Proceedings of the 2012 IEEE 12th International Conference on Bioinformatics & Bioengineering (BIBE), Larnaca, Cyprus, 11-13 November 2012

Modelling biological pathway dynamics with Timed Automata

Stefano Schivo*§, Jetse Scholma*‡, Brend Wanders¶, Ricardo A. Urquidi Camacho‡,
Paul E. van der Vet¶, Marcel Karperien‡, Rom Langerak§, Jaco van de Pol§ and Janine N. Post‡
§Formal Methods and Tools, Faculty of EEMCS
‡Developmental BioEngineering, MIRA Institute for Biomedical Technology and Technical Medicine
¶Human Media Interaction, Faculty of EEMCS
University of Twente, Enschede, The Netherlands
*These authors contributed equally to this work.

***Abstract*—When analysing complex interaction networks occurring in biological cells, a biologist needs computational support in order to understand the effects of signalling molecules (e.g. growth factors, drugs). ANIMO (Analysis of Networks with Interactive MOdelling) is a tool that allows the user to create and explore executable models of biological networks, helping to derive hypotheses and to plan wet-lab experiments. The tool is based on the formalism of Timed Automata, which can be analysed via the UPPAAL model checker. Thanks to Timed Automata, we can provide a formal semantics for the domain-specific language used to represent signalling networks. This enforces precision and uniformity in the definition of signalling pathways, contributing to the integration of signalling event models into complex, crosstalk-driven networks. We propose an approach to discretization of reaction kinetics that allows us to efficiently use UPPAAL as the computational engine to explore the dynamic cell behaviour. A user friendly interface makes the use of Timed Automata completely transparent to the biologist, while keeping the expressive power intact. This allows to define relatively simple, yet faithful models of complex biological interactions. The resulting timed behaviour is displayed graphically, allowing for an intuitive and interactive modelling experience.**

***Index Terms*—timed automata; signalling pathway; modelling; dynamic behaviour**

I. Introduction

Systems biology is a branch of bioinformatics that focuses on modelling the static relations and dynamic behaviours of biological systems, in order to help biologists to get a better understanding of the complex interactions occurring inside living beings. Executable models are often made available to biologists, allowing them to make a further step from the classical static representation of biological reactions and interactions, and properly study the dynamic evolution of a system: this kind of approach is referred to as *executable biology* [1]. ODEs (ordinary differential equations) are often used to model the dynamics of biochemical processes, and the availability of supporting tools [2], [3], [4] makes their power more accessible to end users. Tools based on process calculi [5], [6] have been successfully employed in modelling complex biological events [7], [8] as well. However, it is often difficult for a biologist to operate with the modelling formalisms and tools that allow to develop executable biological models, especially because such tools often require theoretical foundations or training that a biologist needs to acquire beforehand. Moreover, it is not uncommon for a modelling tool to require its user to input a number of numerical parameters to properly define the dynamic behaviour of one reaction: this is also a problem, because biological reaction rates are often unknown or difficult to measure.

For these reasons, we present the use of Timed Automata [9] as a modelling formalism for biological signalling pathways. As Timed Automata are mainly represented in a visual form, their behaviour can be more intuitively grasped by a non-technical user. Moreover, Timed Automata allow users to be less precise in parameter settings, allowing for more flexibility and implicitly enforcing model robustness. Finally, we developed a user interface for the proposed Timed Automata model (ANIMO, see [10]), implemented as a plug-in to the widespread network modelling tool Cytoscape [11], further improving the user friendliness of the approach and widening its applicability to the biological context. The modelling process is entirely performed via the user interface, allowing the users to completely forget the existence of the underlying Timed Automata model, if they so wish. In this paper, we present a detailed description of how Timed Automata and the model checker UPPAAL [12] can be used for the modelling and analysis of biological networks. We provide also an illustrative case study to show how the use of ANIMO is an asset for biological research.

The plan of the paper is as follows: after a brief introduction on the basic aspects of biological signalling networks and Timed Automata in Section II, we will explain in Section III how our modelling approach works, showing in Section IV an example application. After discussing related work in Section V, Section VI concludes the paper, showing some perspectives for future developments.

