AN APPROACH TO COMBINE CONTINUOUS AND DISCRETE MODELING OF INTRA-CELLULAR NETWORKS WITH APPLICATION TO RAS SIGNALING

by

Mickaël Jean-François Vimbert

Submitted to the Graduate Faculty of

The Swanson School of Engineering in partial fulfillment

of the requirements for the degree of

Master of Science

University of Pittsburgh

2018

UNIVERSITY OF PITTSBURGH

SWANSON SCHOOL OF ENGINEERING

This thesis was presented

by

Mickaël Vimbert

It was defended on

April 4, 2018

and approved by

Natasa Miskov-Zivanov, Ph.D., Assistant Professor,
Department of Electrical and Computer Engineering

Samuel Dickerson, Ph.D., Assistant Professor,
Department of Electrical and Computer Engineering

Murat Akcakaya, Ph.D., Assistant Professor,
Department of Electrical and Computer Engineering

Cheryl Telmer, Ph.D., Research Biologist,
Carnegie Mellon University

Dr. Robert Stephens, Informatics Group Lead,
National Cancer Institute

Thesis Advisor: Dr. Natasa Miskov-Zivanov, Assistant professor,
Department of Electrical and Computer Engineering

AN APPROACH TO COMBINE CONTINUOUS AND DISCRETE MODELING OF INTRA-CELLULAR NETWORKS WITH APPLICATION TO RAS SIGNALING

Mickael Vimbert, M.S

University of Pittsburgh, 2018

The three Rat Sarcoma (RAS) genes in humans (HRAS, KRAS, and NRAS) are the most common oncogenes that are mutated in human cancer. Mutations that permanently activate RAS are found in 20% to 25% of all human tumors and up to 90% in certain types of cancers. For this reason, the RAS pathway is being heavily studied, however, there are still many unknowns in regulation and downstream effects of RAS in cancer. Previous computational work that focused on signaling in the RAS pathway relied mainly on continuous models that use ordinary differential equations or reaction rule-based models. These approaches are most often limited in size due to the lack of the information required to create such models. In this work, we use a discrete modeling approach, in which elements of the intracellular signaling network are modeled as discrete variables, and the elements' regulatory functions are either implemented as logical functions, weighted sums, or min/max functions. The network that we are modeling includes a number of intertwined feedback and feed-forward loops. We developed several versions of the RAS pathway model, and our results demonstrate the advantages of the discrete logical modeling approach when studying intra-cellular signaling networks. In addition to model design and analysis, we have worked on developing a translator from the biological mechanism representation used by continuous models, into the decision tree representation, namely Binary and Algebraic Decision Diagrams. The goal of this

translation effort is to allow for a standardized and comprehensive modeling approach, which can capture all the available system parameters, while at the same time enabling model design and evaluation even in the absence of these parameters.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1.0 INTRODUCTION

The Rat Sarcoma (RAS) gene family are the most commonly mutated genes in human tumors. Three elements in this family (HRAS, KRAS, and NRAS) are known to promote cancer proliferation and growth in tumors [1]. Mutations in other upstream or downstream components of the RAS signaling pathway shown in Figure 1 can also be found in a variety of cancers. The mitogen-activated protein kinases (MAPK) – part of the EGFR/RAS/MAPK [8] pathway and downstream of the RAS signaling pathway – increase significantly in activity once there is a mutation in the RAS protein [6][7]. KRAS mutations are especially common events in cancer. They are detected in 40-45% of all colorectal cancers [2][3] and have been detected in both early and late stages of colorectal cancer [4][5], therefore, KRAS mutation might be an early event in tumor development.

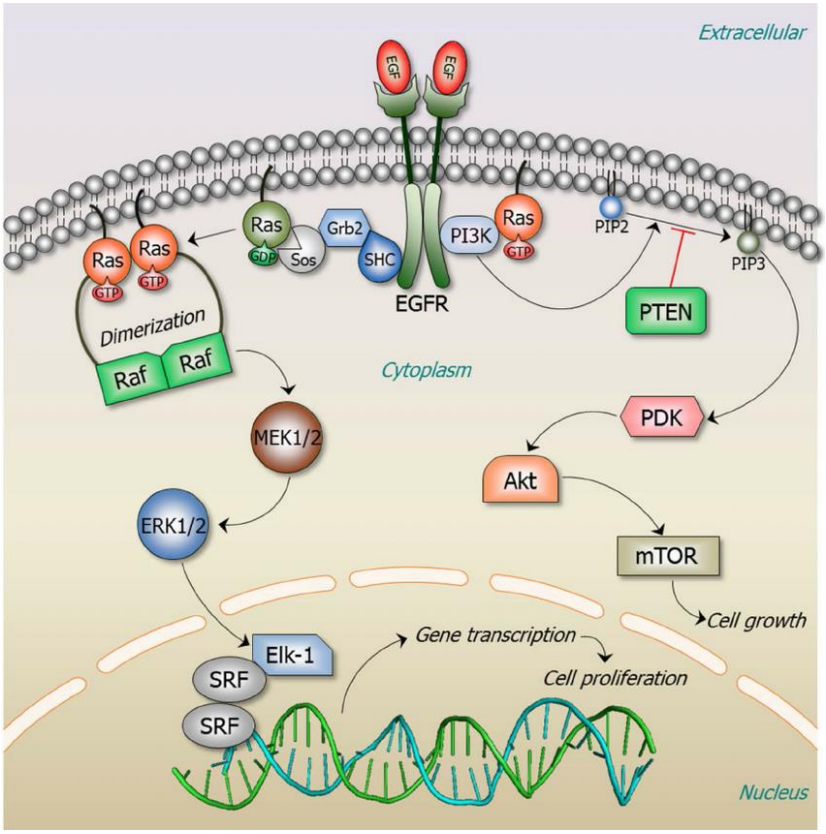


Figure 1 : Ras pathway in a cell, and its upstream and downstream elements [27]

Ras proteins are molecular switches that, by cycling between inactive guanosine diphosphate(GDP)-bound and active guanosine triphosphate(GTP)-bound forms, regulate multiple cellular signaling pathways, including those that control growth and differentiation. The oscillation between inactive and active states is a key feature to regulate cellular growth. The mutation of KRAS and its constant activity disrupt this oscillating behavior of the GDP/GTP form, thus leading to a growth that is no longer regulated, and the creation of a tumor. Modeling this pathway could give critical information about how to regulate the behavior of mutated of a mutated RAS signaling pathway to prevent the development of a tumor.

The computational modeling of intra-cellular signaling pathways such as RAS pathway usually includes Ordinary Differential Equations (ODE) that are derived from biochemical reactions [10]. Such an approach requires not only listing all the reactions that are involved in the

modeled pathway, but also the knowledge of all the reaction rates and molecular concentrations. As a result, the ODE-based modeling approach is limited in scale, usually targets only few elements on one pathway, and furthermore, the rates of all the reactions are not always known. To overcome the problem of exponential growth when modeling biochemical reaction networks, a *reaction rule-based* approach has been proposed and has been used by a number of researchers [11][12]. However, the reaction rule-based approach still requires the knowledge of reaction rates and molecular concentrations, which are not always known or available.

In the effort to avoid the problem of limited knowledge, this work uses a *discrete, element rule-based* modeling approach, which also allows to include a much larger number of elements without increasing duration or complexity of the simulation when compared to continuous modeling. The discrete modeling approach can capture characteristic dynamic behavior such as multi-stability, excitation and adaptation behavior. Whereas utilizing incomplete information about pathways (which is the case for many intra-cellular pathways) is usually not practical in ODE models, it can be dealt with by using indirect causal evidence when building discrete models.

This work aims to demonstrate how to reduce the complexity and increase the scope of a continuous model by translating it into a discrete model, without loss of biological information. For this purpose, we will analyze a reaction rule-based model from Kochańczyk et al. [11] and translate it into a discrete model, while keeping the important characteristics of their model. We developed a method that utilizes Binary and Algebraic Decision Diagrams (BDDs and ADDs) to implement both reaction rules and discrete element update rules. We show how to use these data structures in the biological context, when modeling intra-cellular networks. Due to the complexity of the RAS pathway, and its critical role in cancer development, we demonstrate our approach on

this pathway. Therefore, *this work introduces a novel element rule-based discrete model of the RAS pathway, its upstream regulation and downstream effectors*.

In Section 2.0 , we briefly discuss intra-cellular signaling networks, continuous ODE-based and reaction rule-based modeling, as well as the logical and discrete modeling. We also briefly describe the BDD data structure. Next, in Section 3.0 , we provide details of our modeling methodology to tackle several biological phenomena: spontaneous behavior, oscillations and phosphorylation. Then, we will explain in Section 4.0 how to use the data structure of an ADD and their application to analyze biological events. Finally, we will show the results obtained for the application of this work to the RAS pathway and to what it could lead to in the future.

2.0 BACKGROUND

This work applies methods from electrical and computer engineering to biology. Thus, in this section we discuss relevant background information in both domains. We first describe different modeling approaches that we use in this work, continuous (reaction rule-based) and discrete logical (element rule-based) modeling approach, followed by the discussion on an implementation method and data structures that could be used to represent models in both approaches.

2.1 MODELING INTRA-CELLULAR NETWORKS

The construction of a model begins with identifying key components of the studied system. Each component can be represented as a model element, and for each element, we determine its regulators. In Figure 2, we show several examples of intra-cellular networks. There are three main types of these networks:

- Signal transduction networks
- Metabolic networks
- Gene regulation networks

Most often, computational modelers focus only on one network type. Signal transduction networks are mainly focused on protein-protein interactions and protein kinases. The proteins responsible for detecting a stimulus are called receptors. The ligand will bind to the receptor and create a signaling cascade. This cascade results in interactions such as phosphorylation, transcription, translation of gene, post-translational changes in proteins. Metabolic networks are

directed graphs which contain genes, enzymes and biochemical reactions such as oxidation and reduction, the breakdown of glucose or the joining of amino-acids to form a protein. Gene regulatory networks can represent genes, mRNAs, protein/protein complexes or cellular processes and the interactions between these elements are usually represented as activation, inhibition and binding.

