METHODOLOGY

Formal models of biological networks in ANIMO 2

Stefano Schivo[1#], Jetse Scholma[2#], Paul E. van der Vet[3], Marcel Karperien[2], Janine N. Post[2], Jaco van de Pol[1] and Rom Langerak[1*]

[*]Correspondence:
r.langerak@utwente.nl
[1]Formal Methods and Tools, Faculty of EEMCS, University of Twente, 7522NH Enschede, The Netherlands
Full list of author information is available at the end of the article
[#]Equal contributor

Abstract

Background: The large amount of data with which an experimental biologist must deal makes it largely impossible to reason on the dynamics of complex systems without relying on computational support. In order to be fully profitable, computational modelling tools must be both accessible and powerful, providing a means to formalise and analyse experimental outcomes without requiring additional training.

Results: We show that the software tool ANIMO (Analysis of Networks with Interactive MOdelling) can be used in different modelling tasks, and illustrate how this can be an asset for the biological researcher. ANIMO models for *Drosophila melanogaster* circadian clock and signal transduction events downstream of TNFα and EGF in HT-29 human colon carcinoma cells are used as case studies to illustrate the possibilities offered by the tool.

Conclusions: The software tool ANIMO, recently updated to version 2, allows the biologist to develop a formal model through a user friendly interface, achieving useful insight without the need to study any mathematical formalism.

Keywords: modelling; signalling pathway; timed automata; dynamic behaviour

Background

Modelling in cell biology

In living cells, processes are regulated by networks of interacting molecules. Aberrations in these networks underlie a wide range of pathologies. The development of new therapies requires a thorough insight in the functioning of these networks. Obtaining such insight can be a challenging task. Feedback loops and crosstalk between pathways lead to an intricate wiring of the network. Since the human brain is ill-suited to grasp the non-linear dynamics of these complex networks and the entailed emergent properties, the role of computational support is increasing in molecular biology. In this context, we can profit from models in different ways:

1 to organize data and store knowledge,
2 to structure reasoning and discussion
3 to perform *in silico* experiments and derive hypotheses.

As models are a formalization of knowledge or theories, an underlying formalism is needed to express this knowledge. Different formal methods have been successfully applied to construct representations of biological systems. Among these methods are Boolean logic [1, 2], ordinary differential equations (ODEs, reviewed by [3]), interacting state machines [4, 5], process calculi [6, 7], Timed Automata [8, 9, 10] and Petri nets [11, 12]. Most of these formal methods have been implemented into software tools to aid the process of modelling. Due to the lack of such a supporting tool, Timed Automata have remained a less frequently applied method.

Timed Automata for biological models

Timed Automata have been developed to model the dynamic behaviour of systems with processes running in parallel [13]. As such, Timed Automata have been applied in communication protocols and industrial control engineering [14, 15, 16]. The parallels between these application areas and regulatory processes in cells have triggered the step towards their use in biology. In [8], the authors used Timed Automata to extend a classical modelling paradigm [17], allowing to add temporal dynamics to gene network models. In [9], a model of biological oscillators was described and synchronization properties were tested in this dynamic system. A discretization of ODEs to Timed Automata was proposed in [10], applying a translation between the two formalisms to an example gene regulatory network. Two different approaches to transforming a Petri net model into Timed Automata were presented in [18], where the important issue of state space explosion was also addressed. Finally, the authors of [19] proposed an *ad hoc* Timed Automata model of a radiation treatment system, which was then validated through the analysis tool UPPAAL [20].
Each of these approaches has been successfully validated, demonstrating the potential of Timed Automata in biological applications. However, these approaches were all limited to simple or specific examples and none of these modelling methods has led to a tool implementation of the proposed method to encourage a broader use of Timed Automata in molecular biology. We have recently introduced the software tool ANIMO (Analysis of Networks with Interactive MOdelling, [21]), which aims at making available the power of Timed Automata to the experimental biologists without requiring them to be fluent in any mathematical formalism.

In this paper we present ANIMO 2.0, an updated version of ANIMO which allows for a considerably more efficient analysis of models. ANIMO 2.0 also introduces new features, such as the possibility to interrogate an ANIMO model with model checking queries represented as human language sentences instead of mathematical formulae. The case studies presented in this paper are centered around possible usage scenarios for ANIMO, and aim at illustrating how a biologist can profit from the tool in its latest iteration. An installation and user manual for ANIMO 2.0 is available at `http://fmt.cs.utwente.nl/tools/animo/manual.html`.