II. Preliminaries

A. Signalling pathways in biology

A signalling pathway represents the interactions occurring inside a biological cell when one or more types of signalling molecules come in contact with the cell surface receptors. A typical interaction occurring in a signalling pathway involves

978-1-4673-4358-9/12/$31.00 ©2012 IEEE

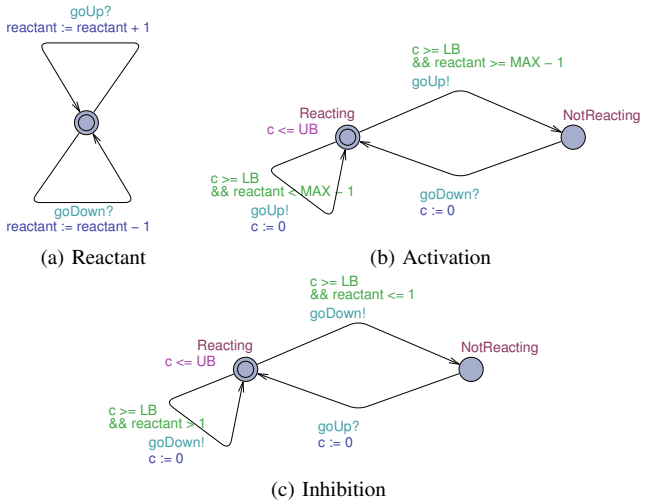
an upstream molecule inducing a post-translational modification (e.g. phosphorylation) to a downstream substrate. Changing the state of a molecule can often result in its *activation*: a chain of such reactions is how a signal is relayed to the target(s) of the pathway. The most common way of graphically representing a pathway identifies molecular species as nodes and reactions as edges, with $\rightarrow$ representing activation and $\dashv$ standing for inhibition (for an example, see Fig. 3).

Experimental evidence shows that signalling interactions often assume the shape of a network including feedback loops instead of a simple chain of activations, thus making the study of such networks a complex task for the biologist. This also makes for a stronger push towards a proper way to represent the dynamic behaviour of a signalling network, which can hardly ever be promptly deduced from a traditional static representation of the network topology. The availability of computational models of such complex, dynamic networks would allow the biologists to perform *in silico* experiments, obtaining useful insights to help them plan new wet-lab experiments or formulate updated theories by adapting their models. Moreover, traditional representations in the domain-specific language used for biological signalling networks are often ambiguous, with extensive use of *ad hoc* semantics, where the same graphical elements (e.g. $\rightarrow$) acquire different meanings (e.g. activation, translocation, promotion) in different network representations. While helping the understanding of the specific cases, this ambiguity makes it difficult to deal with various representations at the same time, always requiring human intervention when integrating information from multiple sources into larger networks.

B. Timed Automata

Timed Automata (TA) [9] are finite-state automata to which real-valued clocks and communication channels have been added. In particular, clocks are used to define conditions enabling certain transitions between locations of an automaton, or to limit the permanence in locations. These conditions are called *guards* and *invariants* respectively. Performing a transition may also require two automata to interact via *synchronization*, where each participant performs one of two complementary actions (called *input* and *output*) on a shared communication channel. Such channels can also be defined as *broadcast*, thus allowing multi-part communication (one sender, many receivers). In order to obtain answers to interesting questions about the behaviour of a model, the technique of model checking [13] can be applied to a TA model using a software tool such as UPPAAL [12].

As an example of TA we introduce in Figure 1 a first basic model of a generic signalling network. The model allows us to represent the active fraction (called from now on *activity level*, or simply *activity* when not ambiguous) of a population of molecules as an integer variable (reactant, Fig. 1a), whose value is changed by reactions with positive (Fig. 1b) or negative (Fig. 1c) effect. In the example model, the value of reactant is bounded inside the interval $[0, \mathsf{MAX}]$, and reaching a bound is the only factor on which the decision to enable or disable a reaction (locations Reacting, NotReacting in Figs. 1b, 1c) is made. A more precise model of a signalling network should also take into account the activity of upstream components when determining the availability of an activating reaction: we will explain in the next section our proposed solution.



(a) Reactant (b) Activation (c) Inhibition

Fig. 1. Three TA templates to model a signalling pathway as represented by UPPAAL's user interface. Each automaton starts evolving from the location marked with two concentric circles. The automaton in (a) updates the reactant variable whenever a reaction occurs, changing its activity level (reactant := reactant+1 increases the activity). (b) represents an activating reaction, which occurs when its internal clock c is inside the interval [LB, UB] (invariant $\mathsf{c} >= \mathsf{LB}$ && $\mathsf{reactant} < \mathsf{MAX} - 1$ and guard $\mathsf{c} <= \mathsf{UB}$), making the target reactant's activity level increase. (c) represents an inhibition reaction, whose effect is the inverse of activation. Shared channels goUp and goDown are used for communication between reaction and reactant automata.