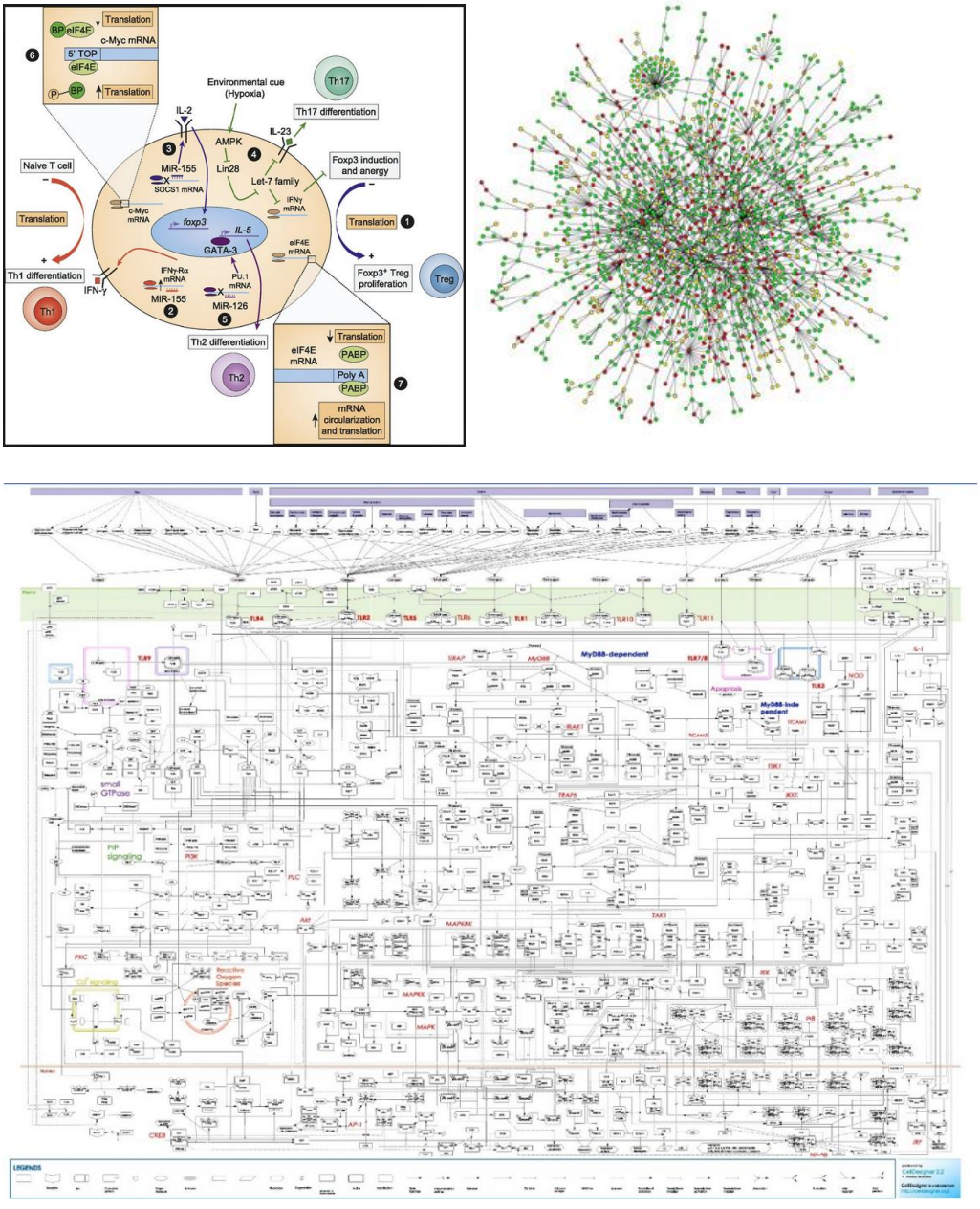


Figure 2 : Different types of intra-cellular networks.

In this work, we incorporate in our models the elements that belong to all three types of intra-cellular networks, including receptor activation, receptors' downstream protein-protein interactions that lead to activation of transcription factors and gene regulations, post-translational modifications, and metabolic regulations that affect cellular processes such as apoptosis.

2.1.1 Biological interactions

In this section, we list the most common biological motifs that we encountered while developing models of intra-cellular networks, and we show their computational representation in the form of ordinary differential equations.

- **Activation**

  An activation of an element can represent two phenomena:

  (a) an increase in element's *amount* in the cell

  (b) an increase in element's *activity* with respect to other cellular elements it acts upon.

  If element *A* is activated by element *B*, this can be represented as:

$$B \xrightarrow{k} A$$

  We can also assume that in some cases *k* may be given to represents the rate at which the activation of *A* by *B* occurs. The change in element *A*'s amount/activity with respect to the amount/activity of *B* can then be computed as:

$$\frac{d[A]}{dt} = k[B] \tag{1}$$

  Note that we use $[X]$ to represent the amount or activity of element *X*.

- **Inhibition**

  An inhibition of an element can represent two phenomena:

(a) a decrease in element's amount in the cell

(b) a decrease in element's activity with respect to other cellular elements it acts upon.

If element *A* is inhibited by element *B*, this can be represented as:

$$B \dashv A$$

Similar to activation, we can also assume that a *k* is given that represents the rate at which the inhibition of *A* by *B* occurs. The change in element *A*'s amount/activity with respect to the amount/activity of *B* can then be computed as:

$$\frac{d[A]}{dt} = -k[B] \tag{2}$$

- **Binding/Unbinding**

Molecular binding is an attractive interaction in which two molecules come together to form a protein complex. After binding, the two molecules will have different properties than they had before. A protein complex can be classified as either a reversible or irreversible covalent bond. If the bond is reversible then the reverse action is called unbinding. The following equations show a reversible binding between elements *A* and *B* to form complex *D*. We can assume that k_1 represents the rate at which *D* unbinds, and k_2 the rate at which *A* and *B* bind.

$$A + B \underset{k_1}{\overset{k_2}{\rightleftarrows}} D$$

The changes in amount/activity of elements *A*, *B* and *D* can be computed as:

$$\frac{d[A]}{dt} = k_1[D] - k_2[A][B] \tag{3}$$

$$\frac{d[B]}{dt} = k_1[D] - k_2[A][B] \tag{4}$$

$$\frac{d[D]}{dt} = -k_1[D] + k_2[A][B] \tag{5}$$

- **Phosphorylation**

The addition of a phosphate group to a molecule is called phosphorylation. When phosphorylated, the properties and behavior of the element change with respect to the unphosphorylated state. For example, it may bind to other molecules at a lower rate, or not bind at all. Phosphorylation can be cooperative, when additional phosphorylation events make it easier to phosphorylate, or uncooperative, when additional events make it harder to phosphorylate. If element *A* can be phosphorylated two times, this can be represented as:

$$A_u \underset{k_2}{\overset{2k_1}{\rightleftarrows}} A_p \underset{2k_2}{\overset{k_1}{\rightleftarrows}} A_{pp}$$

We can compute the rate of change of the unphosphorylated A_u, the rate of change of the phosphorylated A_p, and the rate of change of A_{pp} that is twice phosphorylated, as follows:

$$\frac{d[A]_u}{dt} = -2k_1[A]_u + k_2[A]_p \tag{6}$$

$$\frac{d[A]_p}{dt} = -k_1[A]_p - k_2[A]_p + 2k_1[A]_u + 2k_2[A]_{pp} \tag{7}$$

$$\frac{d[A]_{pp}}{dt} = k_1[A]_p - 2k_2[A]_{pp} \tag{8}$$

There are also other types of interactions, as described at the beginning of Section 2.1. In this work, we focus mainly on the interactions that we detailed above.

2.1.2 Reaction rule-based modeling

Rule-based modeling is an approach that uses a set of rules to represent reactions. These rules can then be translated into Markov chains or differential equations for simulation of the system. To establish a rule-based model we need to know the interactions of the model components and have information about the kinetic laws. A powerful feature of this type of modeling is the ability to include current states and potential bindings for each model element. In the case of phosphorylation, this feature allows for easy tracking of every phosphorylation state of an element. A widely-used simulator is BioNetGen [16]. The binding reaction is presented in Figure 3.

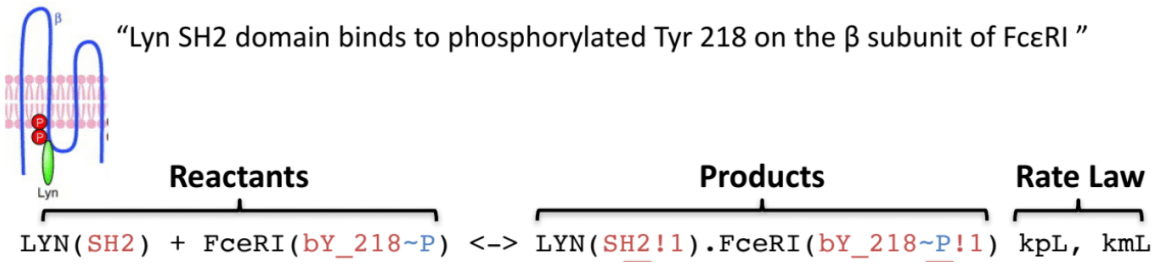


Figure 3 : Rule representing a binding reaction using BioNetGen Language [13]

2.1.3 Logical Modeling

The biologists often draw networks that they are studying as graphs, like the one shown in Figure 4, where nodes represent model elements, and edges represent the interactions between elements. Besides being directed, the edges also indicate the polarity of regulation, positive (regular arrows in Figure 4) or negative (blunt arrows in Figure 4).

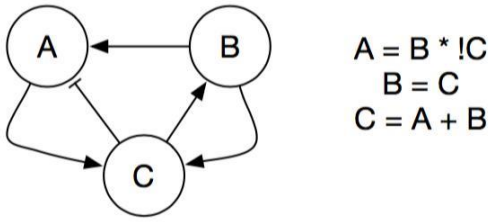


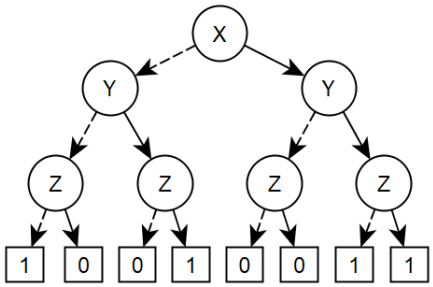
Figure 4: A toy example of a logical model. '+' represent an OR logical operator, '*' represents AND operator, and '!' represents a NOT operator.

Table 1 : Truth table for AND, OR, NOT

A	B	A AND B	A OR B	NOT A
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0

2.2 BINARY AND ALGEBRAIC DECISION DIAGRAMS

Binary Decision Diagrams (BDDs) are a data structure that is used to represent a Boolean function. Figure 5 shows a binary decision tree without the reductions rules. A solid line represents an edge when the variable is high (equals one) and the dotted line represents an edge when the variable is low. This representation is useful to analyze the importance of a variable. For example, when X=1 and Y=1, the value of Z does not matter and the outcome will always be F=1.



X	Y	Z	F
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Figure 5 : A BDD and a truth table for the function F(X,Y,Z)= !X*!Y*!Z + X*Y + Y*Z

This conclusion leads to the second type of a decision tree that is now commonly used as a more optimized version of a decision tree: Reduced Order Binary Decision Diagram (ROBDD). As discussed above, the diagram can contain useless or redundant nodes that do not carry any additional information. Therefore, we can eliminate them to create an ROBDD. The figure below shows the ROBDD for the function F declared above.

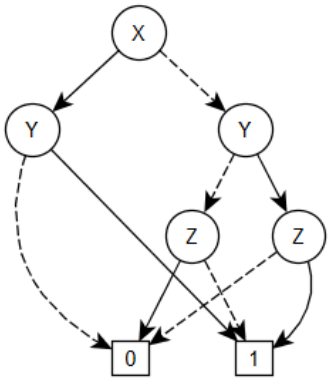


Figure 6 : Reduced Order Binary Decision Diagram for the example function in Figure 5.