A short introduction to ANIMO

[[briefly describe ANIMO, its modelling assumptions and its intended use. Probably good idea to refer to Gene paper]]

Results

Modelling oscillation

Results obtained with ANIMO are comparable to results with other modeling approaches. To demonstrate this, Figure 3 represents an ANIMO model of the circadian clock in *Drosophila Melanogaster*, based on the work by [22], where ordinary differential equations (ODEs) were used. The cyclic behavior of the circadian clock is based on the alternating formation and destruction of the CYC/CLK protein complex. Concentration levels of this complex are in turn regulated by a series of proteins which are produced as a consequence of CYC/CLK formation. The CWO protein is central to the functioning of the network, as it degrades the mRNA

for most of the involved proteins. As such, CWO acts as an inhibitor that counterbalances the effect of CYC/CLK. The positive influence of the light-regulated cryptochrome CRY on the degradation of TIM is a consequence of the passage between day and night, allowing the circadian clock to synchronize to a time zone (see Suppl. Sect. **??**).

The output of the ANIMO model in Figure 3 closely matches the original ODE model. In particular, the oscillations in both models show the same periods and phases (see Suppl. Fig **??**). Due to the compositional nature of Timed Automata ANIMO allows for intuitive *in silico* knock-out experiments, by right-clicking a node in the model and disabling it. Such experiments have been done before [22] and give similar results in our model.

[[Insert description of K.O. experiments with Drosophila model]]

Using ANIMO to generate hypotheses

We constructed a model of the signaling network downstream of TNFα and EGF, formalizing the crosstalk that takes place between the pathways at different levels of cellular regulation. We first modeled the two pathways in isolation (Suppl. Figs. **??**, **??**), using information on protein interactions from the KEGG [23] and phosphosite [24] databases. These models were fitted to experimental data from studies by [25] and [26]. We then merged the two pathways into a single model and added autocrine crosstalk between the pathways that has been described by [26]. Briefly, stimulation with TNFα (TNFa in the model) leads to a rapid release of TGFα (TGFa), which activates the EGF receptor (EGFR). This activation causes secretion of IL-1α (IL-1a) at later time points. The effect of IL-1α is down-regulated by the secretion of IL-1 receptor antagonist (IL-1ra) downstream of TNFα. The resulting model (Fig. 4A) was compared to the experimental data for treatments with 100 ng/ml TNF alone and 100 ng/ml EGF alone (data not shown) [25].

At this point, the behavior of the model deviated from the data for some of the nodes. Changing the parameters of the model was not enough to reproduce the behaviour shown by experimental data. This is an interesting situation, as it requires changes in the topology of the model, which can be interpreted as new hypotheses. Below, we give two examples and show how adaptation of the model can be used to generate novel testable hypotheses.

Experimentally, treatment with TGFα alone does not lead to secretion of IL-1α. Instead, a co-stimulation with TGFα and TNFα is required [26]. However, in the first version of the model, treatment with TGFα was sufficient for IL-1α expression (Fig. 4B). Given the time delay until secretion of IL-1α, it can be expected that *de novo* synthesis of IL-1α is required and that both TNFα and TGFα are needed to activate transcription of the IL-1α gene. JNK1 and ERK signal downstream of TNFα and TGFα, respectively, and are known to affect the activity of multiple transcription factors. We altered the model to make activation of IL-1α expression dependent on both JNK1 activity and ERK activity (Suppl. Fig. **??**, arrows linking JNK1 and ERK to IL-1a gene). After this modification to the model, IL-1α was no longer secreted upon stimulation with TGFα alone, which greatly improved the fit between the measured IL-1α levels and the model (Fig. 5B). This hypothesis

could now be used to design a new experiment to validate IL-1α as a target of combined JNK1 activity and ERK activity in HT-29 cells. For example, kinase inhibitors specific to JNK1 and ERK could be used to confirm that activity of both kinases is required for expression and secretion of IL-1α. Performing the experiment is beyond the scope of this study, but this hypothesis finds support in literature. Transcription factors c-Jun and c-Fos together form a heterodimer known as AP-1 and are activated by JNK1 and ERK, respectively [27, 28]. AP-1 has been reported to bind to the promoter of IL-1α, providing evidence for a role in the regulation of IL-1α expression [29]. Based on these findings in literature we included c-Jun and c-Fos in our model as transcriptional activators of IL-1α (Fig.5A).

