degradation, and subsequent recycling of cellular products. It is a major mechanism by which a starving cell reallocates nutrients from unnecessary processes to more-essential processes. In some cellular settings, autophagy can serve as a cell survival pathway, suppressing apoptosis, and in others, it can lead to death itself, either in collaboration with apoptosis or as a back-up mechanism when the former is defective. Some of recent studies indicate that autophagy may be important in the regulation of cancer development and progression and in determining the response of cancer cells to anticancer therapy [40, 49].

mTOR regulates autophagy [55]. mTOR is a critical protein kinase that regulates autophagy induction. In pancreatic cancer, the upstream signaling pathway PI3K→PIP3→AKT can activate mTOR, and then indirectly inhibit autophagy. While, another upstream pathway MEK→ERK downregulates mTOR via cJUN, and upregulates autophagy in an indirect way.

Overexpression of anti-apoptotic factors promotes autophagy [52]. The functional relationship between apoptosis and autophagy is complex. Under certain circumstances, autophagy constitutes a stress adaptation that escapes from cell death via suppressing apoptosis. But, in other cellular settings, it constitutes an alternative cell-death pathway. Autophagy and apoptosis may be triggered by common upstream signals, and sometimes this leads to combined autophagy and apoptosis; in other instances, the cell switches between the two responses in a mutually exclusive manner. On a molecular level, this means that the apoptotic and autophagic response machineries share common pathways that either link or polarize the cellular responses. In the case of pancreatic cancer development, in the very beginning, apoptosis is increased, which inhibits

autophagy. With the progression of cancer, once apoptosis is inhibited by the high expression of anti-apoptotic factors, autophagy gradually occupies the leading role with respect to the death of cancer cells. Specifically, the overexpressed NFκB and Beclin1 can initiate autophagy.

6.2.2 Intracellular signaling network of PSCs

Pathways regulating activation

Pancreatic cancer cells can activate the surrounding PSCs. This may occur by cancer-cell-induced release of mitogenic and fibrogenic factors, such as PDGFBB, TGFβ1, and TNFα.

PDGFBB induces the activation of PSCs [37]. As a major growth factor regulating the cell functions of pancreatic stellate cells, PDGFBB activates PSCs through the downstream ERK$\rightarrow$AP1 signaling pathway.

TGFβ1 also activates PSCs [37]. Another independent signaling cascade that contributes to the activation of PSCs is mediated by TGFβ1 $\rightarrow$ TGFR $\rightarrow$ SAMD. Also, the autocrine signaling of TGFβ1 can maintain the activation of PSCs.

TNFα involves in activating PSCs [53]. As a cytokine, TNFα is also involved in activating PSCs through binding to TGFR, and then indirectly activates NFκB.

Pathways regulating migration

Migration is another characteristic cell function of pancreatic stellate cells. Activated PSCs will move towards mutated PCCs, and form a cocoon for the tumor cells, which can protect tumor from therapies' attacks [5, 32].

Different growth factors promote migration. Growth factors existing in the microenvironment, such as EGF, bFGF, and VEGF, can bind with their corresponding receptors on PCSs, and activate the migration through the MAPK signaling pathway.

PDGFBB contributes to the migration [58]. PDGFBB regulates the migration of PSCs mainly through two downstream signaling pathways. First, PDGFBB can activate PI3K$\rightarrow$PIP3$\rightarrow$ AKT pathway in PSCs. Activation of this pathway mediates PDGF-induced PSCs migration, but

not proliferation. Another pathway of equal importance is the involvement of ERK→AP1 pathway that regulates activation, migration, and proliferation of PSCs.

Pathways regulating proliferation

Growth factors activate proliferation. In PSCs, as key downstream components for several signaling pathways initiated by distinct growth factors (i.e. EGF, bFGF, VEGF, and PDGF), the ERK→AP1 cascade activates the proliferation of PSCs. Compared to inactive PSCs, active ones proliferate more rapidly.

Tumor suppressers repress proliferation. Similar to PCCs, P53, P21, and PTEN act as the suppresser for the proliferation of PSCs.

Pathways regulating apoptosis

P53 upregulates modulator of apoptosis [44]. The apoptosis of PSCs is initiated by P53, which is regulated by its upstream MAPK signaling pathway.

6.2.3 Interactions between PCCs and PSCs

The mechanisms underlying the interplay between the tumor cells and the stroma are complex. PCCs release mitogenic and fibrogenic stimulants, such as EGF, bFGF, VEGF, TGFβ1, PDGF, sonic hedgehog, galectin 3, endothelin 1 and serine protease inhibitor nexin 2 [28]. These simulants may promote the activated PSC phenotype. Stellate cells in turn secrete various factors, including stromal-derived factor 1, FGF, secreted protein acidic and rich in cysteine, matrix metalloproteinases, small leucine-rich proteoglycans, periostin and collagen type I that mediate effects on tumor growth, invasion, metastasis and resistance to chemotherapy [28]. Among them, EGF, bFGF, VEGF, TGFβ1, and PDGFBB are essential molecules that have been considered in our model.

Autocrine and paracrine involving EGF/bFGF [51]. EGF and FGF can be secreted by both PCCs and PSCs. In turn, they will bind will EGFR and FGFR on both types of cells to activate the cell proliferation and further secretion of EGF and FGF.

Interplay through VEGF [72]. As a proangiogenic factor, VEGF is found to be of great importance in the activation of PSCs and angiogenesis during the progression of PCs. VEGF, secreted by PCCs, can bind with VEGFR on PSCs to activate the PI3K pathway. It further promotes the migration of PSCs through PIP3→AKT, and suppresses the transcription activity of P53 via MDM2.

Autocrine and paracrine involving TGFβ1 [51]. The TGFβ1 signaling system controls a wide range of cellular functions that depend on cell types. In epithelial cells, TGFβ1 may play several roles including inhibition of cell growth, and initiation of apoptosis. In contrast, the effects of TGFβ1 on cellular growth and apoptosis in stromal fibroblasts are minor compared with its potent ability to stimulate cell-matrix adhesion and matrix remodeling and promotion of cell motility. PSCs by themselves are capable of synthesizing cytokines, such as TGFβ1, suggesting the existence of autocrine loops that may contribute to the perpetuation of PSC activation after an initial exogenous signal, thereby promoting the development of pancreatic fibrosis.

Interplay through PDGFBB [28]. PDGFBB exists in the secretion of pancreatic cancer cells. Its production is regulated by TGFβ1 signaling pathway. PDGFBB is highly involved in the intracellular signaling network. It can activate PSCs and initiate migration and proliferation as well.

6.3 Results and Discussion

Simulation can recapitulate a number of experimental observations and provide new insights into the system. However, it is not easy to manually analyze a significant amount of simulation results, especially when there is a large set of system properties to be tested. Thus, for our model, we apply statistical model checking (StatMC) [45]. Given a system property expressed as a Bounded Linear Temporal Logic (BLTL) [45] formula and the set of simulation trajectories with respect to the model, StatMC will return the estimated probability of the model satisfying the property with seconds. In this section, we present and discuss formal analysis results for our pancreatic cancer

microenvironment model. We implement this model in multiscale hybrid rule-based modeling language. All the experiments reported below were conducted on a machine with a 1.7 GHz Intel Core i7 processor and 8GBRAM, running on Ubuntu 14.04.1 LTS. In our experiments, we use Bayesian sequential estimation with 0.01 as the estimation error bound, coverage probability 0.99, and a uniform prior (α = β = 1).

Scenario I: mutated PCCs with no treatments

With our model, we can study both molecular dynamics, such as impacts on cell fates from key oncoproteins and tumor suppressors, and cellular behaviors. In here, to highlight the ability of our proposed modeling language in expressing cellular interactions, comparing to logical models and traditional rule-based models, we choose to look into some BLTL properties involving the interplay between PCCs and PSCs.