Now we can clearly see that when X=1, the value of Y will determine the value of F. Binary Decision Diagrams (BDDs – used as a short for ROBDDs) [20][21] have significantly changed the landscape of synthesis, formal verification, and testing of digital circuits. BDDs provide an efficient and canonical form of representation of those functions. Despite being reduced to the

simplest structure, conventional algorithms such as breadth-first search, do not do well on large graphs, simply because they process vertices and edges on an individual basis. Several examples of successful applications of symbolic graph algorithms in reachability analysis of finite state machines have been reported. All these applications are topological; they address problems where connectivity is the principal issue. However, a follow-up work [22] introduced a class of algorithms for arithmetic symbolic computation, based on a new kind of BDDs called a Multi-Terminal Binary Decision Diagram (MTBDD), whose main feature is the adoption of multiple terminal nodes. One application of these MTBDD are Algebraic Decision Diagram (ADDs), which provide an efficient means for representing and performing arithmetic operations on functions from a factored Boolean domain to a real-valued range. The ADDs can also be reduced and ordered, thus in this work ADD will refer to ROMTBDD.

3.0 MODELING BIOLOGICAL EVENTS USING DISCRETE APPROACH

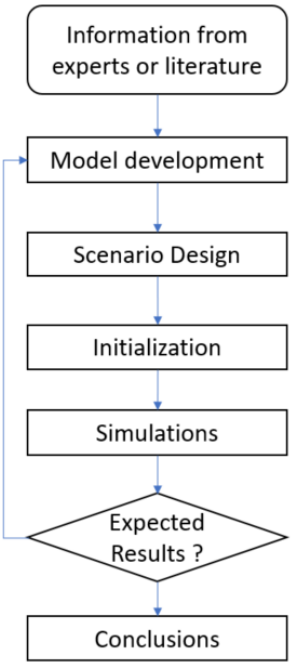


Figure 7 : Flow diagram for our model assembly and analysis methodology.

The development and analysis of a discrete model follows the flowchart in Figure 7. Model development relies on the approaches discussed in Section 2.1, which are also further expanded in our modeling approach, as will be discussed in the following sections. In short, we decide the number of discrete values that each element can take and create update rules for all model elements.

Model analysis starts by identifying or designing an analysis scenario, which guides initialization of all model elements, external model stimulations, or internal perturbations, and the choice of simulation scheme (deterministic or stochastic) [25]. Depending on the simulation scheme, one or more elements can be selected simultaneously and updated according to their corresponding update rules. The choice of elements to be updated in each simulation step can be

either deterministic or random. We obtain simulation traces for all model elements, and in the case of random update scheme, we also compute average trace for each element across multiple simulation runs. Finally, we plot all or selected element traces to conduct the analysis and guide model updates if necessary.

In this work, we approach model analysis from two different perspectives. In the first case, we are using a model from literature, with the goal to reproduce the results of that model using our modeling approach, and to compare and contrast modeling approaches. In the second case, we focus on new scenarios, not previously explored. For example, we explore the influence of a reaction on a pathway, or on the overall model. To do so, we start by turning OFF this particular reaction, and if we do not observe any changes in model's behavior, we use our model and simulation results to explain this observation. If the absence of the reaction changes simulation result, then we further explore the reasons for the influence of this reaction, and the overall model sensitivity to the reaction.

3.1 DISCRETE APPROACH

In this work, we extended beyond commonly used Boolean modeling approach, which assumes variables that have only 0 and 1 values, to discrete modeling approach. The limited number of possible element values in the Boolean modeling approach leads to a lack of precision and can even introduce some errors. For example, let us assume that an element C can be activated by element A and inhibited by element B. The truth table associated with this regulatory rule is shown in Table 2. From left to right, increase in A and B value, from top to bottom increase in C value. If the cell is blue, this indicates that the value of C at the next step is minimal (0 in both Boolean

and discrete cases), if the cell is green, the value of C at the next step will be intermediate, and if the cell is red, the value of C at the next step will be maximal (1 in Boolean and 2 in discrete case).

Table 2 : Example truth tables: (left) Boolean variable C; (right) discrete variable with three values.

C\AB	00	01	10	11
0				
1				

C\AB	00	01	02	10	11	12	20	21	22
0									
1									
2									

Table 3 : Percentages comparison of the two cases in Table 2.

	0-1	0-2
Decrease	3/8 -> 37.5%	6/27 -> 22.2%
Stay	4/8 -> 50%	15/27 -> 55.5%
Increase	1/8 -> 12.5%	6/27 -> 22.2%
	6/8 -> 75%	9/27 -> 33.3%
	0/8 -> 0%	9/27 -> 33.3%
	2/8 -> 25%	9/27 -> 33.3%

We can quantify the lack of precision and errors introduced when working with 0-1 values versus 0-2 values, as shown in Table 3. One could encode all discrete model elements with Boolean variables only, which would in the case of our example mean that each element A, B and C is represented with two Boolean variables (e.g., A is represented with A_{high} and A_{low}). With the Boolean approach and the truth table in Table 2, the equation for C_{high} (using the same notation as in Section 2.1.3) is:

$C_{low}*C_{high}*(!A_{low}*!A_{high}*!B_{low}*!B_{high} + A_{low}*!A_{high}*!B_{low}*!B_{high} + A_{low}*!A_{high}*B_{low}*!B_{high} + A_{low}*A_{high}*!B_{low}*!B_{high} + A_{low}*A_{high}*B_{low}*!B_{high} + A_{low}*A_{high}*B_{low}*B_{high}) +$

$C_{low}*!C_{high}*(A_{low}*!A_{high}*!B_{low}*!B_{high} + A_{low}*A_{high}*!B_{low}*!B_{high} + A_{low}*A_{high}*B_{low}*!B_{high}).$

The logic functions become even more complicated if we are representing discrete variables and their update functions using the Boolean approach. An increase in the number of regulators leads

to a further increase in the complexity of logic rules, and thus, when modeling networks such as the RAS signaling network that has more than 200 elements, a development of a more practical method is needed.

In this work, we use algebraic operations instead of Boolean operations. To this end, we define a "discrete OR" operation on three-valued variables, e.g., C = A OR B, as shown in the truth table in Table 4.

Table 4 : A truth table for the "discrete OR" operation.

AB	00	01	02	10	11	12	20	21	22
C	0	1	2	1	1	2	2	2	2

The Table 4 shows that the OR operator when using 0-2 can be replaced by the operator MAX. The analysis of other Boolean operators allows a similar translation. For example, the operator AND can be translated to MIN.

3.2 SPONTANEOUS INCREASE AND DECREASE

One key feature that had to be implemented is the spontaneous increase/decrease behavior feature. Let us take a simple example of element A activating element B. When element A is present, the value of element B is increasing. Now we turn off A with injection of a drug, and therefore, A is no longer present. In a real system, B can decrease with a biological event called *spontaneous decrease*. With no presence of an activator, B return to a baseline level. In our modeling approach, before the implementation of this feature, B would stay at a high level. This missing feature was

the biggest source of error from model simulations, because without this feature, the elements were reaching a wrong steady-state.

3.2.1 Example in the RAS model

In the RAS model, one example of spontaneous decay is shown with the elements RALA, B and EXOC1-8. As can be seen from Figure 8, the only regulations of EXOC1-8 is an activation from RALA and RALB.

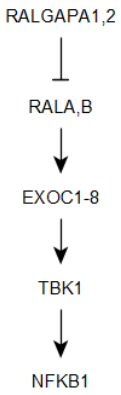
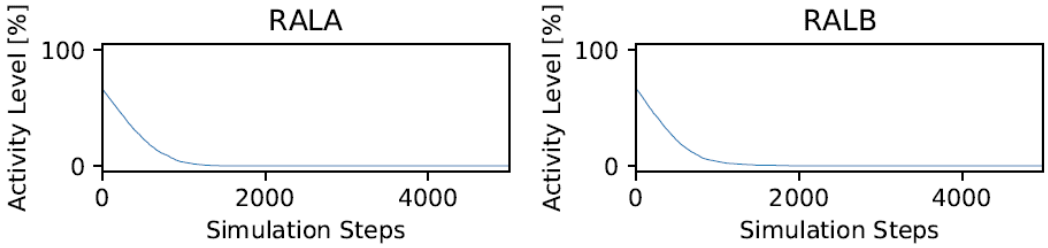


Figure 8 : A pathway example used to illustrate the spontaneous behavior.

The results of simulation for elements RALA, RALB, EXOC2, EXOC3, before extending our modeling approach to include the spontaneous behavior, are shown in Figure 9.



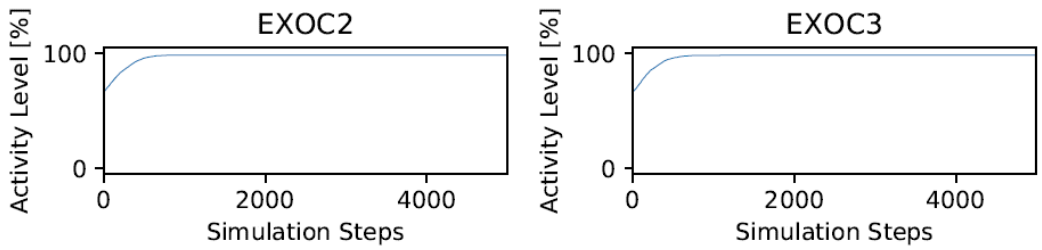


Figure 9 : Traces of RALA, B and EXOC2,3 without the spontaneous decay implementation.

We can observe from Figure 9 that RALA, B is going down and is not present anymore after 2000 steps. However, EXOC2,3 which is only activated by RALA, B is still at max level after step 2000. As explained in the introduction of this subsection, the behavior of EXOC2,3 is not the one we should expect in biology. Thus, the simulation results improve after implementing the spontaneous decay, as shown in Figure 10.

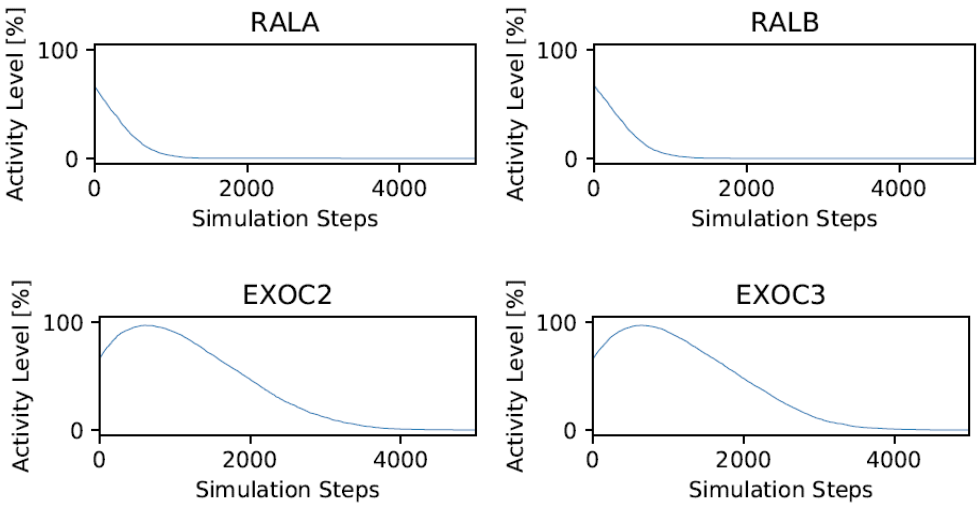


Figure 10 : Traces of RALA, B and EXOC2,3 with the spontaneous decay implementation.