As a second example, we considered the behaviour of JNK1 and MK2. In the model, both proteins were located downstream of TNFα but not TGFα or EGF. Hence, the model did not show an effect of C225, a pharmacological inhibitor of ligand-EGFR binding, on activation of JNK1 or MK2 after stimulation with TNFα. However, experimental data show that C225 strongly reduces activation of JNK1 and MK2 upon stimulation with TNFα [26]. This fact is indicative of a role for EGFR in activation of JNK1 and MK2. Since both JNK1 and MK2 are located downstream of MEKK1, we hypothesized that activation of MEKK1 is dependent on both TNFα-signalling and TGFα-signalling. In the model we added a new hypothetical node Hyp 2 (hypothesis 2) to link EGFR to MEKK1 (Suppl. Fig. **??**). This addition led to an improved fit of the model to the data upon treatment with TNFα + C225: activation of both MK2 and JNK1 was strongly suppressed by C225 (Fig. 5C). Stimulation with EGF alone did not lead to activation of JNK1 and MK2. These data support the validity of the modification to the model. Further support for a link between EGFR and MEKK1 was found in literature. Specifically, Ras has been reported as a direct activator of MEKK1 [30]. EGFR is a well-known and potent activator of Ras, which is why it was already in our network [23]. Other studies also report activation of JNK1 and phosphorylation of c-Jun downstream of Ras, which is consistent with an interaction between Ras and MEKK1 [28, 31]. Based on these findings, we adapted our model by removing the Hyp 2 node and creating a direct interaction between Ras and MEKK1 (Fig. 5A). Experimentally, the role of Ras could be confirmed by using a pharmacological inhibitor of Ras activity, and measuring the effect of this inhibitor on the activation of JNK1 and MK2. Together, our model suggests that EGFR activity is required but not sufficient for activation of JNK1 and MK2 in HT-29 cells.

There are other nodes for which the experimental data deviates from the model in one or more of the experimental conditions. A comparison between model and experimental data can be found in Figures **??**, **??** and **??**. A complete deciphering of the signalling events in this biological system is outside the scope of this paper. Instead, we illustrated how interactive modelling of the dynamic behaviour of a signal transduction network can be used to extend previous pathway topologies and can lead to the generation of novel hypotheses.

Testing the performances of ANIMO 2.0

In addition to some new features, ANIMO 2.0 provides a boost in terms of performances (see Methods section for details). In order to measure how significant

the performance increase is, we applied a number of ANIMO analysis queries to a case study, comparing the performances of ANIMO 2.0 with the previous version of the tool. Moreover, ANIMO 2.0 provides the biolgist with a proper access to the formal technique of model checking [32]. As we will show, the performance improvement is even more significant in this case, making this type of formal analysis more accessible.

The case study we use as a testbed is shown in Figure 6, and is a model signaling events downstream of growth factors EGF (epidermal growth factor) and NGF (nerve growth factor) in PC12 cells. The topology for this network was proposed in [33], and it was analyzed with an ANIMO model, reproducing the experimentally observed ERK (extracellular signal-regulated kinase) activity changes [34]. In particular, a 10 minutes stimulation with EGF resulted in transient behavior (i.e. peak-shaped, see also the graph in Fig. 6), while NGF stimulation led to sustained activity (see Fig. 8).