III. Modelling Signalling Pathways

The model presented in Figure 1 will now be extended in order to increase its applicability. We keep the discretization of the reactant activity via integer variables: this allows us to offer more flexibility than a simplistic boolean representation, where the whole population of molecular species A can be seen as either active or inactive. The same reasoning we can now apply to the reactions, *upgrading* them from the boolean view used in the model of Figure 1, where a reaction can either proceed at a fixed (within bounds) speed or can be completely inactive. By making a reaction speed depend on the current activity levels of the involved reactants, we propose an approach closer to reality: when the (inactive) substrate is abundant a reaction proceeds faster, while less (active) enzyme makes the reaction proceed slower. We allow the user to choose the abstraction level of a model by providing a choice in a 2-100 scale for the number of activity levels of each reactant, while different degrees of complexity are taken into account when defining reaction kinetics.

Given a reaction R where A (enzyme) activates B (substrate), we define the duration of R using a kinetic function f that depends on the current activity levels of both A and B: $\text{duration}(R) = f(a, b)$. As we represent the current activity levels a, b of A and B via integer variables, there is a finite

amount of combinations for the values of a and b. Thus we allow for complex kinetic formulae without weakening model checking performance by pre-computing a two-dimensional table containing all possible values of duration(R), based on the user-chosen limits for its input reactants. Finally, in order to retain the possibility of using time intervals as in the model in Figure 1 (cf. LB and UB time bounds), each entry in the time table for R will contain two values: the lower and upper bounds defining the interval inside which the duration of the reaction is expected to be. We will thus compute these values

$$\begin{aligned} \text{duration}(R)[b][a]_{\text{lower bound}} &= f(a,b) \times 0.95 \\ \text{duration}(R)[b][a]_{\text{upper bound}} &= f(a,b) \times 1.05 \end{aligned}$$

for all values of a and b. Note that, in order to account for a natural variability, we made the lower and upper bound of duration$(R)[b][a]$ differ by 5% from the exact value of $f(a,b)$, whose definition is as follows:

$$f(a,b) = \begin{cases} \textit{levelsScale} \times \textit{timeScale} \times \frac{1}{r(a,b)} & \text{if } r(a,b) \neq 0 \\ \infty & \text{otherwise} \end{cases} \quad (1)$$

levelsScale is a scale factor which depends on the number of levels of the involved reactants: this allows us to keep the reaction parameters independent from the granularity of its reactants, allowing the user to freely choose the most suitable number of levels for each reactant. *timeScale* is a global parameter of the model, and is chosen by the user: it is the rate between real life seconds and TA time units. This allows the user to ask for simulation runs using a real-life unit of measurement for the time limit, instead of less intuitive "time units". r is the reaction rate, whose definition includes a kinetic constant k and is given by the user when selecting one of the three simplified kinetic scenarios available:

Scenario 1: $r(a) = k \times a$ the reaction rate depends on the activity level of the enzyme (in this case, also f is unary and a vector is generated instead of a two-dimensional table)

Scenario 2: $r(a,b) = k \times a \times b$ the reaction rate depends on the activity levels of both the enzyme and the substrate, and more precisely b represents the inactive fraction of substrate if the reaction has activating effect, while it refers to the active fraction when the effect is inhibitory

Scenario 3: $r(c,d) = k \times c \times d$ the reaction rate depends on two user-selected reactants, none of which needs to be the target of the reaction.

As an example, Table I shows the lower and upper bounds table for a reaction $A \rightarrow B$, using scenario 2 and $k = 0.02$.

Please note that the approach presented here to discretize reaction dynamics is less general than the one proposed in [14] because our primary objective is to allow the biologists to define more abstract models. Such models are aimed at helping to speed up the experimental research by evaluating and formalizing the existing network topologies, or by formulating alternative hypotheses rather than define closely-matching descriptions based on precise biochemical reaction kinetics. For the same reason, the choice of the scenario and the value for its single constant k are the only inputs requested to the user when defining the kinetics of a reaction: this helps in simplifying the task of the modeller. As a further simplification, we give the user the possibility to make a qualitative choice instead of a quantitative one when defining the value for k: by providing a pre-defined set of clearly labelled reaction speeds (*very slow*, *slow*, *medium*, *fast*, *very fast*), we encourage the user to work (at least initially) in a perspective where the relative speed of reactions defines the dynamics of a network. It is in fact easier for a biologist to guess the speed of a reaction if compared with another one ("R_1 is certainly/probably faster than R_2"), rather than to find its correct kinetic constant [3]. The task of making the model fit to experimental data by more precisely setting the values for k is suggested as a second step, when a balance of speeds among the components of a network is already present and working as intended.

TABLE I
LOWER AND UPPER BOUNDS FOR THE TIME TABLE OF A REACTION $A \rightarrow B$ WITH KINETICS SCENARIO 2, $k = 0.02$ AND TIME SCALE OF 0.1 SECONDS PER UPPAAL TIME STEP. A HAS 5 ACTIVITY LEVELS, WHILE B HAS 3. IN BOTH CASES, 0 MEANS COMPLETELY INACTIVE.