Without implementing the decay the branch of the Ras pathway shown in Figure 8 would not be able to oscillate unless we use negative autoregulation on every element. This solution would have produced the expected results but would be biologically incorrect. Now, if RALA, B oscillates then the elements downstream are going to oscillate too, depending on the duration of

time interval where RALA, B stay at level zero, the decay in EXOC2,3 varies. The whole RAS pathway has an oscillatory behavior, and therefore, implementing the spontaneous increase/decrease to allow for oscillations in the model was critical to accurately capture this system behavior.

3.2.2 Approach to modeling spontaneous increase/decrease

The solution adopted is to use a counter that starts when the score of an element's activators (inhibitors) is zero. Given that not all reactions happen at the same rate, we can use a parameter to specify how many steps to wait before the spontaneous decrease (increase). In the example used above, the counter would start when RALA and RALB are equal to 0. If one of the two elements are not 0, then the counter will not start. At every step, one element is updated and the counter will only increase when the element is picked. For example, if we have 100 elements with a uniform probability distribution and we specified 1 step for decay, then we would need an average of 100 steps for the element's value to decrease by one. If the number of steps specified is greater than one, then the counter will reset if the conditions are not met. Furthermore, if we use a much larger network such as the one describe later in 5.3 and shown in Figure 25, which contains 230 elements, and the value range is from 0 to 9, when EXOC2 starts to decay it will need 9*230 =2070 steps approximately to reach 0.

The use of the spontaneous increase/decrease feature implies being careful when introducing more variable to represent elements such as dimers. The extra variable used to represent the dimers will slow down the decay downstream, as it will require the variable representing the dimers to be zero before the element downstream can decay whereas the decay can start as soon as the element is 0 if not using an extra variable.

3.3 OSCILLATIONS

As stated briefly in the introduction, the behavior of the entire RAS signaling pathway is oscillatory. This behavior is mostly due to the activation of EGFR by EGF and its spontaneous deactivation. The work from [11] established a correlation between the trace of the concentration of EGFR over time and the elements downstream of it. This implies that most of the oscillations are due to spontaneous behavior. However, positive or negative feedback from elements downstream will change the characteristics of these oscillations such as rising time or amplitude.

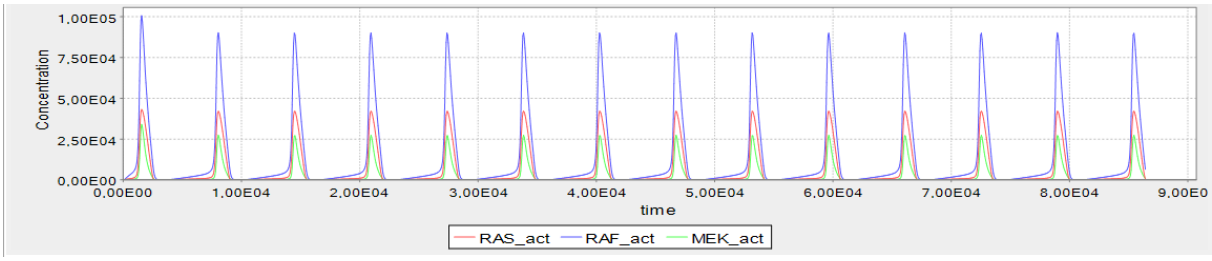


Figure 11 : Concentration (moles) of activated RAS, RAF and MEK over time (seconds)

3.3.1 Spontaneous behavior

Once the spontaneous behavior was implemented as described in the previous sections, we can see oscillations in the pathway. For example, let us use Figure 12 and remove all the feedback.

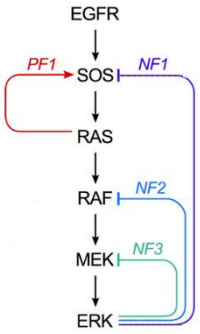


Figure 12 : Graph of the feedback in the BioNetGen model [11]

With the simplest model containing only activations, and with a manual oscillation of EGFR (deactivation at step 150 during 150 steps), the traces of the six elements in Figure 12 are the following:

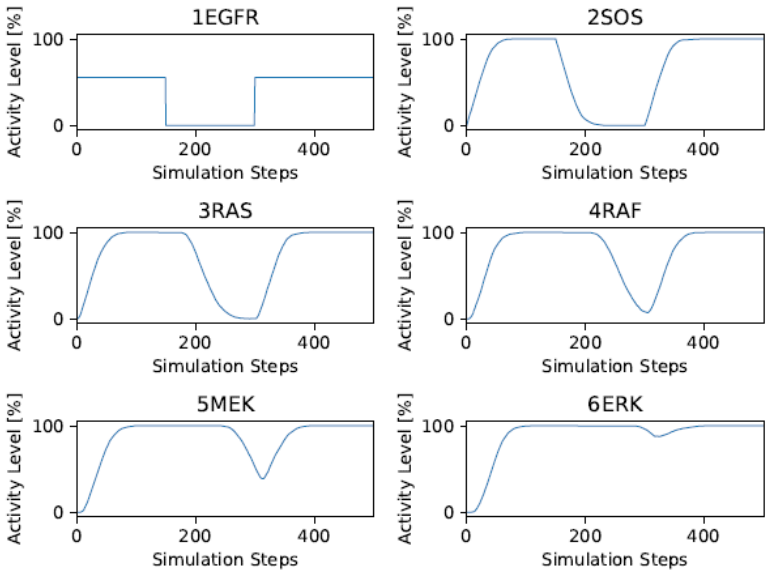


Figure 13 : The trajectories for pathway elements assuming oscillations in EGFR only.

With the previous explanation of the spontaneous behavior mechanism and its implementation, we analyze the plots in Figure 12, and we can observe the propagation of the decay through the signaling pathway.

3.3.2 Feedbacks

All the feedbacks presented in Figure 12 have a specific role. The positive feedback from RAS to SOS allows for signal amplification. Negative feedbacks emanating from ERK have been associated mainly with response attenuation, but negative feedbacks in general may be tackled to ensure perfect adaptation or give rise to oscillations [11]. For example, sustained oscillations may

arise when a negative feedback loop is embedded within a relatively slow positive feedback loop or when a positive feedback loop is embedded within a relatively slow negative feedback loop. The time constant of the negative feedback from ERK to SOS is 3e-9, and the positive feedback has a time constant of 1e-6 or 1e-7, depending on whether RAS is binding to SOS GTP or SOS GDP, respectively. Therefore, in this pathway we have a fast-positive feedback from RAS encapsulated in a slower negative feedback from ERK. When using only the feedback in the model, without the oscillations in EGFR, we obtain the result shown in Figure 14.

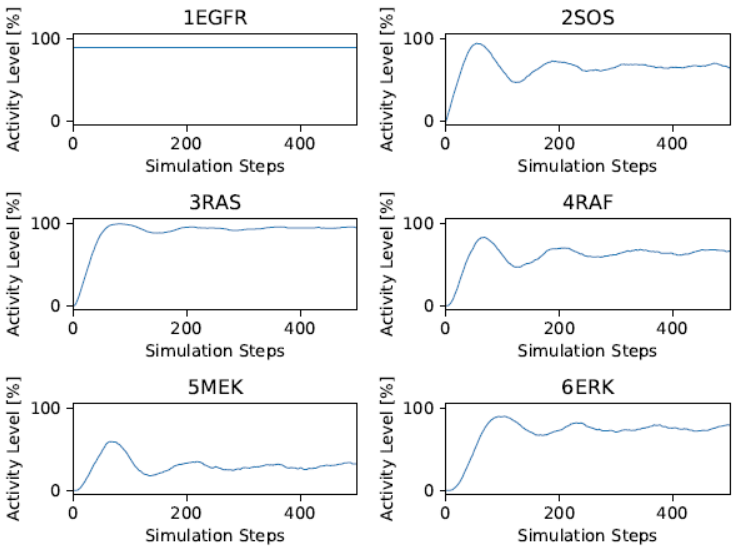


Figure 14 : The trajectories for pathway elements assuming negative feedbacks only.

As can be seen from Figure 14, the oscillations in EGFR were not manually enforced, however, for elements downstream of EGFR we can still observe oscillations around a mean value before they reach steady state. Finally, it is the combination of both the feedback and the spontaneous decay due to the oscillation of EGFR that will produce more complex results.

3.4 PHOSHPORYLATION

3.4.1 Phosphorylation in BioNetGen model

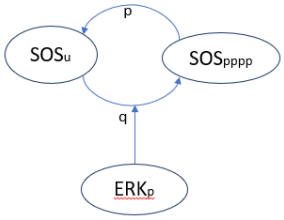


Figure 15 : Phosphorylation of SOS by ERK.

The model from [11] includes a biological event that is not implemented in our modeling approach. This event is the phosphorylation of an element. In Figure 15, we can observe that SOS can be phosphorylated up to 4 times and it can only bind to RAS when it is unphosphorylated. Therefore, unphosphorylated SOS can be considered as active, whereas phosphorylated SOS (one, two, three or four times) can be considered as inactive. Secondly, given the reaction rates q=3e-9s and p=3e-4s, the speed of the overall reaction is unbalanced as the unphosphorylation happens faster than the phosphorylation. Besides, the phosphorylation is not cooperative: when SOS changes from the unphosphorylated to the single phosphorylated form, this is happening faster than transforming SOS from the 3-phosphorylated to the 4-phosphorylated form. In the translated model, we will use delay to represent this difference in the kinetics of phosphorylation.

3.4.2 Implementation of phosphorylation in discrete modeling

Our first goal here was to develop a representation for different phosphorylation states of an element. So far, one variable was used to represent an element. If its activity was 0, then it is

inactive, otherwise it is considered as active. Let us define SOS= {0,1,2,3,4}, the active state of SOS is represented as value 4, phosphorylated one time is represented as value 3, phosphorylated two times is represented as value 2, phosphorylated once is represented as value 1, and finally, unphosphorylated is represented as value 0. Thus, the truth table for RAS, which is activated by the binding of SOS unphosphorylated and EGFR active, can be written as shown in Table 5.

Table 5 : Truth table for RAS

EGFR	SOS	RAS
0	0	0
0	1	0
0	2	0
0	3	0
0	4	0
1	0	0
1	1	0
1	2	0
1	3	0
1	4	1

Although this method is technically possible, it is not suited for a large model, as it would require significant manual (not automated) work. Using more variables is a necessity to represent multiple phosphorylation. However, as we stated at the beginning of this thesis, one of the goals of our work and the discrete modeling approach is to be able to use abstraction to overcome the issues of incomplete or uncertain information. If too many details are introduced then there is no advantage to use this method compared to ODEs, except the speed of simulation. Therefore, using one variable for each state of an element was ruled out and we introduced two variables to represent the different states: One for the inactive/phosphorylated state and one for the unphosphorylated/active state. A high activity of the phosphorylated variable would represent a

highly phosphorylated state of the element. In Figure 16, we show the behavior of the two variables that we introduced to represent different phosphorylation states of SOS.

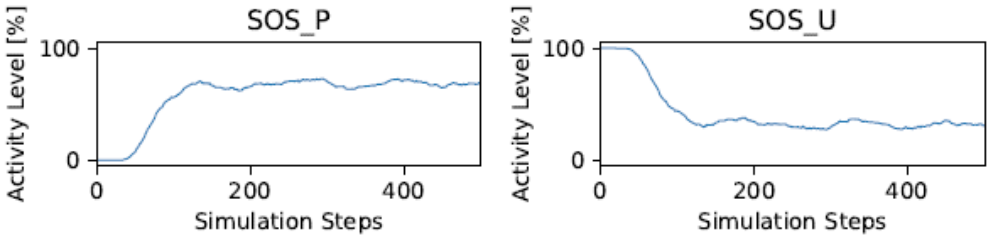


Figure 16 : Trajectories for the two SOS variables: phosphorylated (left) and unphosphorylated (right).