Simulation-based analysis

The most basic analysis that can be performed with ANIMO is based on simulating the behaviour of the network during a given period of time (see also the analyses of the Drosphila and TNF-α/EGF models above). As this type of analysis is also normally used when looking for a match between model parameters and experimental data, the computation speed of a simulation run deeply affects the level of interactivity perceived by the user. In order to make the modeling approach in ANIMO as smooth as possible, it is desirable to decrease dead times, i.e. significantly reduce simulation costs. Our first performance test compares the execution time for simulation runs on the Timed Automata models generated by ANIMO applying the old (i.e., ANIMO 1.0) and new (ANIMO 2.0) modeling approaches to the case study. To define the initial state of the model, we consider the starting condition to be the treatment with 50 ng/ml NGF, which translates into setting the initial activity level of nodes NGF to 15/15, and EGF to 0/15. We have chosen this configuration as the model is expected to continue reacting indefinitely (sustained ERK activity), without reaching a deadlock state where all reactions are inactive. This makes the model more demanding when generating the state space because having more reactions occur means having more states in the transition system underlying the Timed Automata model. Table 2 illustrates the computation time and memory usage when performing 100 simulation runs on each of the two considered models. Computing the simulation runs took about 92% less time with the new version of ANIMO, using 87% less memory. This decrease in computation time for long simulation runs brings the approach nearer to the idea of interactive exploration of a network. This advantage is particularly noticeable when analysing more complex networks.

A good reason to use formal methods: model checking

To show why one should use a formal model like Timed Automata, we demonstrate the application one of the biggest advantages of Timed Automata: model checking [32].

The technique of model checking consists in verifying the truth of some properties that describe the behaviour of a model. This is usually done by exploring every

possible execution path of the model starting from its initial configuration. The set of all the configurations that can be reached by a model during all its possible executions is called *state space*. The properties on the evolution of the model are usually expressed in a formal language such as computation tree logic (CTL, [35]), which allows to formally describe how and when a property should hold. Examples of such properties can be expressed in natural language as: "the activity of node A stays always above 20", "it is possible to reach a state where nodes A and B are both active for at least 60 %", "whenever node A becomes active for at least 20 %, node B becomes active for at least 30 %".

Model checking performances in ANIMO 2.0 have been tested using the case study of EGF-NGF signalling as reference. We compared the execution times and memory requirements in ANIMO and ANIMO 2.0 for a number of interesting queries:

- (1) and (2): `A[] not deadlock`. The model continues to execute indefinitely (`A` refers to all possible paths in the transition system of the model, and `[]` asks the property to always hold along a path).
- (3): `RKIP < 10 --> ERK >= 40`. After RKIP (Raf kinase inhibitory protein) activity has been lowered, ERK activity increases. As in the model RKIP has 20 levels of granularity and ERK has 100 levels, `RKIP < 10` means that RKIP is less than half active, and `ERK >= 40` means that ERK activity is at least 40%.
- (4): `E<> RKIP < 10`. Find a point when RKIP is low (`E` asks for the existence of at least one path for which the property holds, while `<>` requires the property to hold at least once along a given path). This query is expected to generate a trace, the last point of which will be used as initial configuration for model checking queries (5) and (6).
- (5): `A[] ERK < 70` and (6): `A[] ERK > 35`. Once RKIP activity has significantly decreased, ERK activity is sustained at an intermediate level.

The initial conditions are:

- (1): EGF = 15/15 and NGF = 0/15, all others as the original configuration, corresponding to the treatment condition with 100 ng/ml EGF.
- (2) - (4): EGF = 0/15, NGF = 15/15, all others as the original configuration, corresponding to the treatment condition with 50 ng/ml NGF[1].
- (5) and (6): all activities as in the last state of the trace computed from query (4).

The analysis of query (1) returned false and all other queries returned true. In particular, query (1) confirms that under EGF treatment no other activity is observed in the model after the initial peak, while query (2) confirms that with NGF activity continues indefinitely. Moreover, queries (3) - (6) confirm the result of the simulations shown in [34], with NGF treatment leading to sustained ERK activity. The model checking performance of UPPAAL using the reaction- and reactant-centered model types is shown in Table 3.

Analysis of the results

Note that we focus most of our attention on the NGF treatment, as with this configuration the model was hypothesized to keep reacting permanently (sustained

[1]In the laboratory experimental setting, NGF is used at a lower concentration than EGF, but it is still enough to saturate all NGF receptors, which are rarer than EGF receptors.

ERK activity). The results from queries (1) and (2) show that this is indeed the case, as EGF treatment leads to all reactions dying out after a number of steps (i.e., deadlock due to all automata being in the not_reacting location). This means that the queries (3) - (6) deal with a larger state space, leading to presumably lower model checking performances. Requesting a full inspection of the state space as we do when using a query of the type `A[]`ϕ returning true in cases (5) and (6), allows us to indirectly compare the state space size of the two model versions. As the hundreds-fold computation time improvements in Table 3 show, the reactant-centered model produces indeed a noticeably smaller state space, allowing for a higher level of interactivity also when performing non-trivial model checking.