Property 1: This property aims to estimate the probability that the population of PCCs will eventually reach and maintain in a high level.

$$Prob_{=?}\ \{(PCCtot = 10) \wedge F^{1200}\ G^{100}\ (PCCtot > 200)\}$$

First, we take a look at the impact from the existence of PSCs on the population change of PCCs. As shown in Table 6.1, with PSCs, the probability of the number of PCCs reaching and keeping in a high level (0.9961) is much higher than the one when PSCs are absent (0.405). This indicates that PSCs promote PCCs' proliferation during the progression of PC, which is consistent with experimental findings [5, 28, 72]. Note that, the time bounds and thresholds given in this and following properties are defined considering the model's simulation results.

Property 2: This property aims to estimate the probability that the number of migrated PSCs will eventually reach and maintain in a high amount.

$$Prob_{=?}\ \{(MigPSC = 0) \wedge F^{1200}\ G^{100}\ (MigPSC > 40)\}$$

Property	Estimated Prob	# Succ	# Sample	Time (s)	Note
Scenario I: mutated PCCs with no treatments					
1	0.4053	10585	26112	208.91	w.o. PSCs
	0.9961	256	256	1.83	w. PSCs
2	0.1191	830	6976	49.69	w.o. PCCs
	0.9961	256	256	1.75	w. PCCs
3	0.9961	256	256	5.21	-
4	0.9961	256	256	4.38	-
Scenario II: mutated PCCs with different exsiting treatments					
5	0.0004	0	2304	17.13	cetuximab and erlotinib
	0.0004	0	2304	16.28	bevacizumab
	0.0012	10	9152	68.67	gemcitabine
	0.7810	8873	11360	114.25	nab-paclitaxel
	0.8004	7753	9686	73.83	ruxolitinib
Scenario III: mutated PCCs with blocking out on possible target(s)					
6	0.0792	38363	484128	3727.99	w.o. inhibiting ERK in PSCs
	0.9822	2201	2240	17.37	w. inhibiting ERK in PSCs
7	0.1979	3409	17232	136.39	w.o. inhibiting ERK in PSCs
	0.9961	256	256	2.01	w. inhibiting ERK in PSCs
8	0.2029	2181	10752	92.57	w.o. inhibiting MDM2 in PSCs
	0.9961	256	256	2.18	w. inhibiting MDM2 in PSCs
9	0.0004	0	2304	15.77	w.o. inhibiting RAS in PCCs and ERK in PSCs
	0.9961	256	256	3.15	w. inhibiting RAS in PCCs and ERK in PSCs
10	0.9797	1349	1376	11.98	w.o. inhibiting STAT in PCCs and NFκB in PSCs
	0.1631	1476	9056	81.61	w. inhibiting STAT in PCCs and NFκB in PSCs

Table 6.1: Statistical model checking results for properties under different scenarios

We then study the impacts from PCCs on PSCs. As shown in Table 6.1, without PCCs, it is quite unlikely (0.1191) for quiescent PSCs to be activated. While, when PCCs exist, the chance of PSCs becoming active (0.9961) approaches 1. This confirms the observation [37] that, during the development of PC, PSCs will be activated by growth factors, cytokines, and oxidant stress

secreted or induced by PCCs.

Property 3: This property aims to estimate the probability that the number of PCCs entering the apoptosis phase will be larger than the number of PCCs starting the autophagy programme and this situation will be reversed eventually.

$$Prob_{=?}\ \{F^{400}\ (G^{300}\ (ApoPCC > 50 \wedge AutoPCC < 50)$$

$$\wedge F^{700}\ G^{300}\ (ApoPCC < 50 \wedge AutoPCC > 50))\}$$

We are also interested in the mutually exclusive relationship between apoptosis and autophagy for PCCs reported in [40, 52]. In detail, as PC progresses, apoptosis firstly overwhelms autophagy, and then autophagy takes the leading place after a certain time point. We use property 3 to describe this situation. The estimated probability is close to 1 (see Table 6.1).

Property 4: This property aims to estimate the probability that, it is always the case that, once the population of activated PSCs reaches a high level, the number of migrated PSCs will also increase.

$$Prob_{=?}\ \{G^{1600}\ (ActPSC > 10 \rightarrow F^{100}\ (MigPSC > 10))\}$$

One reason why PC is hard to be cured is that activated PSCs will move towards mutated PCCs, and form a cocoon for the tumor cells, which can protect tumor from attacks caused by therapies [5, 32]. We investigate this by checking property 4, and obtain an estimated probability approaching 1 (see Table 6.1).

Scenario II: mutated PCCs with different exsiting treatments

Property 5: This property aims to estimate the probability that the population of PCCs will eventually drop to and maintain in a low amount.

$$Prob_{=?}\ \{(PCCtot = 10) \wedge F^{1200}\ G^{400}\ (PCCtot < 100)\}$$

Property 5 means that, after some time, the population of PCCs can be maintained in a compara-

tively low amount, indicating that PC is under control. We now consider 6 different drugs that are widely used in PC treatments - cetuximab, erlotinib, bevacizumab, gemcitabine, nab-paclitaxel, and ruxolitinib, and estimate the probabilities for them to satisfy property 5. As shown in Table 6.1, monoclonal antibody targeting EGFR (cetuximab), as well as direct inhibition of EGFR (erlotinib) broadly do not provide a survival benefit in pancreatic cancer. Monoclonal antibody inhibition of VEGFA (bevacizumab) does not improve survival either. Inhibition of MAPK pathway (gemcitabine) has also not been promising. These are consistent with clinical feedbacks from patients [1]. While, strategies aimed at depleting the stroma in pancreatic cancer (i.e. nab-paclitaxel) can be successful (with an estimated probability 0.7810), as reported in [71]. Also, inhibition of Jak/Stat can be very promising (with an estimated probability 0.8004), which has been discussed in [43].

Scenario III: mutated PCCs with blocking out on possible target(s) We have also used our model to predict possible targets for new therapies by considering pathway crosstalking within the signaling network and combinations of distinct targets. In here, we report 4 potential target(s) of interest.

Property 6: This property aims to estimate the probability that the number of PSCs will eventually drop to and maintain in a low level.

$$Prob_{=?}\ \{(PSCtot = 5) \wedge F^{1200}\ G^{400}\ (PSCtot < 30)\}$$

Property 7: This property aims to estimate the probability that the population of migrated PSCs will eventually stay in a low amount.

$$Prob_{=?}\ \{(MigPSC = 0) \wedge F^{1200}\ G^{100}\ (MigPSC < 30)\}$$

As we can tell from Table 6.1, inhibiting ERK in PSCs can not only lower the population of PSCs, but also inhibit PSCs' migration. The former function can reduce the assistance from

PSCs in the progression of PCs indirectly. The later one can prevent PSCs from moving towards PCCs and then form a cocoon, which will be an obstacle for cancer treatments.

Property 8: This property aims to estimate the probability that the number of PSCs entering the proliferation phase will eventually be less than the number of PSCs starting the apoptosis programme and this situation will maintain.

$$Prob_{=?}\ \{F^{1200}\ G^{400}\ ((PSCPro - PSCApop) < 0)\}$$

The increased probability (from 0.2029 to 0.9961 as shown in Table 6.1) indicates that inhibiting MDM2 in PSCs may reduce the number of PSCs by inhibiting PSCs' proliferation and/or promoting their apoptosis. Similar to the former role of inhibiting ERK in PSCs, it can help to treat PCs by alleviating the burden caused by PSCs.

Property 9: This property aims to estimate the probability that the number of bFGF will eventually stay in such a low level.

$$Prob_{=?}\ \{F^{1200}\ G^{400}\ (bFGF < 100)\}$$

As mentioned in property 5, 6, and 7, inhibiting RAS in PCCs can lower the number of PCCs, and downregulating ERK in PSCs can inhibit their proliferation and migration. Besides these, we have found another combinatorial result when inhibiting RAS in PCCs and ERK in PSCs simultaneously. That is, the concentration of bFGF in the microenvironment will drop (see Table 6.1). As bFGF is a key molecule that induces proliferation of both cell types, targeting RAS in PCCs and ERK in PSCs may be a useful treatment for PCs.