As can be seen from Figure 16, we always have a small activity of unphosphorylated SOS ready to bind, and the average of phosphorylation for SOS is the 3-phosphorylated state. This result was expected due to the rates of the reactions: the unphosphorylation happens faster than the phosphorylation, therefore always having a small amount of unphosphorylated SOS is correct.

4.0 IMPLEMENTATION OF REACTION RULE-BASED MODELS USING BDD AND ADD DATA STRUCTURES

As explained in Section 3.1, Boolean simulation was extended to discrete simulation thanks to the use of mathematical operations to translate Boolean operator. In Section 2.2, we explained the structure of an ADD and we saw that it does not have weight on its edges. However, in a biological context we can determine whether an element is high/present or low/absent. Therefore, adding a probability for each node to be equal to one would be equivalent to include the element activity. With this property added to this structure, ADD can now be used in two different ways: first, it can translate biological reaction, and second, it can be used to perform short term analysis of a node through simulations.

4.1 REACTION TRANSLATION USING ADDS

In this section, we will describe our approach to use ADDs to represent the reactions that were defined in Section 2.1.1: activation, inhibition, binding/unbinding and phosphorylation. We illustrate our approach using small examples with elements A, B, C.

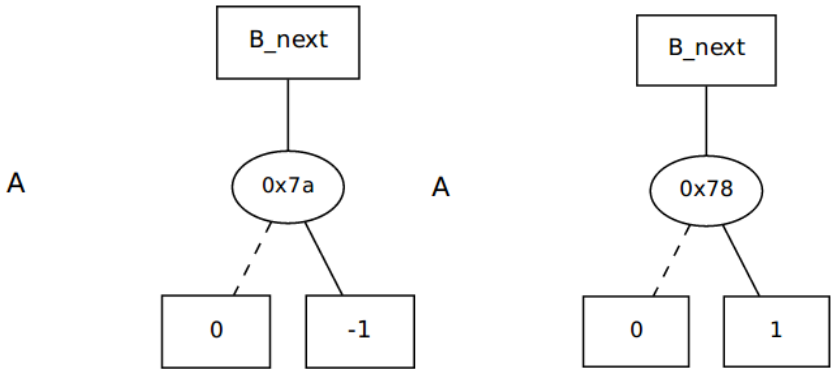


Figure 17 : ADD for the (left) Inhibition rule and (right) Activation rule.

In Figure 17, we show a simple example of this data structure to represent biological reactions. The terminal of the tree represents a change in B activity at the next step: in the left tree, A is inhibiting B, therefore, if A is present, the activity of B will decrease at the next step. But this structure can be used to represent more complex reaction such as unbinding. From Figure 18, we can see that more precision can be achieved on this structure by including a constant rate. In Figure 18 (left), we illustrate the representation of unbinding without including a rate constant, while in Figure 18 (right), we show an ADD with a rate constant of 0.2 for unbinding. Elements A and B bind to create element C. Unbinding, which is the reverse reaction, leads to the decomposition of C and elements A and B are recovered. So, if C is present, meaning we are looking at the true edge, it can unbind. Without the rate constant value in the terminal node, it would mean that all of C is unbinding, so it could be applied to single cell modeling. With the rate value, only a portion of C is unbinding and the value of C_next is 0.8 instead of 0.

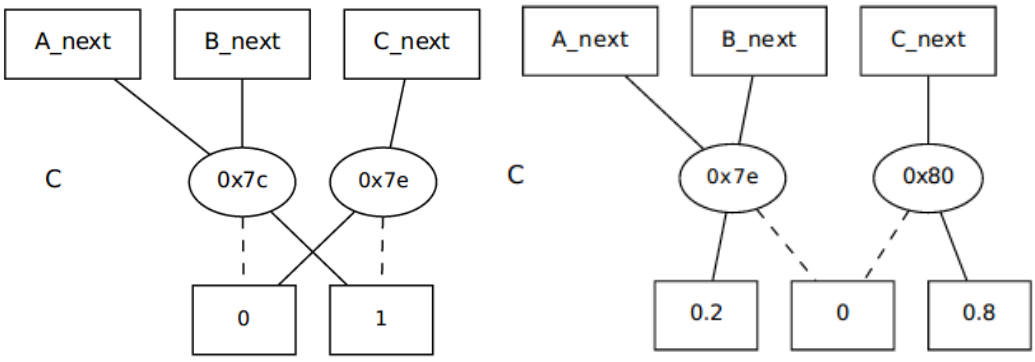


Figure 18 : ADD for the Unbinding rule: (left) with rate constant,(right) without rate constant

Figure 17 and Figure 18 show how we can use mathematical operations to translate biological events. In the case of more complex biological motifs, this structure enables fast translation from reaction rule-based approach to element rule-based approach and it also facilitates validation of the discrete modeling approach.

4.2 SHORT TERM ANALYSIS

Additional application of the ADD structure is in model analysis. The implementation of discrete or logical models using ADDs is straightforward, and therefore ADDs can speed up the analysis of models created using the discrete or logical modeling approach.

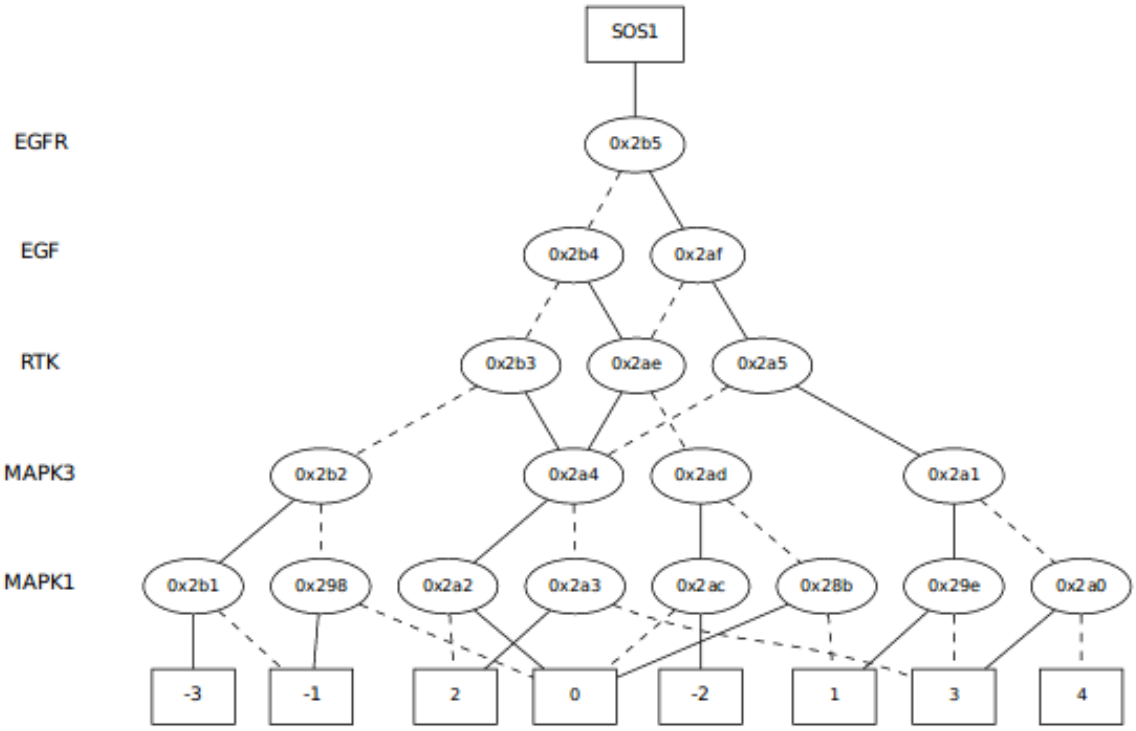


Figure 19 : Representation of SOS1 update rule using an ADD.

With the introduction of probabilities to represent the activity/amount of each variable present at a certain step, we can use this structure to determine what behavior will SOS1 have in the next step. The update rule of SOS1 is EGFR*EGF + RTK – MAPK3*MAPK1. We use two scenarios here. The Scenario 1 contains a high value of RTK which will allow SOS1 to increase. The Scenario 2 shows that when RTK is turned down, SOS1 will not have strong enough activators to increase.

Table 6 : Probabilities of elements

Probability	EGFR	EGF	RTK	MAPK3	MAPK1
Scenario 1	0	0.33	0.7	0.1	0.7
Scenario 2	0	0.33	0.1	0.1	0.7

In Table 6, we present the probabilities for each element in the SOS1 update rule to be equal to 1 in the two scenarios. Using these probabilities, an algorithm will compute the probability

to reach every final value. Table 7 shows the total probability for every terminal value in the two scenarios. If the highest probability is negative, then SOS1 will decrease in the next step. If the probability is 0, SOS1 will stay constant, and if the highest probability is positive then SOS1 will increase. The yellow cells represent the highest probability computed in the two scenarios. These results are confirmed by the simulation traces shown in Figure 20, for Scenario 1 (left) and for Scenario 2 (right).

Table 7 : Total probabilities for every element value.

Probability	-3	-2	-1	0	1	2	3	4
Scenario 1	0.01407	0.006930	0.132660	0.168610	0.026730	0.462000	0.189000	0
Scenario 2	0.042210	0.020790	0.397980	0.365830	0.080190	0.066000	0.027000	0

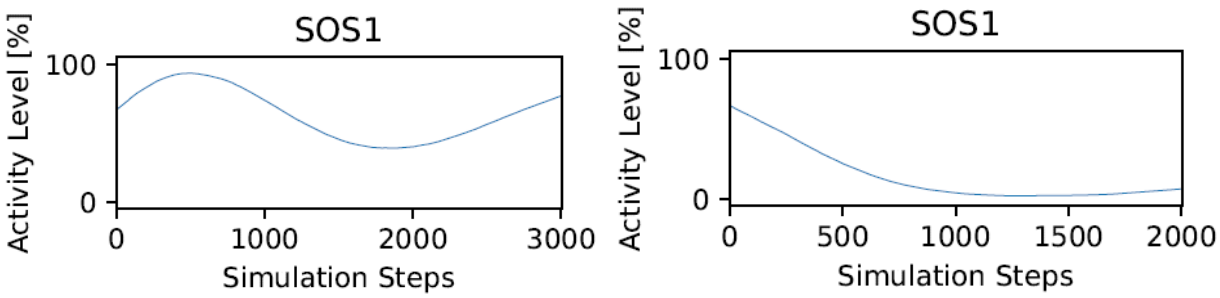


Figure 20 : Traces of SOS 1 in Scenario 1 (left) and Scenario 2 (right).

These examples show that this method is yet limited to short term analysis because once a feedback occurs (around 500 steps in Scenario 1), then SOS1 starts to decay, and the probability of decay is not high enough. Finally, since the ADD structure is automatically reduced, we can aim to use this structure to find simplification in update rules.

5.0 MODELING OF RAS PATHWAY IN CANCER CELL

To model the RAS pathway, we used different paper and data with different aim. So far, four models were used:

(1) A reaction rule-based model written using the BioNetGen language [11], which reproduced oscillations observed in MAPK signaling pathway, but is lacking the dimerization of RAFs.