Moreover, our experiments point out that the reactant-centered approach considerably lowers the memory requirements for the model. This is not only due to the absence of possibly large precomputed time tables, which can contain up to ten thousand elements each.[2] Indeed, this point was further investigated by implementing a reaction-centered model which avoids the use of tables and instead makes on-the-fly computations of the time bounds with the same number representation as in the reactant-centered model. This resulted in improved performances in the cases of reachability and simulation-based queries, with memory requirements similar to the ones for the reactant-centered model. However, in all other cases a much larger amount of memory (more than 2 Gb) was used with respect to the table-based implementation of the same model, without leading to appreciable benefits in terms of execution time: in some cases performances were noticeably deteriorated. These findings support the intuitive idea that a reactant-centered approach leads in general to fewer events in a model, and to a smaller state space.

Note that while the network we consider for the case study is not particularly large, the gain in performances is still considerable and promising in the perspective of analyzing more complex networks.

Model checking for bigger models

When a model becomes particularly complex, and it represents many events occurring concurrently, the state space of the model tends to grow exponentially in size, dramatically increasing the time required for its analysis. To avoid this state space explosion, *statistical model checking* is commonly applied. This type of analysis is based on statistical considerations and allows to obtain answers with a given confidence level in far less time than most exact model checking techniques.

User interface for model checking in ANIMO 2

In order to allow a non-expert user to profit from the power of model checking, we have implemented a template-based user interface to define queries directly inside the ANIMO Cytoscape plug-in: Figure 9 shows the interface for composing a model checking query in ANIMO. The mappings between user interface templates and actual model checking queries are inspired on the ones proposed in [36], and are shown in Table 4.

If the answer to a model checking query contains a (counter-) example trace, the trace is automatically parsed by ANIMO and presented to the user in form of a

[2] In the reaction-centered model, having a reaction depend on two 100-levels reactants leads to the generation of two time tables containing $100 \times 100 = 10\,000$ elements each.

graph of activity levels, in the same fashion as is normally done with simulation runs. Finally, a button positioned near the time slider under a simulation graph allows the user to easily change the initial activity levels of the whole network by setting them as in the currently selected time instant. This feature was used after executing query (4) to set the initial conditions for queries (5)-(6). Such an addition makes it easier to inspect the behavior of a network by using a sequence of model checking interrogations.

Discussion

Comparison between ANIMO and other modelling tools

Different formalisms are in use in the field of computational modelling of biological systems, each with their specific characteristics. Many of these formalisms have been implemented into software tools to support modelling efforts. In order to compare ANIMO with existing tools, we have selected a number of mathematical formalisms, each connected to a supporting tool. With an emphasis on the modelling process rather than the final model, we compared these tools on the basis of the following parameters:

1. **Hidden formalism:** a knowledge of the underlying formalism is not required in order to use the tool
2. **Visual modelling:** the tool allows the user to model using a visual interface, and is not exclusively founded on formula-, text- or table-based input forms
3. **Qualitative parameters:** parameters for reactions can be input as approximated estimations, and not exclusively as numbers
4. **Tight coupling with topology:** models are tightly and clearly coupled to the networks they represent, showing the visual representation of the model in a shape similar or comparable to the representation currently used by biologists for signalling pathways
5. **User-chosen granularity:** if discretization is applied during the modelling process, the user can change the granularity with which such discretization is made, possibly for each component of the model separately

Table 1 shows the comparison between ANIMO and the selected tools.

Going beyond the user interface, there are a number of "pros and cons" for using ANIMO and Timed Automata in the biological context. First and foremost, being Timed Automata an executable formal language, a state space can be derived from a Timed Automata model. This means that state space-related analyses such as model checking can be performed on Timed Automata: as explained in the Results section, this can be done directly in ANIMO, bypassing the need for a deeper understanding of Timed Automata while indirectly applying a powerful model checking tool such as UPPAAL.