Property 10: This property aims to estimate the probability that the concentration of VEGF will eventually reach and keep in a high level.

$$Prob_{=?}\ \{F^{400}\ G^{100}\ (VEGF > 200)\}$$

Last but not least, inhibiting STAT in PCCs and NFκB in PSCs concurrently can postpone and lower the secretion of VEGF (see Table 6.1). VEGF plays an important role in the angiogenesis and metastasis of pancreatic tumors. So, the combination of STAT in PCCs and NFκB in PSCs may be another potential target for PC therapies.

Chapter 7

On-going Work: Biological Systems as General Stochastic Hybrid Models and Probabilistic Bounded Reachability Analysis

7.1 Algae-Fish-Bird-Estrogen Population Model

The fish model follows a simple tropic pyramid structure. The algae is the food source of the fish, which in turn are the food source for the birds. If no estrogen is introduced into the environment, the ecosystem is stable and the model simulates what is essentially the predator-prey interaction. Initially there is a relatively high amount of fish, and relatively low amounts of birds and algae. This puts a strain on the fish population, while simultaneously making it easy for the birds to find prey due to the combination of a large food source and low competition for that food source. Thus this leads to a dip in the fish population and a peak in the bird population. The dip in the fish population also leads to a peak in the algae population, as the algae can grow without being

consumed as fast due to the lack of fish. This scenario puts a strain on the bird population as there is now too much competition for a smaller food source, while simultaneously making it easy for the fish to find food due to the combination of a large food source and low competition for that food source. Thus the population is back to the initial starting conditions, and the model continues to cycle through these scenarios ad infinitum. The user can tamper with the ecosystem by adding varying concentrations of estrogen. The estrogen leads to the feminization of male fish, with higher concentrations of estrogen corresponding to an increased likely hood of feminization. Feminized male fish cannot reproduce, which leads to more frequent dips in the fish population and can throw the entire ecosystem out of the equilibrium that was described above. Essentially the most important thing for the model to do is to capture the effects of estrogenic on a freshwater ecosystem.

To understand how the estrogen level will feminize fish, and then how this fish population and structure change will fluctuate the bird population and algae population, we construct a model using partial differential equations (PDEs) accompanying nonlinear integro-boundary conditions and stochastic differential equations (SDEs) to describe population dynamics for this freshwater ecosystem. There are many well-defined ordinary differential equation (ODE) models for fish-birds populations []. However, ODE models (e.g. the logistic equation) for population-level statistics such as total population size cannot be expected to provide an adequate account of the dynamics of most biological populations unless they are enhanced and supported by individual-level sub-models for birth and death rates. For instance, in reality, only mature fish and birds can give birth to new borns. One way to take differences between individual organisms into account is to consider the age structure of populations. The age-specific birth and death rates are fundamental parameters in both the theory and practice of population dynamics and demography. Thus, in our model, we take the age structures for fish and birds into consideration. While, for algae, we consider the randomness caused by outside factors with respect to their reproduction rate. The equations depicting population dynamics are given as follows.

For the dynamics of the population of algae $X(t)$, we have

$$\begin{cases} \mathrm{d}X(t) = X(t)(p_1 - e_1 Y(t) - d_1)\mathrm{d}t + \sigma X(t)\mathrm{d}W_t \\ X(0) = x_0 \end{cases} \tag{7.1}$$

where,

- p_1, as a constant, is the reproduction rate for algae;
- e_1, as a constant, is the eaten rate by fish;
- d_1, as a constant, is the natural death rate for algae;
- x_0 is the initial population of algae; and
- σ is fluctuation rate.

For the fish population, we consider three different types respectively: female fish $Y_f(t)$, male fish $Y_m(t)$, and feminized male fish $Y_{m2f}(t)$.

The dynamics for the population of female fish $Y_f(t)$ is defined as follows.

$$\begin{cases} \frac{\partial Y_f(a,t)}{\partial a} + \frac{\partial Y_f(a,t)}{\partial t} = -Y_f(a,t)(d_2(a) + e_2 Z(t) + oY(t) - s_1 X(t)) \\ Y_f(0,t) = \frac{1}{2} p_2 \int_{a_{fmat}}^{a_{fmax}} Y_m(a_1,t)da_1 \int_{a_{fmat}}^{a_{fmax}} Y_f(a_2,t)da_2 \\ Y_f(a,0) = y_{f0}(a) \\ Y_f(t) = \int_0^{a_{fmax}} Y_f(a,t)da \end{cases} \tag{7.2}$$

where,

- a_{fmat}, as a constant, is the mature age of fish;
- a_{fmax}, as a constant, is the maximum age of fish;

- $d_2(a)$ is the natural death rate for fish. This function is defined as

$$d_2(a) = \begin{cases} 0, & a = 0 \\ d_2, & a \in (0, a_{fmax}) \\ 1, & a = a_{fmax} \end{cases}$$

- e_2, as a constant, is the eaten rate by bird;
- o, as a constant, is the death rate caused by the overcrowding;
- s_1, as a constant, is the surviving rate due to food consuming;
- p_2, as a constant, is contact rate between mature male and female fish for the reproduction; and
- $y_{f0}(a)$ is the initial population and age structure of female fish.

The dynamics for the population of male fish $Y_m(t)$ is defined as follows.

$$\begin{cases} \frac{\partial Y_m(a,t)}{\partial a} + \frac{\partial Y_m(a,t)}{\partial t} = -Y_m(a,t)(d_2(a) + e_2 Z(t) + oY(t) - s_1 X(t)) - \int_0^{a_{fmax}} f(a) Y_m(a,t) da \\ Y_m(0,t) = \frac{1}{2} p_2 \int_{a_{fmat}}^{a_{fmax}} Y_m(a_1,t) da_1 \int_{a_{fmat}}^{a_{fmax}} Y_f(a_2,t) da_2 \\ Y_m(a,0) = y_{m0}(a) \\ Y_m(t) = \int_0^{a_{fmax}} Y_m(a,t) da \end{cases} \tag{7.3}$$

where, $f(a)$ is the feminized rate for male fish. As the older a fish is, the more the accumulated estrogen in its body is. As the feminized rate is positively linear to the accumulated estrogen amount in the body, we define $f(a) = \frac{a}{a_{fmax}}$. $y_{m0}(a)$ is the initial population and age structure of male fish.

The dynamics for the population of feminized male fish $Y_{m2f}(t)$ is defined as follows.

$$\begin{cases} \frac{\mathrm{d}Y_{m2f}(t)}{\mathrm{d}t} = Y_{m2f}(t)(s_1 X(t) - d_2(a) - e_2 Z(t) - oY(t)) + \int_0^{a_{fmax}} f(a) Y_m(a,t) da \\ Y_{m2f}(0) = 0 \end{cases} \tag{7.4}$$

Then, the total number of fish $Y(t)$ is still the sum of three distinct types:

$$Y(t) = Y_f(t) + Y_m(t) + Y_{m2f}(t)$$

Last, the dynamics for the population of birds $Z(t)$ is defined as follows.

$$\begin{cases} \frac{\partial Z(a,t)}{\partial a} + \frac{\partial Z(a,t)}{\partial t} = Z(a,t)(s_2 Y(t) - d_3(a)) \\ Z(0,t) = \int_{a_{bmat}}^{a_{bmax}} p_3 Z(a,t) da \\ Z(a,0) = z_0(a) \\ Z(t) = \int_0^{a_{bmax}} Z(a,t) da \end{cases} \tag{7.5}$$

where,

- a_{bmat}, as a constant, is the mature age of birds;
- a_{bmax}, as a constant, is the maximum age of birds;
- $d_3(a)$ is the natural death rate for birds. This function is defined as

$$d_3(a) = \begin{cases} 0, & a = 0 \\ d_3, & a \in (0, a_{bmax}) \\ 1, & a = a_{bmax} \end{cases}$$

- p_3, as a constant, is the reproduction rate for birds;
- s_2, as a constant, is the surviving rate due to food consuming; and
- $z_0(a)$ is the initial population and age structure of birds.

7.2 Modeling Formalism: Stochastic Hybrid Systems

General Stochastic Hybrid Systems (GSHS) are a class of non-linear stochastic continuous-time hybrid dynamical systems. GSHS are characterized by a hybrid state defined by two components:

the continuous state and the discrete state. The continuous and the discrete parts of the state variable have their own natural dynamics, but the main point is to capture the interaction between them.