(2) A model translated from (1) into our discrete modeling approach to determine if our approach can capture the pathway characteristics.

(3) An extended version of the model in (2) that includes more elements than the model in (2) to evaluate the impact of a bigger scale network.

(4) A model created using our discrete modeling approach, following a graph from [24].

The last model listed was the first model created in this work, however, the graph from [24] is unclear, leading to uncertainties. Therefore, to unveil some of these uncertainties, the use of literature and previous modeling work was needed. The analysis of the BioNetGen model, targeting only the key elements of this pathway, helped to get a better comprehension of this signaling pathway.

5.1 REACTION RULE-BASED MODEL OF RAS PATHWAY

We decided to investigate the model from [11], that is, to start from this smaller model of a single pathway, and then expand it later to include other pathways, instead of trying to validate a large model with more than a thousand element interactions. The reaction rule-based model from [11]

is using numerical solution of ODEs to determine the concentrations of every element in the model. As discussed earlier in this thesis, such approach can ideally provide precise numerical values but it requires a lot of knowledge. For example, this detailed approach describes the mechanisms of reactions at the beginning of the pathway, however, the dimerization of RAFs, which are activating the elements downstream from Ras, is not included due to the lack of knowledge about this mechanism. Therefore, this model would not be useful to test hypothesis about RAFs. However, when looking at the rules, we can extract a few characteristics that could be relevant for the other models.

Besides the simplification of RAF's mechanism, another downside of this model is its poor extensibility. As stated in the introduction, continuous models need a lot of information that is not all available for this pathway. The objective of our work is to model the entire network that includes not only the pathway shown in Figure 12, but also other key regulators of this pathway, as well as its downstream effectors. The interaction map of this network is shown in Figure 25, and is adopted from [24]. Furthermore, we are interested in including RAF dimerization, given that the previous work investigating this mechanism has shown interesting behavior on the pathway [26], and our goal is to reproduce this behavior in simulations. That implementation was not possible using the reaction rule-based modeling, while we were able to implement it using the discrete modeling approach.

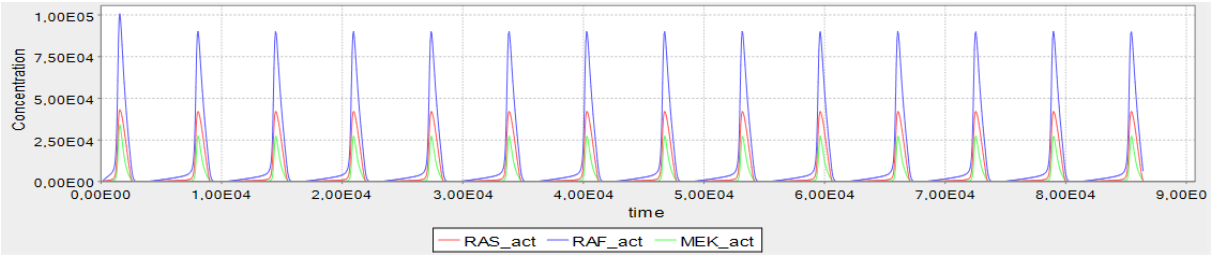


Figure 21 : Concentration (moles) over time (seconds) of RAS, RAF and MEK.

In Figure 21, we show results for RAS, RAF and MEK, obtained from the model in [11], using BioNetGen tools [16]. Two important conclusions can be drawn from this graph. First, we can see that the amplitude of the peak is different for every element (Only RAS, RAF and MEK are included for a better visibility) with the concentration of active RAF, twice superior than the concentration of active RAS, but RAS and MEK have the same magnitude. The other characteristic is the speed of the signaling pathway, which is rapid, as we can barely distinguish the time for the peak of RAS (the most upstream of the signaling pathway between these three elements), and the peak of MEK (the most downstream in the pathway). Therefore, every model that does not show these two characteristics needs to be refined.

5.2 TRANSLATED MODEL FROM BIONETGEN TO DISCRETE VARIABLES

After finalizing the analysis of the previous model, we created a discrete model based on it. The creation of this discrete model, and the corresponding analysis have been beneficial in the process of understanding and improving the discrete modeling approach. We have developed a method to achieve oscillations when using one variable per element vs. multiple variables per element, and a method to incorporate into the model the delay and spontaneous increase/decay in order to reproduce the cooperative/uncooperative behavior of the reactions.

In the BioNetGen model, EGFR is oscillating between active and inactive state, with a fast increase to a high concentration (almost instantaneous), a brief time at the highest concentration and then decaying, so active EGFR is present approximately 25% of the duty cycle. By adding a toggle feature on our simulator, we can simulate pulse of active EGFR with the frequency that we desire. When testing different scenarios, the result is that the oscillation of EGFR is necessary for

the entire pathway to oscillate, and the speed of this oscillation determines the speed of the oscillations in the pathway: Even when slowing down the reactions of other elements, such as the dephosphorylation of SOS, the cycle between RAS-GDP (inactive) and RAS-GTP (active) seems to be determined by EGFR.

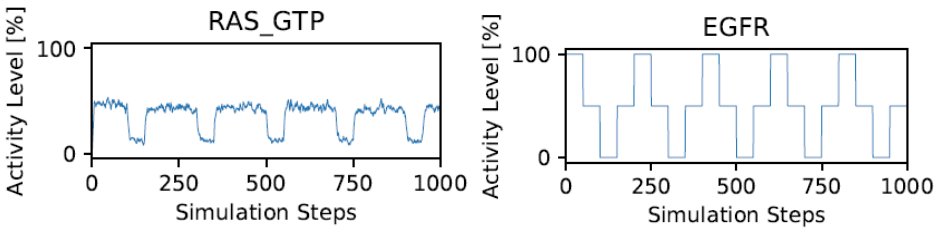


Figure 22 : Activity of RAS_GTP with active EGFR 80% duty cycle.

As can be seen from the simulation results shown in Figure 22, if EGFR activity is not zero, we observe a medium activity level of active RAS. When EGFR is removed, we have this fast decay and active RAS is dropping down to 10%. This shows that a high activity of EGFR is not required to activate RAS. Since high EGFR is not required, our signal was equivalent to 80% active and 20% inactive which is the opposite of the model in Section 5.1. Therefore, EGFR signal was modified to 200 steps inactive, 50 steps active, which is equivalent to 80% inactive and 20% active.

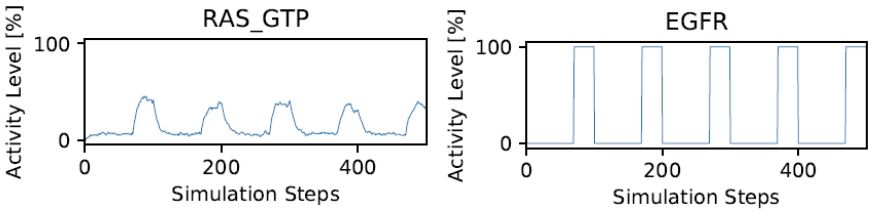


Figure 23 : New EGFR signal, and the RAS response.

We can observe from Figure 23 that the behavior of active RAS is the one expected, as it is almost 0 when EGFR is not present. To accentuate the peak waveform, the only parameter to play with is the delay, which is closely related to the kinetics of the reactions. These constants

were defined in the rule-based model and were used in our discrete approach to increase the number of steps required for a change in an element's value. The results below are even closer to the real observed behavior because RAS does not stay at a steady low level for an extended time. RAS is rising fast and then decaying like in the model in Section **Error! Reference source not found.**. We also observe in Figure 23 that the RAS activity stagnates for a couple of steps before decaying, and that the RAS magnitude is increased.

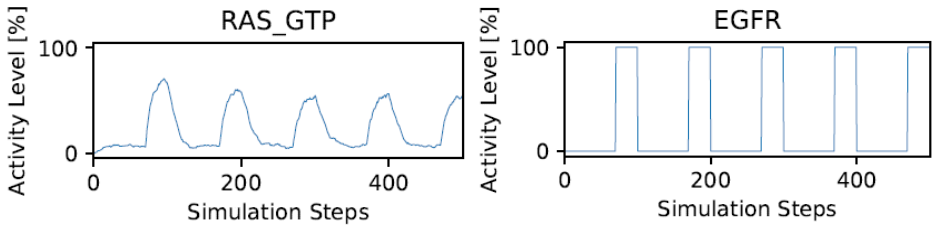


Figure 24 : The same EGFR signal as in Figure 23, with increased delay in the model.

We can conclude from the discussion in this section and the results, that the characteristics of the model described in Section **Error! Reference source not found.**, which uses ODEs, were s uccessfully reproduced using a discrete model. However, this required a lot of manual work and analysis. Therefore, our goal is to develop a unique representation for biological events that we described in Section 2.1.1, that can accurately incorporate these mechanistic events into a discrete model. A promising data structure that allows us to develop such representation are binary decision diagrams.

5.3 MAIN MODEL

The main model that we created in this work was developed following the interaction map shown in Figure 25.

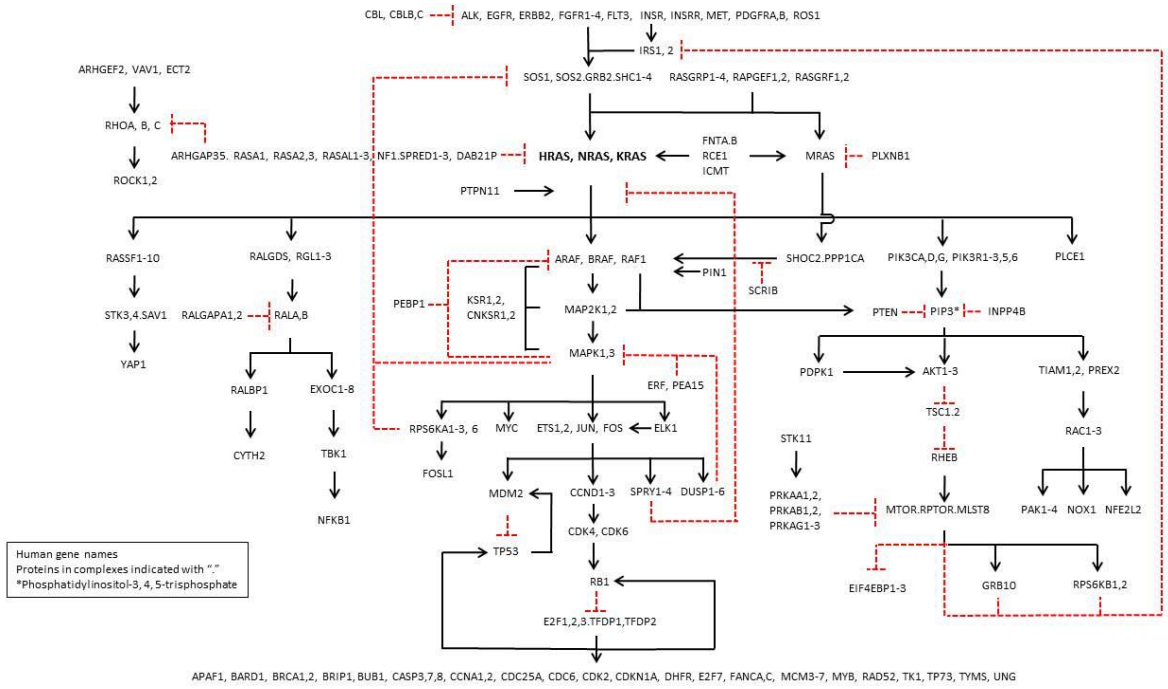


Figure 25 : Graph used to create our main model [24]