The smaller amount of parameters needed to build an ANIMO model, if compared e.g. to a differential equation-based approach, makes the modelling process faster, allowing the biologist to obtain interesting results in less time. However, a less parameter-intensive modelling approach is also less precise: in our case we abstract sequences of biochemical reactions into activation or inactivation interactions, simplifying any relation between two entities in a network as either activating or inhibiting. While this allows for some simplifications in the networks and the underlying formal models, it does not allow to represent in full detail more specific

concepts as reactant concentration or reaction affinity. In case such concepts are of primary importance, we encourage the interested reader to turn to more precise modelling approaches, such as those based on differential equations. In particular, all reactants modelled in ANIMO are assumed to be present at the same (unitary) concentration. While this assumption is not always applicable, it encourages abstract thought: many biological processes can be represented as networks driven by *activity-based* interactions. Even if with a limited scope, ANIMO can be applied also in the analysis of metabolic processes, considering the production of protein A as the activation of node A, with degratation being represented as inhibition. As stated before, it cannot be expected from such models to be truly faithful to their respective biological processes, but they can still be an useful tool.

Final remarks

ANIMO is not the first modelling tool to provide an interface to a modelling formalism. Such interfaces exist in many other tools (see Suppl. Tab. 1). With its focus on user-friendliness and intuitive modelling, ANIMO's main contribution lies in making computational modelling more accessible to experts in biology. Making use of the visual interface provided by Cytoscape, network representations subscribe to biological conventions. Model parameters are kept to a minimum and can be directly accessed by mouse-clicking on nodes and edges. Because of the automatic translation of the network topology and user-defined parameters into an underlying formal model, training in the use of formal methods is not needed. In Supplementary Section , a more in-depth comparison between ANIMO and other modeling tools is given. For this comparison we selected a tool for each of the most commonly used formalisms, and used criteria with a strong focus on user-friendliness.

In Section , we described the construction of an ANIMO model of the ciracadian clock in *Drosophila Melanogaster*. This model captured the dynamics of the regulatory network and led to similar conclusions as an ODE model that had been published previously [22]. This finding supports the use of the series of modelling abstractions that we proposed. The biggest difference between the construction of these models is that the model by [22] is constructed by writing a system of mathematical equations, together with an algorithm for simulation. In ANIMO, instead, a number of network nodes is drawn for the molecules involved. These nodes are then linked by directed interactions that represent cause-and-effect relationships, with a single parameter that defines the strength of each interaction. This is a more intuitive approach to construct a model. Further contributing to an interactive modelling process is the compositionality of the model. Each node in the network can be disabled at any time by the user, or extra nodes can be added, without having to change any of the existing interactions.

In Section , we showed the construction of an executable model of signalling events downstream of TNFα and EGF in human colon carcinoma cells. This data set has been used for previous modelling studies, based on partial least-squares regression and fuzzy logic [37, 38]. The partial least-squares regression model describes an abstract data-driven model that uses statistical correlations to relate signal transduction events to various cellular decisions. This type of modelling is very useful in uncovering new and unexpected relations. It is also successful in making predictions, but gives little direct in the dynamic behaviour of the network. Fuzzy logic

analysis led to a model that gave a better fit to the dynamic network behaviour than discrete logic (Boolean) models. Inspection of the inputs to the logical gates that were used to model protein behaviour led to the prediction of novel interactions between proteins, showing the usefulness of this approach. For most of the proteins, such as JNK1, time was used as an input parameter. For example, the logical gates "if TNFα is high *AND* time is low, then JNK1 is high" and "if TNFα is high *AND* time is high, then JNK1 is low" were used to describe the dynamic behaviour of JNK1. Although this leads to a representative description of the dynamic behaviour of JNK1, peaks in protein activity at early time points were not reproduced by the fuzzy logic model. Moreover, it gives no insight in the molecular interactions that are involved in activation or inhibition.

In this study, we used the same data set and performed a single round of the empirical cycle. This cycle starts off with the experiments carried out by [25]. We used the resulting experimental data, together with knowledge from curated databases [23, 24] to construct an executable model of the biological system. In contrast with the two approaches described above, ANIMO is aimed at the construction of more mechanistic models, mimicking biochemical interactions *in silico*. This way of modelling gives a different type of insight. In the process of model construction, we extended a prior-knowledge network with time-dependent extracellular crosstalk that has been reported previously [26]. To come up with possible explanations for a disagreement between the model and the experimental data, two additional layers of crosstalk were introduced, at the signal transduction and transcriptional level. These modifications improved the fit of the model to the data and can be interpreted as novel testable hypotheses. Finally, we proposed new experiments that could be carried out to test these hypotheses, closing the empirical cycle. Together, our model sheds more light on the intricate entanglement between the TNFα and EGF pathways at multiple cellular levels. But above all, the model provides an excellent starting point for further investigation.