The time t is measured continuously. The state of the system is represented by a continuous variable x and a discrete variable i. The continuous variable evolves in some "cells" X^i (open sets in the Euclidean space) and the discrete variable belongs to a countable set Q. The intrinsic difference between the discrete and continuous variables, consists of the way that they evolve through time. The continuous state evolves according to an SDE whose vector field and drift factor depend on the hybrid state. The discrete dynamics produces transitions in both (continuous and discrete) state variables x, i. Switching between two discrete states is governed by a probability law or occurs when the continuous state hits the boundary of its state space. Whenever a switching occurs, the hybrid state is reset instantly to a new state according to a probability law which depends itself on the past hybrid state. Transitions, which occur when the continuous state hits the boundary of the state space are called forced transitions, and those which occur probabilistically according to a state dependent rate are called spontaneous transitions. Thus, a sample trajectory has the form $(q_t, x_t, t \geq 0)$, where $(x_t, t \geq 0)$ is piecewise continuous and $q_t \in Q$ is piecewise constant. Let $(0 \leq T_1 < T_2 < \cdots < T_i < T_{i+1} < \cdots)$ be the sequence of jump times.

It is easy to show that GSHS include, as special cases, many classes of stochastic hybrid processes found in the literature PDMP, SHS, etc.

If X is a Hausdorff topological space we use to denote by $\mathcal{B}(X)$ or $\mathcal{B}$ its Borel σ-algebra (the σ-algebra generated by all open sets). A topological space, which is homeomorphic to a Borel subset of a complete separable metric space is called Borel space. A topological space, which is is a homeomorphic with a Borel subset of a compact metric space is called Lusin space.

State space. Let Q be a countable set of discrete states, and let $d : Q \to \mathbb{N}$ and $\mathcal{X} : Q \to \mathbb{R}^{d(.)}$ be two maps assigning to each discrete state $i \in Q$ an open subset X^i of $\mathbb{R}^{d(i)}$. We call the set

$$X(Q, d, \mathcal{X}) = \bigcup_{i \in Q} \{i\} \times X^i$$

where

$$\partial X = \bigcup_{i \in Q} \{i\} \times \partial X^i.$$

It is clear that, for each $i \in Q$, the state space X^i is a Borel space. It is possible to define a metric ρ on X such that $\rho(x_n, x) \to 0$ as $n \to \infty$ with $x_n = (i_n, x_n^{i_n})$, $x = (i, x^i)$ if and onlyif there exists m such that $i_n = i$ for all $n \geq m$ and $x_{m+k}^i \to x^i$ as $k \to \infty$. The metric ρ restricted to any component X^i is equivalent to the usual Euclidean metric [27]. Each $\{i\} \times X^i$, being a Borel space, will be homeomorphic to a measurable subset of the Hilbert cube, $\mathcal{H}$ (Urysohn's theorem, Prop. 7.2 [11]). Recall that $\mathcal{H}$ is the product of countable many copies of $[0, 1]$. The definition of X shows that X is, as well, homeomorphic to a measurable subset of H. Then $(X, \mathcal{B}(X))$ is a Borel space. Moreover, X is a Lusin space because it is a locally compact Hausdorff space with countable base.

Continuous and discrete dynamics. In each mode X^i, the continuous evolution is driven by the following stochastic differential equation (SDE)

$$dx(t) = b(i, x(t))dt + \sigma(i, x(i))dW_t, \tag{7.6}$$

where $(W_t, t \geq 0)$ is the m-dimensional standard Wiener process in a complete probability space.

This assumption ensures, for any $i \in Q$, the existence and uniqueness (Theorem 6.2.2. in [4]) of the solution for the above SDE.

Assumption 7.2.1 (Continuous evolution) Suppose that $b : Q \times X^{(\cdot)} \to \mathbb{R}^{d(\cdot)}$, $\sigma : Q \times X^{(\cdot)} \to \mathbb{R}^{d(\cdot) \times m}$, $m \in \mathbb{N}$, are bounded and Lipschitz continuous in x.

In this way, when i runs in Q, the equation 7.6 defines a family of diffusion processes $M^i = (\Omega^i, \mathcal{F}^i, \mathcal{F}_t^i, x_t^i, \theta_t^i, P^i)$, $i \in Q$ with the state spaces $\mathbb{R}^{d(i)}$, $i \in Q$. For each $i \in Q$, the elements

$\mathcal{F}^i, \mathcal{F}^i_t, \theta^i_t, P^i, P^i_x$ have the usual meaning as in the Markov process theory.

The jump (switching) mechanism between the diffusions is governed by two functions: the jump rate λ and the transition measure R. The jump rate $\lambda : X \to \mathbb{R}_+$ is a measurable bounded function and the transition measure R maps X into the set $\mathcal{P}(X)$ of probability measures on $(X, \mathcal{B}(X))$. Alternatively, one can consider the transition measure $R : \bar{X} \times \mathcal{B} \to [0, 1]$ as a reset probability kernel.

Assumption 7.2.2 (Discrete transitions) (i) for all $A \in \mathcal{B}$, $R(\cdot, A)$ is measurable;
(ii) for all $x \in \bar{X}$ the function $R(x, \cdot)$ is a probability measure.
(iii) $\lambda : X \to \mathbb{R}_+$ is a measurable function such that $t \to \lambda(x^i_t(\omega^i))$ is integrable on $[0, \varepsilon(\omega^i))$, for some $\varepsilon(\omega^i) > 0$, for each $\omega^i \in \Omega^i$.

Since $\bar{X}$ is a Borel space, then $\bar{X}$ is homeomorphic to a subset of the Hilbert cube, $\mathcal{H}$. Therefore, its space of probabilities is homeomorphic to the space of probabilities of the corresponding subset of $\mathcal{H}$ (Lemma 7.10 [11]). There exists a measurable function $\mathfrak{F} : \mathcal{H} \times \bar{X}$ such that $R(x, A) = \mathfrak{p}\mathfrak{F}^{-1}(A)$, $A \in \mathcal{B}(X)$, where $\mathfrak{p}$ is the probability measure on $\mathcal{H}$ associated to $R(x, \cdot)$ and $\mathfrak{F}^{-1}(A) = \{\omega \in \mathcal{H} | \mathfrak{F}(\omega, x) \in A\}$. The measurability of such a function is guaranteed by the measurability properties of the transition measure R.

Construction. We construct an GSHS as a Markov 'sequence' H, which admits (M^i) as subprocesses. The sample path of the stochastic process $(x_t)_{t>0}$ with values in X, starting from a fixed initial point $x_0 = (i_0, x^{i_0}_0) \in X$ is defined in a similar manner as PDMP [27].

Let ω^i be a trajectory which starts in (i, x^i). Let $t_*(\omega^i)$ be the first hitting time of ∂X^i of the process (x^i_t). Let us define the following right continuous multiplicative functional

$$F(t, \omega^i) = I_{t<t_*(\omega^i)} exp[-\int_0^t \lambda(i, x^i_s(\omega^i))ds]. \tag{7.7}$$

This function will be the survivor function for the stopping time S^i associated to the diffusion (x^i_t), which will be employed in the construction of our model. This means that "killing" of the process (x^i_t) is done according to the multiplicative functional $F(t, \cdot)$. The stopping time S^i can

be thought of as the minimum of two other stopping times:

1. first hitting time of boundary, i.e. $t_{*|\Omega^i}$;
2. the stopping time $S^{i'}$ given by the following continuous multiplicative functional (which plays the role of the survivor function)

$$M(t,\omega^i) = exp(-\int_0^t \lambda(i, x_s^i(\omega^i)))ds.$$

The stopping time $S^{i'}$ can be defined as

$$S^{i'}(\omega^i) = sup\{t|\Lambda_t^i(\omega^i) \leq m^i(\omega^i)\},$$

where Λ_t^i is the following additive functional associated to the diffusion (x_t^i)

$$\Lambda_t^i(\omega^i) = \int_0 t\lambda(i, x_s^i(\omega^i))ds$$

and m^i is an $\mathbb{R}_+$-valued random variable on Ω^i, which is expontantially distributed with the survivor function $P_x^i[m^i > t] = e^{-t}$. Then

$$P_{x^i}^i[S^{i'} > t] = P_{x^i}^i[\Lambda_t^i \leq m']. \tag{7.8}$$

We set $\omega = \omega^{i_0}$ and the first jump time of the process is $T_1(\omega) = T_1(\omega^{i_0}) = S^{i_0}(\omega^{i_0})$. The sample path $x_t(\omega)$ up to the first jump time is now defined as follows:

if $T_1(\omega) = \infty$: $x_t(\omega) = (i_0, x_t^{i_0}(\omega^{i_0}))$, $t \geq 0$

if $T_1(\omega) < \infty$: $x_t(\omega) = (i_0, x_t^{i_0}(\omega^{i_0}))$, $0 \leq t < T_1(\omega)$

$x_{T_1}(\omega)$ is a r.v. w.r.t. $R((i_0, x_{T_1}^{i_0}(\omega^{i_0})), \cdot)$.