The first step in the development of this larger model was to determine the correct interpretation for a number of interactions shown in the graph in Figure 25, which contains many uncertain connections. We had to make a number of assumptions when developing element-based rules for the model. For example, MAPK1,3 is joining PEBP1 to inhibit ARAF, BRAF and RAF1. This edge can have different interpretations: It could be represented as an AND relation between MAPK1,3 and PEBP1, which means that both elements need to be active to inhibits every RAF. However, we found out that PEBP1 is targeting only RAF1 and that the relation between MAPK1,3 and PEBP1 is an OR relation. Additionally, we can use the available databases, such as

Universal Protein Resource (UniProt) [5], which is a freely accessible database of protein sequences and functional information, to check the interactions between some elements. As our first approach, and due to the fact that the graph contains 229 elements, we tried to reduce the size of the model by grouping the elements that seemed redundant. For example, SHC1-4 have the same relations (activating RASs), such that they could be gathered into a single variable SHC. However, expression level data from the FNLCR (see Table 8) containing information about every element in this pathway shows that even if some elements are from the same family, they are not acting the same way and might not have the same activity. The column 1 is the expression of the gene in the normal and tumor case. The column 2 shows the value assigned to the variable in the normal case when using three levels. The column 3 shows the value assigned to the variable in the normal case when the range of values for each element is from 0 to 9. The data range in column 1 is from 0 to 16, and given that we are using 0,1 and 2 in the model, we defined thresholds in the data, such that every element with a value between 0 and 5.3 will be 0 in our simulator, between 5.33 and 10.66 will be 1 and from 10.66 to 16 will be a 2. SHC1-2 will be declared as 2 and SHC3-4 will be declared as 0. Therefore, lumping together the SHC1-4 into a single variable SHC and initializing it to 0, 1 or 2 would mean that we are introducing 2, 4 or 2 errors, respectively. This data shows that we need to keep each element separated but also that using more levels could give more precision. Looking at column 4, the initial value for SHC3 is still 0 but for SHC4 is now 1.

Table 8 : Expression level of SHC1-4 and their initialization

	Normal	Initial Value 0-2	Initial Value 0-9
SHC1	11.32276	2	7
SHC2	11.0359	2	7
SHC3	0	0	0
SHC4	1.6896	0	1

When initializing an intermediate model element (that has both upstream and downstream elements), this difference will not have a significant impact on the result. But an input element of this pathway such as CBL being declared as 0 instead of 1 could lead to different results because CBL value will not change during the simulation. Due to the large size of this model, a significant number of elements have a different initial value when using the wider range. These differences will lead to a better precision on the simulation results.

In the special case of discrete modeling, when we use only Boolean variables and logic operations to compute update rules, a problem of scalability appeared. We were not able to create the update rules for elements that have a lot of regulators, and with the computer used to run these simulations, having 13 regulators was the limit before exceeding memory allocated. This problem occurred on many elements and some of them are key elements in the pathway. An example is shown in Figure 26, where SOS1, SOS2, GRB2 and SHC1,2,3,4 will all have 15 regulators.

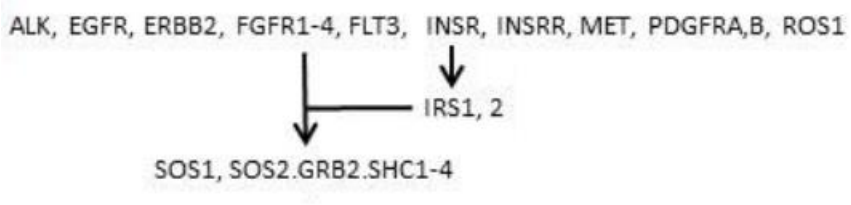


Figure 26 : SOS1, SOS2, GRB2 and SHC1-4 creating a complex [24]

Since all the edges are activating edges, we could can assume a smaller number of inputs in the influence set of SOS, while still obtaining correct results. In this specific case, that is a possible solution. When applied to the whole pathway, the file containing the rules with a maximum of 13 regulators for every update rule is 630 megabytes (MB). An order of comparison is the previous model with gathered elements weighted 28MB and was running in 1 hour and 30 minutes approximately when repeating the simulation 200 times, with 2000 steps in each

simulation run. This model is 22 times bigger, and therefore, the simulation time would be 22 times longer, and we could expect it to run in 33 hours. One of the advantages of simulation is that we do not have to solve ODEs, and thus, the computational cost is smaller, and it is faster than continuous model. To this end, in order to reach the limits of 13 regulators to reduce the size of the model, but still obtain correct simulations results and speed up the running time, one solution that we considered is to use intermediate variables to compute the expression which allows us to easily reduce the size of the model by 99%. With that technique, we managed to reduce the computational time from 1 hour and 30 minutes to 2 minutes for 200 runs and 2000 steps.

In Figure 27, using a toy example, we illustrate our method to reduce the number of regulators for elements that have large number of regulators. First, we group the regulators for a given element into pairs (Step 2 in the figure). This will divide by two the number of element in an update rule, while all the regulations are still included in the model. If after this first step, the number of regulators was still high (more than six), then the pairs would be paired into an intermediate variable in the next layer (Step 3 in the figure). The computational cost of a node is S^N with S the number of possible values and N the number of regulators (positive or negative).

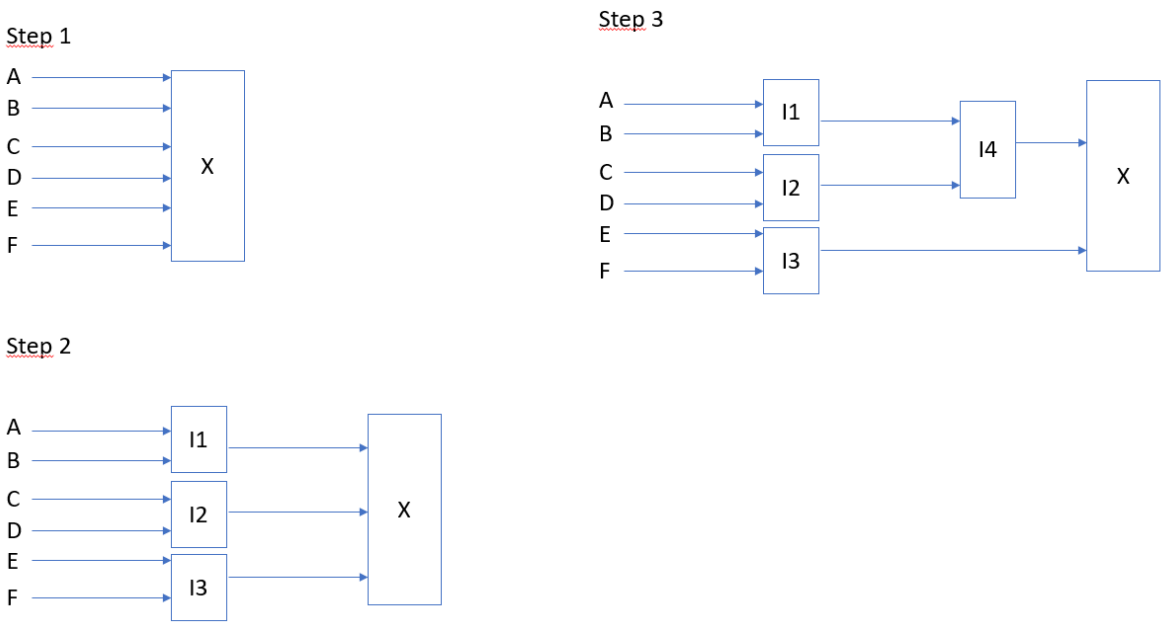


Figure 27 : The method to introduce intermediate variables.

Table 9. Computational cost with S=3

Step	Formula	Computational Cost
1	3^8	6 561
2	3*3^2 + 3^3	54
3	5*3^2	45

Table 10. Computational cost with S=10

Step	Formula	Computational Cost
1	10^8	100 000 000
2	3*10^2 + 10^3	1300
3	5*10^2	500

In Table 9, we use three values for every element, and the gain is 99.31%. In Table 10, where 10 values are possible for every element, the gain is 99.99% in the second, which means that if the original node took 100 seconds to run, on step 3 it will take 0.1s to run.

By including new variables one can change the behavior of the system and add delays. Opposite to delays, we can use the grouped update scheme of the simulator [25], with all intermediate variables to speed up the activation of these variable and reduce the delay. However, it is more complicated to reproduce the experimentally observed behavior in the latter case, because gathering variables together implies making sure that the sensitivity of the update rule to each variable is not changing. To study the sensitivity, we run different simulations using different initial values of regulators, and we can observe if a different initial state leads to a different steady state. However, at what value should the intermediate variables start? It is not always straightforward to determine all initial values for model elements. The goal is to see if we can gain simulation time while getting the same simulation results. To do so, we created a toy model (Figure 28) and decided to take an element from the model, its tier one and tier two regulators. The blue arrows represent element's activation and the yellow arrows represent its inhibition.

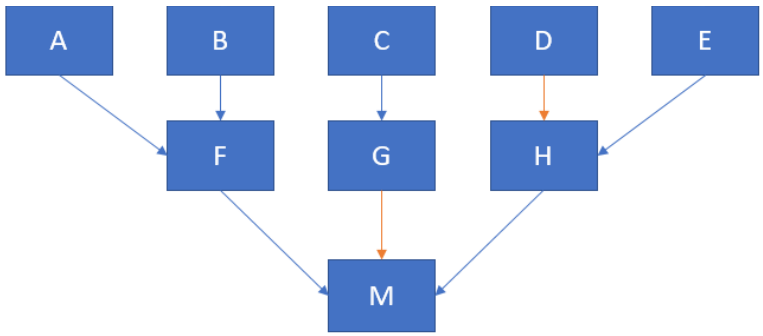


Figure 28 : Example of how an element can be studied with respect to its regulators.

Elements	Positive	Negative
A	L	
B		
C		
D		
E		
F	A,B	
G	C	
H	E	D
M	F,H	G

Figure 29 : Influence sets with positive and negative regulators for elements in Figure 28.

Here, the tier one regulators are F, H and G. The tier two regulators are A, B, C, D and E. L is a tier three regulator, and therefore, will not be included. The first element we decided to study is the complex created by MTOR, RPTOR and MLST8, and the result are shown in Table 11.