Conclusions

We have presented here how the ANIMO tool was improved to provide an interactive approach to model checking. Thanks to the increased performances of the new reactant-centered modeling approach, we are able to obtain answers to model checking queries in a matter of seconds. Coherently with the user friendliness of the tool, we ensure that the biologist can access the features of model checking without the need to directly deal with Timed Automata models. In this way, ANIMO acts as an intermediary between the biologist and a formal representation of biological signaling pathways, letting the experts concentrate on investigating the mechanisms of cellular responses.

In order to enforce the concept of user interaction as a primary focus of the tool, we plan to extend ANIMO by adding support for parameter sensitivity analysis and automatic parameter sweeps. We already allow for only one numeric parameter per reaction, but an automatic way of adjusting such parameters to fit experimental data will allow the user to concentrate more on defining a sensible topology for a pathway. Such an automatic search could be limited by setting specific bounds on reaction rates based on previous knowledge, or on a trade-off with performances.

Moreover, inspired by works on automata learning [39], we plan to add also the possibility to automatically derive a network topology based on experimental data and user-defined restrictions based on previous knowledge.

We aim at widening the available set of model checking queries, in order to allow biologists to perform in silico experiments on an already fitting model and to obtain answers to more relevant questions. This would increase the usefulness of a model as a help to drive wet-lab investigation. In order to allow for meaningful in silico experiments, we plan to purposefully introduce user-defined non-deterministic parts in our models, which would allow for drug dosage investigations through model checking. This can be done e.g. through the definition of intervals for the values of some reaction kinetic constants, adding considerable uncertainty in the timing of those reactions. A model of this type could then be interrogated with queries like "How much X, Y and Z do we need to provide, and at which time points, in order to obtain targets A, B and C to be active?". More useful results can be achieved through proper application of the statistical model checking part of UPPAAL [40] to an extended version of our model to include realistically significant stochastic behavior. Finally, in order to further improve performances, the applicability a multi-core model checking approach based on the works by Dalsgaard et al. [41] is under study.

Methods

A new way of modeling signaling pathways with Timed Automata

We propose here a novel model to represent signaling pathways with Timed Automata. We define the model currently used in ANIMO to be *reaction-centered*, as for each reaction in the network an instance of a Timed Automaton template is generated to mimic the occurrences of that reaction. Observing that signaling networks tend to be highly connected, containing noticeably more reactions than reactants, we propose to shift the focus on reactants instead, achieving what we call a *reactant-centered* model. This change of view is inspired by the classical way in which biological events are modeled with ordinary differential equations (ODEs) [42].

Reactant-centered model

The reactant-centered model presented here is based on the concept of *net effect* of a set of reactions on a reactant: instead of considering each reaction in isolation, we consider their combined influence on each reactant. As an example, consider a reactant A activated by reactions R_1 and R_2, and inhibited by reaction R_3 (see Fig. 7A). The net effect of these three reactions on A defines the *net reaction* $R_A = R_1 + R_2 - R_3$. Applying a concept similar to the definition of an ODE, where the rate of change of each reactant depends on the rate of the reactions influencing it, the rate of R_A is computed as the sum of the rates of the reactions influencing A:

$$r_A = r_1 + r_2 - r_3$$

where r_i is the rate of reaction R_i and is defined as follows. Consider R_1 to be the reaction $B \rightarrow A$ with kinetic constant k_1. Suppose the settings in the ANIMO

network for R_1 make its rate depend only on the activity level of B. Then we compute the rate of R_1 as $r_1 = [B] \times k_1$, with $[B]$ the current activity level of B. We do similarly for the reactions R_2 and R_3.

As with ODEs, if r_A is positive the activity level of A will increase; otherwise, A will decrease its activity level. The absolute value of r_A determines the speed with which such change happens. The value of the reaction rate is thus translated into a time value to be used as time bound in the Timed Automaton representing R_A (see Fig. 7B) by computing

$$T_A = \frac{1}{\text{abs}(r_a)}$$

with $\text{abs}(r_a)$ the absolute value of r_a. In order to represent a natural variability in the timing of reactions, we relax the exact value of T_A by defining bounds of $\pm 5\%$:

$$\begin{aligned} \mathsf{tL} &= T_A \times 0.95 \\ \mathsf{tU} &= T_A \times 1.05 \end{aligned}$$

Analogously to the reaction-centered model of [34], we will call tL the *lower time bound* and tU the *upper time bound*.