The process restarts from $x_{T_1}(\omega) = (i_1, x_1^{i_1})$ according to the same recipe, using now the process $x_t^{i_1}$. Thus if $T_1(\omega) < \infty$ we define $\omega = (\omega^{i_0}, \omega^{i_1})$ and the next jump time

$$T_2(\omega) = T_2(\omega^{i_0}, \omega^{i_1}) = T_1(\omega^{i_0}) + S^{i_1}(\omega^{i_1})$$

The sample path $x_t(\omega)$ between the two jump times is now defined as follows:

if $T_2(\omega) = \infty$: $x_t(\omega) = (i_1, x^{i_1}_{t-T_1}(\omega))$, $t \geq T_1(\omega)$

if $T_2(\omega) < \infty$: $x_t(\omega) = (i_1, x^{i_1}_t(\omega))$, $0 \leq T_1(\omega) \leq t < T_2(\omega)$

$x_{T_2}(\omega)$ is a r.v. w.r.t. $R((i_1, x^{i_1}_{T_2}(\omega)), \cdot)$.

and so on.

We denote $N_t(\omega) = \sum I_{(t \geq T_k)}$.

Assumption 7.2.3 (Non-Zeno executions) For every starting point $x \in X$, $EN_t < \infty$, for all $t \in \mathbb{R}_+$.

We can now define GSHS formally by:

Definition 7.2.1 (GSHS) *A General Stochastic Hybrid System (GSHS) is a collection* $H = ((Q, d, \mathcal{X}), b, \sigma,$
$Init, \lambda, R)$ *where*

- Q *is a countable set of discrete variables;*
- $d : Q \to \mathbb{N}$ *is a map giving the dimensions of the continuous state spaces;*
- $\mathcal{X} : Q \to \mathbb{R}^{d(.)}$ *maps each* $q \in Q$ *into an open subset* X^q *of* $\mathbb{R}^{d(q)}$*;*
- $b : X(Q, d, \mathcal{X}) \to \mathbb{R}^{d(.)}$ *is a vector field;*
- $\sigma : X(Q, d, \mathcal{X}) \to \mathbb{R}^{d(.) \times m}$ *is a* $X^{(.)}$*-valued matrix,* $m \in \mathbb{N}$*;*
- $Init : \mathcal{B}(X) \to [0, 1]$ *is an initial probability measure on* $(X, \mathcal{B}(S))$*;*
- $\lambda : \bar{X}(Q, d, \mathcal{X}) \to \mathbb{R}^+$ *is a transition rate function;*
- $R : \bar{X} \times \mathcal{B}(\bar{X}) \to [0, 1]$ *is a transition measure.*

Following [66], we note that if R_c is a transition measure from $(X \times Q, \mathcal{B}(X \times Q))$ to $(X, \mathcal{B}(X))$ and R_d is a transition measure from $(X, \mathcal{B}(X))$ to $(Q, \mathcal{B}(Q))$ (where Q is equipped with the discrete topology) then one might define a transition measure as follows

$$R(x^i, A) = \sum_{q \in Q} R_d(x^i, q) R_c(x^i, q, A^q)$$

for all $A \in \mathcal{B}(X)$, where $A^q = A \cap (q, X^q)$. Taking in the definition of a GSHS a such kind of reset map, the change of the continuous state at a jump depends on the pre jump location (continuous and discrete) as well as on the post jump discrete state. This construction can be used to prove that the stochastic hybrid processes with jumps, developed in [14], are a particular class of GSHS.

Also we can define GSHS executions as:

Definition 7.2.2 (GSHS Execution) *A stochastic process $x_t = (q(t), x(t))$ is called a GSHS execution if there exists a sequence of stopping times $T_0 = 0 < T_1 < T_2 \leq \cdots$ such that for each $k \in N$,*

- *$x_0 = (q_0, x_0^{q_0})$ is a $Q \times X$-valued random variable extracted according to the probability measure $Init$;*
- *For $t \in [T_k, T_{k+1})$, $q_t = q_{T_k}$ is constant and $x(t)$ is a (continuous) solution of the SDE:*

$$dx(t) = b(q_{T_k}, x(t))dt + \sigma(q_{T_k}, x(t))dW_t \tag{7.9}$$

  where W_t is a the m-dimensional standard Wiener;
- *$T_{k+1} = T_k + S^{i_k}$ where S^{i_k} is chosen according with the survivor function 7.8;*
- *The probability distribution of $x(T_{k+1})$ is governed by the law $R((q_{T_k}, x(T_{k+1}^-)), \cdot)$.*

Chapter 8

On-going Work: Joint Efforts of Formal Methods and Machine Learning to Automate Biological Model Design

We propose to create a framework that will allow for creating and studying causal, explanatory models of complicated biological systems in which interactions have important causal effects. The modules included in the framework (as in Figure 8.1) will provide functionality necessary for automation of information mining, information assembly and explanation of such systems. Within this framework, besides validating input models, explaining existing experimental observations, and offering new information for designing new experiments, model checking techniques can be used as a (sub)model selection method. In detail, when integrating multiple model fragments obtained via information mining, model checking can help to decide which fragment(s) should be included into the final model by considering the verification results against a set of basic system properties.

As the initial step of this work (see Figure 8.2), we first consider the biomedical pathways of pancreatic cancer. We use our model in Chapter 2 as the initial model, and apply BioNELL together with a given set of pathway keywords to learn additional causal relations from pancreatic

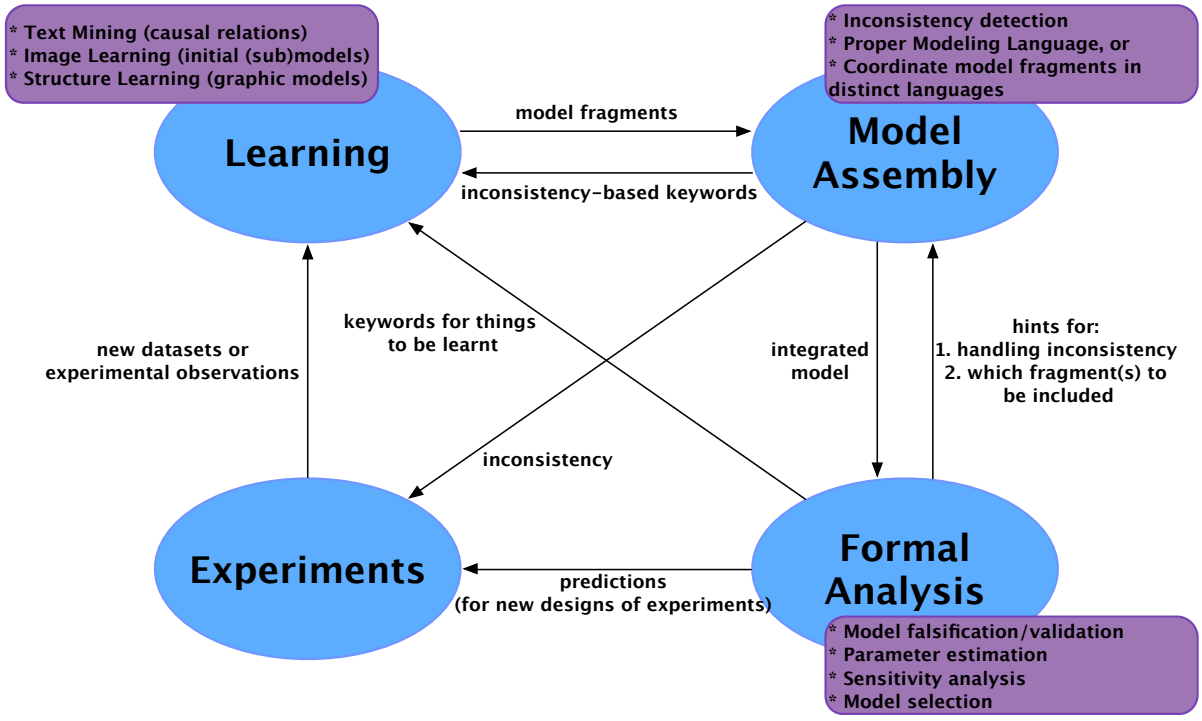

Figure 8.1: Schematic view of how formal methods and machine learning can take joint efforts to automate the model design for biological models.