Table 11. Comparison of RPTOR rules

Original Model		
RPTOR	(RHEB, RPTOR, MLST8)	(PRKAA1, PRKAA2, PRKAB1, PRKAB2, PRKAG1, PRKAG2, PRKAG3, RPTOR, MLST8)
Decomposed model		
RPTOR	(RHEB, MTOR, MLST8)	(PRKA, PRKB, PRKAG3, MTOR, MLST8)

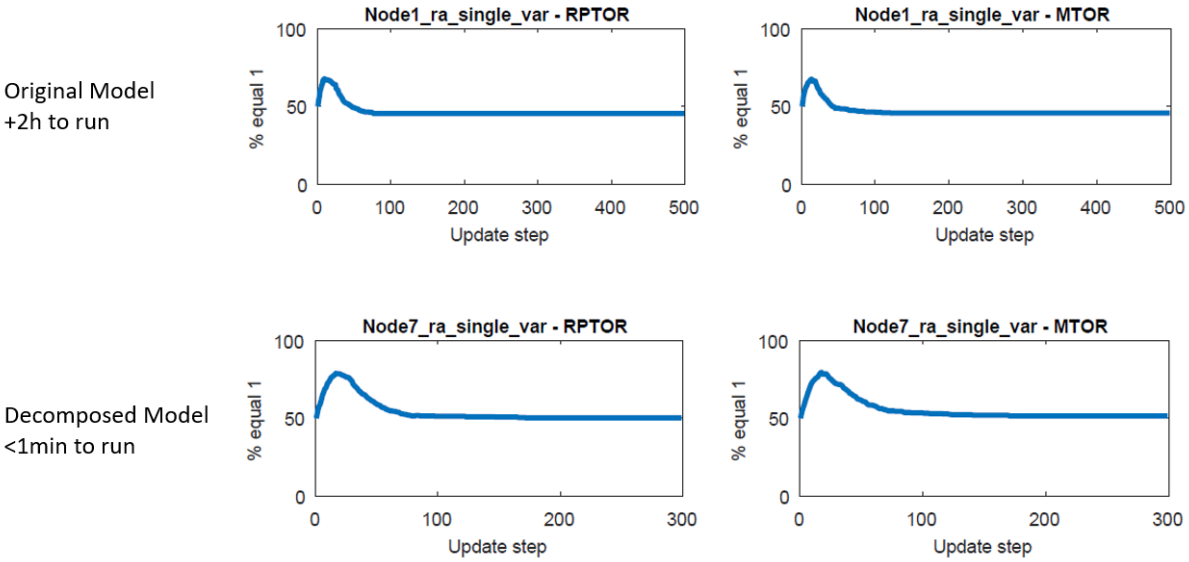


Figure 30 : Results for RPTOR using the regulator lumping approach.

The best results were obtained by initializing the intermediate variable to 0. As foreseen earlier, this solution created a delay. That is why the amplitude of the rise is bigger and the decay is slower. However, these results still show the same behavior, in terms of rise and decay, and the change did not affect the steady state result.

Table 12. Truth table for an AND gate. C=A AND B

A	B	C
0	0	0
0	1	0
1	1	1
1	0	0

Another approach to speed up simulations is to just use discrete variables and common algebraic operations instead of Boolean operations. For example, if we define C = A AND B (truth table shown in Table 12), we can obtain the same result by using the function MIN(A,B). Doing these mathematical operations on discrete values is much faster than comparing Boolean states, and thus, all the Boolean operators can be translated into mathematical functions. The DiSH

simulator [25] that we have been using supports these mathematical operations, by computing the total value of the positive activators and the total value of the negative activators, and If the positive score is higher, then the element value will increase, but if the negative score is higher, then the element value will decrease.

5.3.1 Modeling results

We have thus far developed multiple versions of the extended model, and we keep improving it. This is an iterative process that requires regular discussions with biologists. For example, one of the first simulation results we obtained for the extended model is shown in Figure 31.

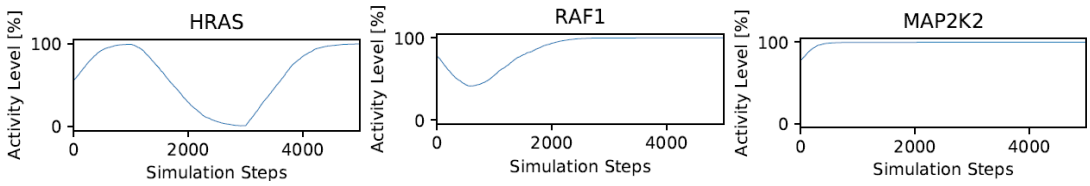


Figure 31 : Activity of HRAS, RAF1 and MAP2K2 before removal of CNKSR

When EGFR is oscillating, we are expecting the oscillations to propagate down the pathway. However, what we can observe from Figure 31 is that RAF1 is barely oscillating with one decay at the beginning then a high activity steady state level, and MAP2K2 (also referred to as MEK) is reaching its steady state at the beginning of the simulation. In the discussion with expert biologists, we determined that some element declared as activators for RAF1 and MAP2K2 were not supposed to act this way. KSR1,2 and CNKSR1,2 were used as direct activators of RAF1 and MAP2K2 but instead of activating the element, they are creating a scaffold. This means that they are necessary for an activation to happen, but not sufficient for this reaction to take place.

When applying this modification, we obtained a new set of results, a few of which are shown in Figure 32.

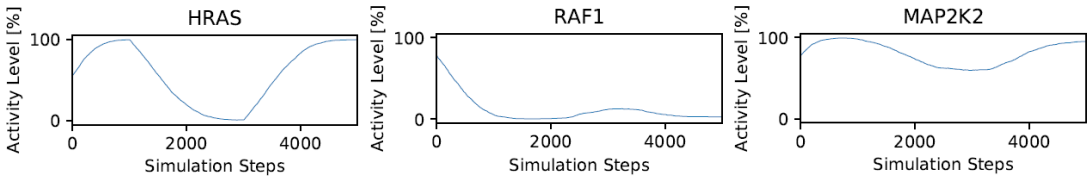


Figure 32 : Traces of HRAS, RAF1 and MAP2K2 after the new feedback from experts.

After a number of iterations, and many discussions with experts, and with the implementations of several new features in our modeling approach, the extended RAS pathway model that we have developed contains more biological motifs than the continuous model found in literature, while it also can reproduce results of specific scenarios such as the use of V600E BRAF and its role in dimerization [26]. These results are promising and can allow biologist to ask for precise scenario and study the response of the RAS pathway in these scenarios.

6.0 CONCLUSION

By combining continuous and discrete modeling, this work aims to improve the precision of discrete modeling, in general, and of the RAS pathway model, in particular. First, we showed how to move from Boolean to discrete modeling and the implementation of several new features in the discrete modeling approach to allow for more accurate results when using this approach. Then, we showed a novel analysis method by using the ADD structure to translate biochemical reactions and to analyze update rules. Finally, we applied the previous work to successfully translate a continuous rule-based model into a discrete model without loss of biological information. Thus, we created the first large scale accurate model of the RAS pathway. An important next step for this work is to include the study of the scalability of the use of ADDs to represent more complex reactions and update rules. Another extension would include the improvements of the RAS model using recent published work and data.

BIBLIOGRAPHY

[1] Hanahan D, Weinberg RA. The hallmarks of cancer. Cell. 2000;100:57–70. [PubMed]

[2] C. P. Vaughn, S. D. Zobell, L. V. Furtado, C. L. Baker, W. S. Samowitz, Frequency of KRAS, BRAF, and NRAS mutations in colorectal cancer. *Genes Chromosomes Cancer 50* (2011). [PubMed]

[3] Catalogue Of Somatic Mutations In Cancer [COSMIC]

[4] Wong R, Cunningham D. Using predictive biomarkers to select patients with advanced colorectal cancer for treatment with epidermal growth factor receptor antibodies. J Clin Oncol. 2008;26:5668–5670. [PubMed]

[5] Minamoto T, Mai M, Ronai Z. K-ras mutation: early detection in molecular diagnosis and risk assessment of colorectal, pancreas, and lung cancers - a review. Cancer Detect Prev. 2000;24:1–12.

[6] Mitin N, Rossman KL, Der CJ. Signaling interplay in Ras superfamily function. Curr Biol. 2005;15:R563–R574. [PubMed]

[7] Zenonos K, Kyprianou K. RAS signaling pathways, mutations and their role in colorectal cancer. World J Gastrointest Oncol. 2013;5:97–101. [PMC free article] [PubMed]

[8] Fang JY, Richardson BC. The MAPK signalling pathways and colorectal cancer. Lancet Oncol. 2005;6:322–327. [PubMed]

[9] Hanahan, D. & Weinberg, R. A. *The hallmarks of cancer*. *Cell* **100**, 57–70 (2000). [PubMed]

[10] Schoeberl B., Eichler-Jonsson C., Gilles E. D., Muller G. Computational modeling of the dynamics of the MAP kinase cascade activated by surface and internalized EGF receptors. Nat. Biotechnol. 2002;20:370–375. [PubMed]

[11] Kochańczyk M, et al. Relaxation oscillations and hierarchy of feedbacks in MAPK signaling. Sci. Rep. 2017;7:38244. doi: 10.1038/srep38244. [PMC free article] [PubMed] [Cross Ref]

[12] Danos, V., Feret, J., Fontana, W., Harmer, R., and Krivine, J. (2007) Rule-based modelling of cellular signalling. Lect. Notes Comput. Sci. 4703, 17–41.[CrossRef]

[13] Tutorial to BioNetGen [Dropbox]

[14] A. Naldi, D. Berenguier, A. Faure, F. Lopez, D. Thieffry, and C. Chaouiya. Logical modelling of regulator networks with GINsim 2.3. Biosystems, 97(2):134–9, 2009.

[15] BooleanNet. Available: http://atlas.bx.psu.edu/booleannet/booleannet.html

[16] BioNetGen. Available: http://bionetgen.org/index.php/Main_Page

[17] Cell Collective. Available: https://www.cellcollective.org/

[18] RAS Executive Model. Available: http://rasmodel.org/

[19] John Hooker, Optimization Methods in Logic, Carnegie Mellon University February 2003, published in Y. Crama and P. Hammer, eds., Boolean Methods and Models, Cambridge University Press, 2

[20] K.S. Brace, R. Rudell, and R. Bryant, "Efficient implementation of a BDD package," DAC-27: ACM/IEEE Design Automation Conference, Orlando, FL, June 1990, pp. 40-45

[21] R. Bryant, "Graph-Based Algorithms for Boolean function manipulation," IEEE Transactions on Computers, Vol. C-35, No. 8, pp. 79-85, Aug. 1986.

[22] E.M. Clarke, M. Fujita, P.C. McGeer, K. McMillan, and J. Yang, "Multi-terminal binary decision diagrams: An efficient data structure for matrix representation," IWLS'93: International Workshop on Logic Synthesis, Lake Tahoe, CA, May 1993, pp. 6a:1-15.

[23] Abbruzzese J.L. Pancreatic cancer: Scanning the horizon for focused interventions. [PubMed]

[24] National Cancer Institute [Image]

[25] Sayed, Khaled & Kuo, Yu-Hsin & Kulkarni, Anuva & Miskov-Zivanov, Natasa. (2017). DiSH Simulator: Capturing Dynamics of Cellular Signaling with Heterogeneous Knowledge. [ResearchGate]

[26] Pratilas C.A., Taylor B.S., Ye Q., Viale A., Sander C., Solit D.B., Rosen N. (V600E)BRAF is associated with disabled feedback inhibition of RAF-MEK signaling and elevated transcriptional output of the pathway. Proc. Natl. Acad. Sci. USA. 2009;106:4519–4524. [PubMed]

[27] Nussinov, Ruth & Jang, Hyunbum & Tsai, Chung Jung. (2014). The structural basis for cancer treatment decisions. Oncotarget. 5. 7285-302. 10.18632/oncotarget.2439. [ResearchGate]