B \ A	0	1	2	3	4	5
0	∞	362	181	121	90	72
1	∞	483	241	161	121	97
2	∞	724	362	241	181	145
3	∞	∞	∞	∞	∞	∞

(a) Lower bound

B \ A	0	1	2	3	4	5
0	∞	400	200	133	100	80
1	∞	533	267	178	133	107
2	∞	800	400	267	200	160
3	∞	∞	∞	∞	∞	∞

(b) Upper bound

A. Timed Automata model

The proposed model of signalling pathways contains an instance of the Reaction TA template (see Fig. 2) for each reaction present in the model. An integer variable is defined for each of the n reactants of the network to represent the reactant's current activity level, and is initialized as specified by the user. Finally, a series of channels called reaction_happening$_i$, with $i \in \{1, 2, \ldots, n\}$, is defined to allow reaction processes to communicate among themselves when the activity level of the i-th reactant has been updated.

The TA template in Figure 2 depends on two input reactants, to which reactant1 and reactant2 are references, and influences one reactant (referenced by output): it is used for kinetic scenarios 2 and 3, while a slightly simpler template is used for scenario 1. The basic idea underlying the Reaction TA is to perform a continuous cycle, where at each iteration we wait for the reaction to complete, and then update the activity level of the target reactant (variable output). The variable named delta represents the increment caused in output when the reaction occurs: thus, delta contains $+1$ if the reaction is activating and -1 if it is inhibiting. The locations in the Reaction TA template in Figure 2 have been labelled s1, s2, s3, s4, where

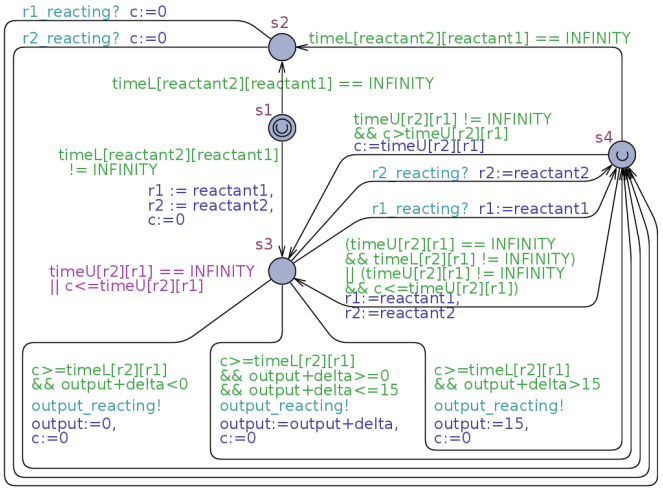


Fig. 2. The TA template for the reactions in the model. The two matrices timeL and timeU define respectively the lower and upper bound times for each possible configuration of the activity levels of the reactants from which the reaction rate depends. Channels r1_reacting, r2_reacting and output_reacting are used for communicating modifications to the value of the involved reactants. Each of these three channels is a reference to a global shared channel reaction_happening$_i$, where i is the index of the corresponding reactant.

s1 is the starting location.

s1 is used to reset the internal clock c and start counting (transition to location s3) or to enter a "dormant" location if the reaction cannot occur (transition to location s2). This is the case when the lower bound of the reaction duration is declared as INFINITY (i.e. the reaction rate is 0, cf. Eq. 1) because the conditions for applicability of the reaction are not met. E.g., if the reaction activates its target and the kinetics are based on scenario 2, the reaction cannot occur if either no inactive substrate or no active enzyme are available (cf. Tab. I). The U symbol inside locations s1 and s4 marks them as *urgent*: while there is at least an automaton in an urgent location, time cannot progress. In this way, all necessary updates are made before the reactions can continue.

The waiting location is identified by the label s3: the automaton can exit from this location when the activity level of an input reactant has been changed by another reaction (transitions from s3 to s4, receiving a communication on channel r1_reacting or r2_reacting), or when the current value of the internal clock c is inside the interval $[\text{duration}(R)[\text{r2}][\text{r1}]_{\text{lower bound}}, \text{duration}(R)[\text{r2}][\text{r1}]_{\text{lower bound}}]$, i.e. when the reaction can occur. The bounds for the duration of the reaction under the current conditions are found in the corresponding tables timeL[][] (corresponding to $\text{duration}(R)[][]_{\text{lower bound}}$) and timeU[][] ($\text{duration}(R)[][]_{\text{lower bound}}$), which are indexed by the current activity levels of the two input reactants r1 and r2.

If the reaction cannot occur (e.g. because all substrate is already active), the automaton stays in location s2 until an update happens which can possibly change the current situation (transitions from s2 to s4).

Finally, location s4 is used to check that the clock settings are consistent with the current (possibly changed) time bounds.