cancer related literature. We rank these mined model fragments according to the frequency of appearance, the number of citations, and so on. 3-value discrete logic modeling language, as an extension of Boolean networks by consider three possible values (low, medium, and high), is used to represent the assembled model. In this work, we use statistical model checking and a set of Bounded LTL properties to select which fragment obtained from literature can be added to the final model. The whole process is automated.

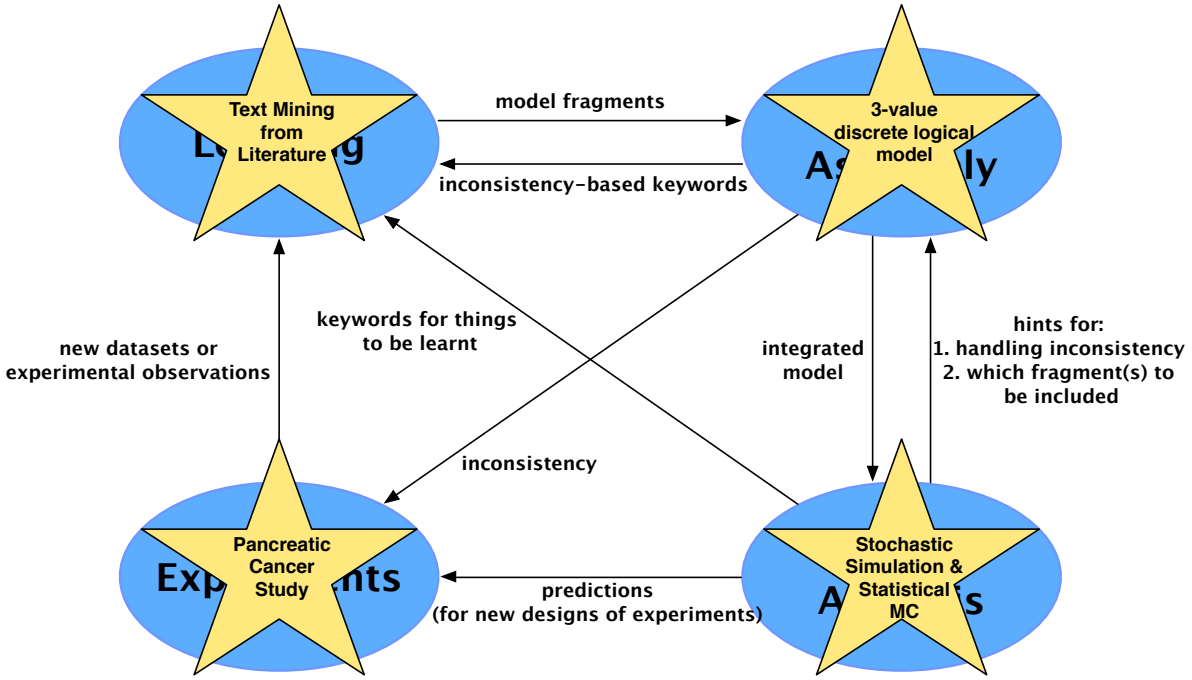
Text Mining
from
Literature
model fragments
inconsistency-based keywords
3-value
discrete logical
model
new datasets or
experimental observations
keywords for things
to be learnt
integrated
model
hints for:
1. handling inconsistency
2. which fragment(s) to
be included
inconsistency
Pancreatic
Cancer
Study
predictions
(for new designs of experiments)
Stochastic
Simulation &
Statistical
MC

Figure 8.2

Chapter 9

Timeline

My proposed timeline of work is:

1. Now - June 2016: Finish chapter 7
2. Now - April 2016: Finish chapter 8
2. July 2016: Defend thesis

Bibliography

[1] Personal communication with Jeffrey Melson Clarke, md (medical instructor in the department of medicine). 6.3

[2] Rosemary J Akhurst and Rik Derynck. Tgf-β signaling in cancer–a double-edged sword. *Trends in cell biology*, 11(11):S44–S51, 2001. 6.2.1

[3] Bruce Alberts, Alexander Johnson, Julian Lewis, Martin Raff, Keith Roberts, and Peter Walter. Molecular biology of the cell (garland science, new york, 2002). 6.1

[4] L Amod. Stochastic differential equations theory and application, 1972. 7.2

[5] MV Apte, S Park, PA Phillips, N Santucci, D Goldstein, RK Kumar, GA Ramm, M Buchler, H Friess, JA McCarroll, et al. Desmoplastic reaction in pancreatic cancer: role of pancreatic stellate cells. *Pancreas*, 29(3):179–187, 2004. 6, 6.2, 6.2.2, 6.3, 6.3

[6] Nichole Boyer Arnold and Murray Korc. Smad7 abrogates transforming growth factor-β1-mediated growth inhibition in colo-357 cells through functional inactivation of the retinoblastoma protein. *Journal of Biological Chemistry*, 280(23):21858–21866, 2005. 2.1

[7] Max G Bachem, Marion Schünemann, Marco Ramadani, Marco Siech, Hans Beger, Andreas Buck, Shaoxia Zhou, Alexandra Schmid-Kotsas, and Guido Adler. Pancreatic carcinoma cells induce fibrosis by stimulating proliferation and matrix synthesis of stellate cells. *Gastroenterology*, 128(4):907–921, 2005. 6, 6.2

[8] Nabeel Bardeesy and Ronald A DePinho. Pancreatic cancer biology and genetics. *Nature*

Reviews Cancer, 2(12):897–909, 2002. 2.2, 6.2.1

[9] David Benque, Sam Bourton, Caitlin Cockerton, Byron Cook, Jasmin Fisher, Samin Ishtiaq, Nir Piterman, Alex Taylor, and Moshe Y Vardi. Bma: Visual tool for modeling and analyzing biological networks. In *Computer Aided Verification*, pages 686–692. Springer, 2012. 3.2

[10] M Bensaid, N Tahiri-Jouti, C Cambillau, N Viguerie, B Colas, C Vidal, JP Tauber, JP Esteve, C Susini, and N Vaysse. Basic fibroblast growth factor induces proliferation of a rat pancreatic cancer cell line. inhibition by somatostatin. *International journal of cancer*, 50 (5):796–799, 1992. 6.2.1

[11] Dimitri P Bertsekas, Steven E Shreve, and Athena Scientific. Stochastic optimal central: The discrete - time case. 1996. 7.2, 7.2

[12] Antje Beyer, Peter Thomason, Xinzhong Li, James Scott, and Jasmin Fisher. Mechanistic insights into metabolic disturbance during type-2 diabetes and obesity using qualitative networks. In *Transactions on Computational Systems Biology XII*, pages 146–162. Springer, 2010. 3, 3.2

[13] Michael L Blinov, James R Faeder, Byron Goldstein, and William S Hlavacek. Bionetgen: software for rule-based modeling of signal transduction based on the interactions of molecular domains. *Bioinformatics*, 20(17):3289–3291, 2004. 6.1

[14] Henk AP Blom. Stochastic hybrid processes with hybrid jumps. *Analysis and Design of Hybrid System*, pages 319–324, 2003. 7.2

[15] Henk AP Blom, John Lygeros, M Everdij, S Loizou, and K Kyriakopoulos. *Stochastic hybrid systems: theory and safety critical applications*. Springer, 2006. 5

[16] Nicholas Bruchovsky, Laurence Klotz, et al. Final results of the Canadian prospective phase ii trial of intermittent androgen suppression for men in biochemical recurrence after radiotherapy for locally advanced prostate cancer. *Cancer*, 107(2):389–395, 2006. 5.3

[17] Nicholas Bruchovsky, Laurence Klotz, Juanita Crook, and Larry Goldenberg. Locally advanced prostate cancer: biochemical results from a prospective phase ii study of intermittent androgen suppression for men with evidence of prostate-specific antigen recurrence after radiotherapy. *Cancer*, 109(5):858–867, 2007. 5.3

[18] Alfonso Bueno-Orovio, Elizabeth M Cherry, and Flavio H Fenton. Minimal model for human ventricular action potentials in tissue. *J. of Theor. Biology*, 253(3):544–560, 2008. 5.3

[19] Daniel C Chung, Suzanne B Brown, Fiona Graeme-Cook, Masao Seto, Andrew L Warshaw, Robert T Jensen, and Andrew Arnold. Overexpression of cyclin d1 occurs frequently in human pancreatic endocrine tumors 1. *The Journal of Clinical Endocrinology & Metabolism*, 85(11):4373–4378, 2000. 2.2

[20] Alessandro Cimatti, Edmund Clarke, Enrico Giunchiglia, Fausto Giunchiglia, Marco Pistore, Marco Roveri, Roberto Sebastiani, and Armando Tacchella. Nusmv 2: An open-source tool for symbolic model checking. In *Computer Aided Verification*, pages 359–364. Springer, 2002. 2.2

[21] Koen Claessen, Jasmin Fisher, Samin Ishtiaq, Nir Piterman, and Qinsi Wang. Model-checking signal transduction networks through decreasing reachability sets. In *Technical Report MSR-TR-2013-30*. Microsoft Research, 2013. 3.1, 3.1, 3.2

[22] Edmund M Clarke and Paolo Zuliani. Statistical model checking for cyber-physical systems. In *ATVA*, pages 1–12. Springer, 2011. 5

[23] Byron Cook, Jasmin Fisher, Elzbieta Krepska, and Nir Piterman. Proving stabilization of biological systems. In *Verification, Model Checking, and Abstract Interpretation*, pages 134–149. Springer, 2011. 3.1

[24] Lucas Cordeiro, Bernd Fischer, and Joao Marques-Silva. SMT-based bounded model checking for embedded ANSI-C software. *IEEE Transactions on Software Engineering*,

38(4):957–974, 2012. 5

[25] Vincent Danos and Cosimo Laneve. Formal molecular biology. *Theoretical Computer Science*, 325(1):69–110, 2004. 6.1

[26] Vincent Danos, Jérôme Feret, Walter Fontana, Russell Harmer, and Jean Krivine. Rule-based modelling of cellular signalling. In *CONCUR 2007–Concurrency Theory*, pages 17–41. Springer, 2007. 6.1

[27] MHA Davis. Markov processes and optimization. *Chapman-Hall, London*, 1993. 7.2, 7.2

[28] Siri Dunér, Jacob Lopatko Lindman, Daniel Ansari, Chinmay Gundewar, and Roland Andersson. Pancreatic cancer: the role of pancreatic stellate cells in tumor progression. *Pancreatology*, 10(6):673–681, 2011. 6.2, 6.2.3, 6.3

[29] M Erkan, C Reiser-Erkan, CW Michalski, and J Kleeff. Tumor microenvironment and progression of pancreatic cancer. *Exp Oncol*, 32(3):128–131, 2010. 6, 6.2

[30] James R Faeder, Michael L Blinov, and William S Hlavacek. Rule-based modeling of biochemical systems with bionetgen. In *Systems biology*, pages 113–167. Springer, 2009. 6, 6.1

[31] Buckminster Farrow, Daniel Albo, and David H Berger. The role of the tumor microenvironment in the progression of pancreatic cancer. *Journal of Surgical Research*, 149(2): 319–328, 2008. 6, 6.2

[32] Christine Feig, Aarthi Gopinathan, Albrecht Neesse, Derek S Chan, Natalie Cook, and David A Tuveson. The pancreas cancer microenvironment. *Clinical Cancer Research*, 18 (16):4266–4276, 2012. 6, 6.2, 6.2.2, 6.3

[33] Sicun Gao, Soonho Kong, and Edmund M Clarke. Satisfiability modulo ODEs. In *FMCAD*, pages 105–112, Oct. 2013. 4

[34] Sicun Gao, Soonho Kong, Wei Chen, and Edmund M Clarke. δ-complete analysis for bounded reachability of hybrid systems. *CoRR*, arXiv:1404.7171, 2014. 5, 5.2

[35] Haijun Gong, Qinsi Wang, Paolo Zuliani, James R Faeder, Michael Lotze, and E Clarke. Symbolic model checking of signaling pathways in pancreatic cancer. In *BICoB*, page 245, 2011. 2

[36] Haijun Gong, Paolo Zuliani, Qinsi Wang, and Edmund M Clarke. Formal analysis for logical models of pancreatic cancer. In *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, pages 4855–4860. IEEE, 2011. 2, 2.1

[37] Paul S Haber, Gregory W Keogh, Minoti V Apte, Corey S Moran, Nancy L Stewart, Darrell HG Crawford, Romano C Pirola, Geoffrey W McCaughan, Grant A Ramm, and Jeremy S Wilson. Activation of pancreatic stellate cells in human and experimental pancreatic fibrosis. *The American journal of pathology*, 155(4):1087–1095, 1999. 6.2.2, 6.3

[38] Ernst Heinmöller, Wolfgang Dietmaier, Hubert Zirngibl, Petra Heinmöller, William Scaringe, Karl-Walter Jauch, Ferdinand Hofstädter, and Josef Rüschoff. Molecular analysis of microdissected tumors and preneoplastic intraductal lesions in pancreatic carcinoma. *The American journal of pathology*, 157(1):83–92, 2000. 2.2

[39] Thomas A Henzinger. *The theory of hybrid automata*. Springer, 2000. 5, 5.1

[40] Melanie M Hippert, Patrick S O'Toole, and Andrew Thorburn. Autophagy in cancer: good, bad, or both? *Cancer research*, 66(19):9349–9351, 2006. 6.2.1, 6.3

[41] William S Hlavacek, James R Faeder, Michael L Blinov, Alan S Perelson, and Byron Goldstein. The complexity of complexes in signal transduction. *Biotechnology and bioengineering*, 84(7):783–794, 2003. 6.1

[42] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *J American Statistical Association*, 58(301):13–30, 1963. 5.2

[43] H Hurwitz, N Uppal, SA Wagner, JC Bendell, JT Beck, S Wade, JJ Nemunaitis, PJ Stella, JM Pipas, ZA Wainberg, et al. A randomized double-blind phase 2 study of ruxolitinib (rux) or placebo (pbo) with capecitabine (cape) as second-line therapy in patients (pts) with

metastatic pancreatic cancer (mpc). *J ClinOncol*, 32:55, 2014. 6.3

[44] Robert Jaster. Molecular regulation of pancreatic stellate cell function. *Molecular cancer*, 3(1):26, 2004. 6.2, 6.2.2

[45] Sumit K Jha, Edmund M Clarke, Christopher J Langmead, Axel Legay, André Platzer, and Paolo Zuliani. A bayesian approach to model checking biological systems. In *Computational Methods in Systems Biology*, pages 218–234. Springer, 2009. 6.3

[46] Siân Jones, Xiaosong Zhang, D Williams Parsons, Jimmy Cheng-Ho Lin, Rebecca J Leary, Philipp Angenendt, Parminder Mankoo, Hannah Carter, Hirohiko Kamiyama, Antonio Jimeno, et al. Core signaling pathways in human pancreatic cancers revealed by global genomic analyses. *science*, 321(5897):1801–1806, 2008. 2, 2.1