For an example run, consider two reactions $R_1 = A \rightarrow B$ and $R_2 = C \dashv B$, both based on scenario 2, with starting activity levels $A = 10/10$, $B = 0/10$, $C = 10/10$. The two automata for R_1 and R_2 will start from location s1 and move immediately to s3 and s2 respectively: as both reactions depend on the activity level of B (see the definition of scenario 2 in Sect. III) and B is completely inactive, R_1 can proceed at full speed, while R_2 cannot occur. After some time (depending on the parameter k of R_1), a transition s3 $\rightarrow$ s4 will be taken by the automaton for R_1, incrementing the activity level of the output reactant B by 1 (output = output + delta in the template). At the same time, a synchronization on channel reaction_happening$_B$ (corresponding to output_reacting for R_1 and to r2_reacting for R_2) will allow the automaton for R_2 to reach location s4.[1] As R_1 can still occur with $A = 10/10$ and $B = 1/10$, the proper s4 $\rightarrow$ s3 transition is taken next. For the same reason, a transition s4 $\rightarrow$ s3 is taken in the automaton for R_2, making thus both reactions active. From this point on, the evolution of the system will proceed depending on the kinetic parameters defined for the reactions, and the evolution of the activity of B will vary depending on which of the two reactions will occur more often.

B. Analysis with ANIMO

We use the UPPAAL model checker for resolving time reachability queries (e.g. $E <>$ time $\geq T$, which can be read as "is it possible to reach a state in which the current time is at least T?") from which we extract the simulation data displayed to the user. In particular, a simulation trace is obtained via the UPPAAL command line tool `verifyta`, to which we ask for "some trace" (`-t0`), using random depth search order (`-o2`). The trace we obtain from UPPAAL is symbolic, i.e. each state in the trace is defined for a time interval. For example, consider the (simplified) trace time $\in [\textit{lowerBound}, \textit{upperBound}], R_1 = a_1, R_2 = a_2, \ldots, R_n = a_n$, with $R_i = a_i$ meaning "the activity level of reactant R_i is a_i". We compute the exact time point for the current state by sampling an uniform distribution over the interval $[\textit{lowerBound}, \textit{upperBound}]$. Parsed traces can be graphically explored in the ANIMO user interface, where they are displayed as time-series graphs of selected reactants. The user can also add to the graph a time series coming from experimental data, in order to quickly compare them with the predictions from the model. Finally, by moving a slider under the time series graph, the original input model is enhanced by color codings, showing the activity levels of all reactants for the highlighted time instant.

Until a proper support to complex queries is added to ANIMO, the interested user can directly employ the UPPAAL tool to get non-trivial questions answered. We note however that taking this non-mediated approach requires at least basic training in temporal logic querying.

[1] Please note that reaction_happening$_B$ corresponds also to r2_reacting in the automaton for R_1, but that automaton is already performing output_reacting!, and no more than one transition can be taken at a time.

IV. Case Study

As an example application of our approach we present a model of the crosstalk between two growth factors important for regulating cell development and function: EGF (epidermal growth factor) and NGF (neuronal growth factor). In particular, we model part of the signalling pathways of these growth factors in PC-12 cells, a special cell type used to study the formation of neuronal cells, and compare the behaviour of our model with the data presented in [15].

It has been observed that the activation of Erk (extracellular regulated kinase) shows significantly different dynamics when a cell is treated with EGF as compared to treatment with NGF. In particular, a transient peak-shaped activation can be observed when a PC-12 cell is treated with EGF, while a sustained activation is measured when the treatment is made with NGF, even after removal of the input signals via growth factor-neutralizing antibody. This led the authors of [15] to formulate the hypothesis that a positive feedback exists from Erk to an upstream node in the network and that some other component interfering with this feedback is inhibited by treatment with NGF. We have implemented these topological reasonings in the ANIMO model in Figure 3; the parameters used in the network are shown in Table II. The experimental conditions to which our model refers are the addition of either EGF or NGF in sufficient quantity to saturate their respective receptors, followed at the 10 minute mark by the addition of a growth factor-neutralizing antibody which binds to the growth factors, shutting off the input signal to the network. The evolution of the network is observed for 60 minutes from the initial treatment.