[47] Robert E Kass and Adrian E Raftery. Bayes factors. *JASA*, 90(430):773–795, 1995. 5.2

[48] Jörg Kleeff, Philipp Beckhove, Irene Esposito, Stephan Herzig, Peter E Huber, J Matthias Löhr, and Helmut Friess. Pancreatic cancer microenvironment. *International journal of cancer*, 121(4):699–705, 2007. 6, 6.2

[49] Yasuko Kondo, Takao Kanzawa, Raymond Sawaya, and Seiji Kondo. The role of autophagy in cancer development and response to therapy. *Nature Reviews Cancer*, 5(9):726–734, 2005. 6.2.1

[50] Tze Leung Lai. Nearly optimal sequential tests of composite hypotheses. *AOS*, 16(2): 856–886, 1988. 5.2

[51] Daruka Mahadevan and Daniel D Von Hoff. Tumor-stroma interactions in pancreatic ductal adenocarcinoma. *Molecular cancer therapeutics*, 6(4):1186–1197, 2007. 6.2.3

[52] Guillermo Mariño, Mireia Niso-Santano, Eric H Baehrecke, and Guido Kroemer. Self-consumption: the interplay of autophagy and apoptosis. *Nature reviews Molecular cell biology*, 15(2):81–94, 2014. 6.2.1, 6.3

[53] Atsushi Masamune, Masahiro Satoh, Kazuhiro Kikuta, Noriaki Suzuki, Kennichi Satoh,

and Tooru Shimosegawa. Ellagic acid blocks activation of pancreatic stellate cells. *Biochemical pharmacology*, 70(6):869–878, 2005. 6.2.2

[54] Natasa Miskov-Zivanov, Qinsi Wang, Cheryl Telmer, and Edmund M. Clarke. Formal analysis provides parameters for guiding hyperoxidation in bacteria using phototoxic proteins. Technical Report CMU-CS-14-137, CMU, 2014. 5.3

[55] Diego Muilenburg, Colin Parsons, Jodi Coates, Subbulakshmi Virudachalam, and Richard J Bold. Role of autophagy in apoptotic regulation by akt in pancreatic cancer. *Anticancer research*, 34(2):631–637, 2014. 6.2.1

[56] LO Murphy, MW Cluck, S Lovas, F Ötvös, RF Murphy, AV Schally, J Permert, J Larsson, JA Knezetic, and TE Adrian. Pancreatic cancer cells require an egf receptor-mediated autocrine pathway for proliferation in serum-free conditions. *British journal of cancer*, 84 (7):926, 2001. 6.2.1

[57] Aurélien Naldi, Denis Thieffry, and Claudine Chaouiya. Decision diagrams for the representation and analysis of logical models of genetic networks. In *Computational methods in systems biology*, pages 233–247. Springer, 2007. 3

[58] PA Phillips, MJ Wu, RK Kumar, E Doherty, JA McCarroll, S Park, Ron C Pirola, JS Wilson, and MV Apte. Cell migration: a novel aspect of pancreatic stellate cell biology. *Gut*, 52 (5):677–682, 2003. 6.2.2

[59] Sergei Pletnev, Nadya G Gurskaya, Nadya V Pletneva, Konstantin A Lukyanov, Dmitri M Chudakov, Vladimir I Martynov, et al. Structural basis for phototoxicity of the genetically encoded photosensitizer killerred. *Journal of Biological Chemistry*, 284(46):32028–32039, 2009. 4

[60] Ester Rozenblum, Mieke Schutte, Michael Goggins, Stephan A Hahn, Shawn Panzer, Marianna Zahurak, Steven N Goodman, Taylor A Sohn, Ralph H Hruban, Charles J Yeo, et al. Tumor-suppressive pathways in pancreatic carcinoma. *Cancer research*, 57(9):1731–1734,

1997. 2.2

[61] Lucas Sanchez and Denis Thieffry. Segmenting the fly embryo:: a logical analysis of the pair-rule cross-regulatory module. *Journal of theoretical Biology*, 224(4):517–537, 2003. 3, 3.2

[62] Marc A Schaub, Thomas A Henzinger, and Jasmin Fisher. Qualitative networks: a symbolic approach to analyze biological signaling networks. *BMC systems biology*, 1(1):4, 2007. 3, 3.2

[63] John AP Sekar and James R Faeder. Rule-based modeling of signal transduction: a primer. In *Computational Modeling of Signaling Networks*, pages 139–218. Springer, 2012. 6.1

[64] Ilya Shmulevich, Edward R Dougherty, Seungchan Kim, and Wei Zhang. Probabilistic boolean networks: a rule-based uncertainty model for gene regulatory networks. *Bioinformatics*, 18(2):261–274, 2002. 3

[65] Peter M Siegel and Joan Massagué. Cytostatic and apoptotic actions of tgf-β in homeostasis and cancer. *Nature Reviews Cancer*, 3(11):807–820, 2003. 6.2.1

[66] Kyle Siegrist. Random evolution processes with feedback. *Transactions of the American Mathematical Society*, 265(2):375–392, 1981. 7.2

[67] Jeremy Sproston. Decidable model checking of probabilistic hybrid automata. In *Formal Techniques in Real-Time and Fault-Tolerant Systems*, pages 31–45. Springer, 2000. 5

[68] Gouhei Tanaka, Yoshito Hirata, Larry Goldenberg, Nicholas Bruchovsky, and Kazuyuki Aihara. Mathematical modelling of prostate cancer growth and its application to hormone therapy. *Phil. Trans. Roy. Soc. A: Math., Phys. and Eng. Sci.*, 368(1930):5029–5044, 2010. 5.3

[69] René Thomas, Denis Thieffry, and Marcelle Kaufman. 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, 1995. 3

[70] Cesare Tinelli. SMT-based model checking. In *NASA Formal Methods*, page 1, 2012. 5

[71] Daniel D Von Hoff, Thomas Ervin, Francis P Arena, E Gabriela Chiorean, Jeffrey Infante, Malcolm Moore, Thomas Seay, Sergei A Tjulandin, Wen Wee Ma, Mansoor N Saleh, et al. Increased survival in pancreatic cancer with nab-paclitaxel plus gemcitabine. *New England Journal of Medicine*, 369(18):1691–1703, 2013. 6.3

[72] Alain Vonlaufen, Swapna Joshi, Changfa Qu, Phoebe A Phillips, Zhihong Xu, Nicole R Parker, Cheryl S Toi, Romano C Pirola, Jeremy S Wilson, David Goldstein, et al. Pancreatic stellate cells: partners in crime with pancreatic cancer cells. *Cancer research*, 68(7):2085–2093, 2008. 6.2, 6.2.3, 6.3

[73] Abraham Wald. Sequential tests of statistical hypotheses. *The Annals of Mathematical Statistics*, 16(2):117–186, 1945. 5.2

[74] Qinsi Wang, Natasa Miskov-Zivanov, Cheryl Telmer, and Edmund M Clarke. Formal analysis provides parameters for guiding hyperoxidation in bacteria using phototoxic proteins. In *Proceedings of the 25th edition on Great Lakes Symposium on VLSI*, pages 315–320. ACM, 2015. 4.1

[75] Robb E Wilentz, Christine A Iacobuzio-Donahue, Pedram Argani, Denis M McCarthy, Jennifer L Parsons, Charles J Yeo, Scott E Kern, and Ralph H Hruban. Loss of expression of dpc4 in pancreatic intraepithelial neoplasia: evidence that dpc4 inactivation occurs late in neoplastic progression. *Cancer Research*, 60(7):2002–2006, 2000. 2.2

[76] Hakan L Younes. Verification and planning for stochastic processes with asynchronous events. Technical report, DTIC Document, 2005. 5.2

[77] Paolo Zuliani, André Platzer, and Edmund M Clarke. Bayesian statistical model checking with application to simulink/stateflow verification. In *Proceedings of the 13th ACM international conference on Hybrid Systems: Computation and Control*, pages 243–252. ACM, 2010. 5.2