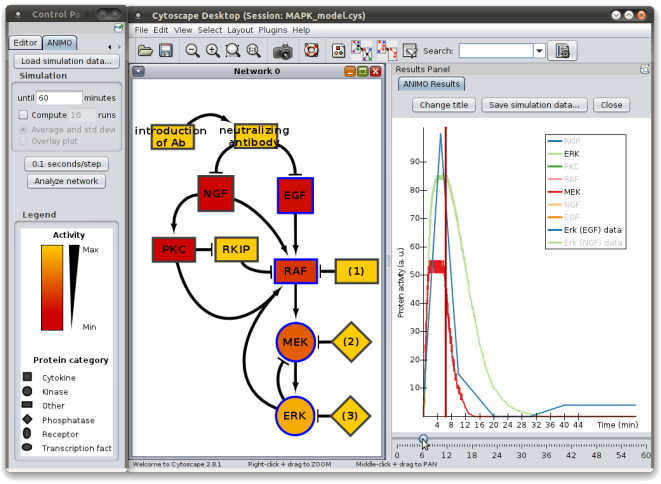


Fig. 3. The MAPK model represented in the ANIMO user interface. The *Network* central panel represents the model in the classical nodes-edges way familiar to the biologists. To this representation, node colours and shapes are added to represent current activity level and protein category, as defined in the *Legend* panel on the left. On the right, the *Results Panel* shows a graph of the activities of selected nodes during the user-chosen interval of 60 minutes. A vertical red bar, which can be moved with the underlying slider, indicates the point in the simulation trace on which the colouring of the nodes in the *Network* panel is based. The graph allows the user to visually compare simulation results with experimental data: the Erk (EGF) data series is based on experimental data from [15].

Nodes labelled as EGF, NGF, PKC (protein kinase C), RKIP (Raf kinase inhibitory protein), RAF (Raf), MEK (MAPK Erk kinase), ERK in the ANIMO model correspond to the ones included in the network topology discussed in [15]. Nodes introduction of Ab and neutralizing antibody are used to represent the introduction of growth factor-neutralizing antibody after 10 minutes from the start of the initial treatment: the reaction activating node neutralizing antibody takes 10 minutes to complete. Finally, nodes labelled with numbers ((1), (2) and (3)) are molecules known to exist and whose task is to inactivate their targets, allowing the network to reset to its initial state after a signal has been processed. In particular, (2) and (3) belong to the family of phosphatases. The feedback activation from ERK to RAF (double edge from ERK and PKC to RAF in Fig. 3) is modelled as a reaction based on scenario 3 kinetic approximation (cf. reaction ERK *and* PKC $\rightarrow$ RAF in Tab. II): this allows us to make the reaction depend both on ERK and PKC activities, while influencing RAF activity. This was done to better underline the importance of PKC in enabling the feedback mechanism causing the observed sustained Erk activation upon NGF treatment (cf. Fig. 4b).

TABLE II
Parameter settings for the model in Figure 3. The Scen. column contains the number of the approximated scenario for each reaction (see Sect. III). $^{(*)}$ In order to reflect experimental treatment conditions, the settings for initial NGF and EGF activity are to be considered mutually exclusive: if one is at maximum activity, the other is set at 0.

Reactants			Reactions		
Name	**Levels**	**Init act.**	**Reaction**	**Scen.**	k
intr. Ab	1	1	intr. Ab $\rightarrow$ neutr. Ab	1	3.5e-02
neutr. Ab	1	0	neutr. Ab $\rightarrow$ NGF	2	3e-03
NGF	15	$15^{(*)}$	neutr. Ab $\dashv$ EGF	2	5e-02
EGF	15	$15^{(*)}$	NGF $\rightarrow$ PKC	2	3.2e-04
PKC	15	0	NGF $\dashv$ RAF	2	4e-03
RKIP	20	20	EGF $\rightarrow$ RAF	2	8e-03
RAF	60	0	PKC $\dashv$ RKIP	2	1.8e-03
(1)	1	1	RKIP $\dashv$ RAF	2	8e-03
MEK	60	0	(1) $\dashv$ RAF	2	2.5e-03
(2)	1	1	ERK *and* PKC $\rightarrow$ RAF	3	3.2e-03
ERK	100	0	RAF $\rightarrow$ MEK	2	4e-02
(3)	1	1	(2) $\dashv$ MEK	2	3e-03
			ERK $\dashv$ MEK	2	1.5e-02
			MEK $\rightarrow$ ERK	2	3e-02
			(3) $\dashv$ ERK	2	3e-03

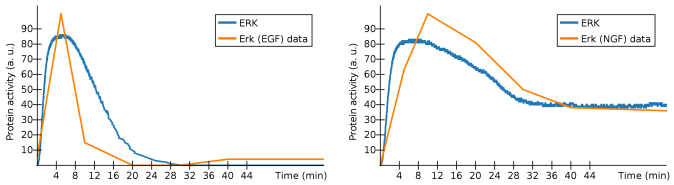


Fig. 4. Comparison between the model and experimental data. **(a)** Treatment with 100 ng/ml EGF resulting in transient Erk activation. **(b)** Treatment with 50 ng/ml NGF resulting in sustained Erk activation. The ERK series are computed from the model, while Erk (EGF) data and Erk (NGF) data are based on experimental data from [15].

As can be seen in the graphs in Figure 4, the general behaviour of Erk observed in experimental data is reflected in our model: transient activation is exhibited upon EGF treatment, while NGF treatment causes sustained Erk activity. A model fitting experimental data more closely could be obtained when considering intermediate nodes, which were deliberately left out of our model. However, the primary objective of the presented case study is showing that it is possible to use TA-based modelling for a first, fast draft of a theoretical signalling network topology, allowing the biologist to obtain a better insight into the studied phenomena. We are currently working on a larger case study, with an ANIMO model consisting of 52 reactants and 57 reactions. So far, our conclusions on user friendliness and usefulness for biological research extend to this setting.

V. Related work

We can divide the existing formal approaches to modelling biological systems into two large groups: one includes approaches based on ordinary differential equations (ODEs), while the other encompasses all methods based on concurrent systems. This distinction captures the main peculiarity of concurrent systems, which allows one to describe a system by specifying its components in isolation, and then define the interaction rules; on the other hand, a method based on ODEs will need to explicitly account for all changes each interaction can cause to each component of the system. Even if less maintainable, models based on ODEs are usually easier to understand, because of their strong connection with the actual chemical reaction laws governing the evolution of a system, while an approach based on concurrency will usually add a layer on top of the canonical description of chemical reactions, requiring a user to acquire some practice before being able to fully profit of the different paradigm.

Tools that allow to obtain models directly based on ODEs include for example COPASI [2], E-Cell [4] and GNA [3]. These tools add to the potential of ODEs by coupling them with additional modelling approaches, like the stochastic models[2] used in COPASI and E-Cell, and by allowing for qualitative modelling, as does GNA.

Methods relying on distributed systems can be either qualitative or quantitative, the distinction being based on the possibility to add to the model numerical (quantitative) information such as speed of reactions, concentration of species, volume of solution. The tool ANIMO we present in this paper places itself among the quantitative methods based on distributed systems. ANIMO distinguishes itself by allowing the user to decide specific levels of precision and granularity for each part of the modelled network. The speed of a reaction can be adjusted on two granularity levels, allowing to choose among 3 approximation scenarios, and by letting the user to select either qualitative or quantitative kinetic parameters. The same kind of individual granularity choice is available for single reactants, each of which can have its activity discretized in a different number levels. While allowing for more precision than boolean networks, ANIMO does not necessarily require the user to be as precise as the models based on ODEs or on stochastic processes, which is a useful feature because precise values are hard to obtain from biological experiments.

[2]Stochastic modelling becomes particularly useful when some molecular species have very small concentration, which nullifies the well-mixed solution assumption used in models based on ODEs.

VI. Conclusions and future work

We contribute to the modelling of biological pathways by introducing a formalization of the domain-specific language traditionally used for pathway representation into a model based on Timed Automata (TA). Thanks to the proposed discretization process, we can account for arbitrarily complex kinetic formulae, while keeping the actual complexity of the model hidden *under the hood*. The foundations laid in the presented approach will make the power of TA available to non expert users, helping biologists to better understand and elaborate models of signalling networks. In this respect, we plan to apply ANIMO in a research project aimed at studying chondrocyte signalling in relation to osteoarthritis. The objective is to enhance cartilage tissue engineering strategies by investigating the effect of extracellular signalling and cell-matrix interactions in order to mimic these signals in the development of biomaterials that provide direct support while stimulating chondrocytes to make the correct extracellular matrix.

As for future developments of the presented approach, we intend to continue exploring the possibilities deriving from the application of TA to the modelling of biological networks, providing more advanced features that allow the user to get more interesting questions answered. In particular, a model that already proves a good fit with existing data should be able to help answering complex, biologically meaningful questions to the user, allowing for *in silico* experiments as e.g. "What is the combination of inputs that leads to activity$(A) \geq 20/50$ and activity$(B) < 10/80$ in 120 minutes?". Moreover, taking advantage of recently introduced statistical model checking capabilities to UPPAAL [16], we plan to extend the current modelling paradigm adding the possibility to define stochastic behaviour, and support probabilistic queries as e.g. "What is the probability that reactant A reaches an activity of at least 40/50 within the first 30 minutes?". Other interesting developments are aimed at further speeding up the modelling phase, in order to let the user start interrogating a model as soon as possible. We plan to include a support for parameter sensitivity analysis and automatic parameter fitting to a given experimental data set. We are also developing techniques based on automata learning [17] for deriving the topology of a biological network based on a series of constraints and experimental data series. Finally, the introduction of more advanced abstraction techniques will help to increase the performance of the model checking phase and thus enable the biologist to manage larger signalling networks with less computational resources.

ACKNOWLEDGMENT

We acknowledge support from Netherlands Organisation for Scientific Research (NWO) CASIMIR to J.S.

REFERENCES

[1] J. Fisher and T. A. Henzinger, "Executable cell biology." *Nature biotechnology*, vol. 25, no. 11, pp. 1239–1249, Nov. 2007.

[2] P. Mendes, S. Hoops, S. Sahle, R. Gauges, J. Dada, and U. Kummer, "Computational modeling of biochemical networks using COPASI systems biology," ser. Methods in Molecular Biology, I. V. Maly, J. M. Walker, and J. M. Walker, Eds. Totowa, NJ: Humana Press, 2009, vol. 500, ch. 2, pp. 17–59.

[3] H. de Jong, J. Geiselmann, C. Hernandez, and M. Page, "Genetic Network Analyzer: qualitative simulation of genetic regulatory networks," *Bioinformatics*, vol. 19, no. 3, pp. 336–344, Feb. 2003.

[4] M. Tomita, K. Hashimoto, K. Takahashi, T. S. Shimizu, Y. Matsuzaki, F. Miyoshi, K. Saito, S. Tanida, K. Yugi, J. C. Venter, and C. A. H. III, "E-CELL: software environment for whole-cell simulation." *Bioinformatics*, vol. 15, no. 1, pp. 72–84, Jan. 1999.

[5] L. Dematté, C. Priami, and A. Romanel, "Modelling and simulation of biological processes in BlenX," *SIGMETRICS Perform. Eval. Rev.*, vol. 35, pp. 32–39, March 2008.

[6] F. Ciocchetta and J. Hillston, "Bio-PEPA: A framework for the modelling and analysis of biological systems," *Theor. Comput. Sci.*, vol. 410, pp. 3065–3084, August 2009.

[7] F. Ciocchetta, A. Degasperi, J. Heath, and J. Hillston, "Modelling and analysis of the NF-κB pathway in Bio-PEPA," in *Transactions on Computational Systems Biology XII*, ser. Lecture Notes in Computer Science, C. Priami, R. Breitling, D. Gilbert, M. Heiner, and A. Uhrmacher, Eds. Springer Berlin / Heidelberg, 2010, vol. 5945, pp. 229–262.

[8] P. Lecca, "BlenX models of α-synuclein and parkin kinetics in neuropathology of Parkinson's disease," *Journal of Biological Systems*, vol. 19, pp. 149–181, June 2011.

[9] R. Alur and D. L. Dill, "A theory of timed automata," *Theor. Comput. Sci.*, vol. 126, pp. 183–235, April 1994.

[10] ANIMO, http://wwwhome.ewi.utwente.nl/~schivos/animo.

[11] S. Killcoyne, G. W. Carter, J. Smith, and J. Boyle, "Cytoscape: a community-based framework for network modeling." *Methods in molecular biology (Clifton, N.J.)*, vol. 563, pp. 219–239, 2009.

[12] K. G. Larsen, P. Pettersson, and W. Yi, "UPPAAL in a nutshell," *International Journal on Software Tools for Technology Transfer (STTT)*, vol. 1, pp. 134–152, 1997.

[13] E. Clarke, "Model checking," in *Foundations of Software Technology and Theoretical Computer Science*, ser. Lecture Notes in Computer Science, S. Ramesh and G. Sivakumar, Eds. Springer Berlin / Heidelberg, 1997, vol. 1346, pp. 54–56.

[14] O. Maler and G. Batt, "Approximating continuous systems by timed automata," in *Formal Methods in Systems Biology*, ser. Lecture Notes in Computer Science, J. Fisher, Ed. Springer Berlin / Heidelberg, 2008, vol. 5054, pp. 77–89.

[15] S. D. M. Santos, P. J. Verveer, and P. I. H. Bastiaens, "Growth factor-induced MAPK network topology shapes Erk response determining PC-12 cell fate," *Nature Cell Biology*, vol. 9, no. 3, pp. 324–330, Feb. 2007.

[16] A. David, K. Larsen, A. Legay, M. Mikučionis, and Z. Wang, "Time for statistical model checking of real-time systems," in *Computer Aided Verification*, ser. Lecture Notes in Computer Science, G. Gopalakrishnan and S. Qadeer, Eds. Springer Berlin / Heidelberg, 2011, vol. 6806, pp. 349–355.

[17] J. Tretmans, "Model-based testing and some steps towards test-based modelling," in *Formal Methods for Eternal Networked Software Systems*, ser. Lecture Notes in Computer Science, M. Bernardo and V. Issarny, Eds. Springer Berlin / Heidelberg, 2011, vol. 6659, pp. 297–326.