Timed Automata model

The behavior of the Timed Automaton representing the net reaction R_A is defined as follows: after a number of seconds variable inside the continuous interval $[\mathsf{tL}, \mathsf{tU}]$ (measured with the local clock c), the activity level of A will increase or decrease by one step, depending on the sign of r_A; then new values for r_A, tL and tU will be computed based on the (possibly) changed conditions.

The process described in Section to compute the values of tL and tU is an instance of the function called update() in the Timed Automaton template of Figure 7B. The computation of update() is the first step taken when initializing a new model taking the transition from the initial location start (identified by two concentric circles) to updating. Location updating is then used to decide whether R_A can be executed or not (functions can_react() and cant_react() used as guards for the transitions exiting from updating). This decision is based on the computed value for r_A and the current activity level of A: if $r_A > 0$ and A is already completely active (or $r_A < 0$ and A is completely inactive), no update to A can be applied, so the automaton switches to location not_reacting. Moreover, r_A can be 0: this means that the reactions influencing A balance each other to complete cancellation, so also in this case R_A cannot occur and the next location will be not_reacting. Otherwise (i.e. if can_react() is true), the activity level of A can be changed, so the next location will be the one labeled waiting, which is reached after having made sure that its clock invariant is respected (committed location between updating and waiting).

When in location waiting, the local clock c is let run until enough time has passed for the reaction to occur: this is obtained by using c >= tL as guard for the transition from waiting to updating. Because of the invariant c <= tU on waiting, we ensure that the reaction occurs after no more than tU seconds have passed. The transition to updating leads to an update to the activity level of A (call to react()), together with the computation of new values for r_A, tL and tU (function react() calls update()).

Moreover, we see in Figure 7B that a communication is sent on channel reacting[0] (0 is the index assigned to reactant A): this is used to warn the other automata that the activity level of A has just changed, possibly changing the conditions for other reactions.

During the course of a model simulation, one of the reactants influencing a reaction on which A depends may change its activity level before R_A can occur: this event may have an influence on the value of r_A, so a re-computation of r_A, tL and tU is needed. In the template of Figure 7B, the indexes of the influencing reactants are 1, 2 and 3, corresponding respectively to B, C, D in Figure 7A. Whenever a communication on one of the channels corresponding to an influencing reactant is received while in location waiting, the automaton moves to location stubborn. This location was introduced to avoid unwanted non-deterministic behavior[3] in the model. In the example, this could happen when the activity level of B is updated concurrently (in the same instant) with the activity level of A. As B influences A through reaction R_1, any change in the activity level of B causes a change in r_1, thus requiring a recomputation of r_A, tL and tU. Due to the interleaving nature of Timed Automata semantics, only one automaton may change location for the Timed Automata system to change state. This means that, should the activity level of B be changed first, the conditions for R_A would be recomputed, possibly preventing the update to A to occur. Vice versa, should the activity level of A be changed first, both A and B could change their activity levels. In such a situation, the non-deterministic order in which the updates are computed decides the behavior of the model. To avoid this, we ensure that if a reaction can occur in a given instant (i.e. c >= tL), it will not consider the concurrent updates to influencing reactants (in the example, having the clock reach tL ensures that R_A occurs, independently from any concurrent update to B). Performing a check on the current value of the clock c with respect to the reaction lower time bound allows an automaton to detect the case of a concurrent update, remaining "stubborn" in its resolution of reacting with the current configuration: the guard c >= tL on the transition returning from stubborn to waiting is the same as the guard on the transition from waiting to updating and implies that the transition could occur in the current time instant. If no conflict is found, i.e. c < tL, a new time limit for the reaction is computed via function update() while moving to location updating. Note that the clock c is not reset when moving from stubborn to updating: should the new conditions only change the value of tL and tU without disabling the reaction, the "partial work" of the reactions is not discarded.

Finally, location not_reacting is used to avoid reacting when not necessary (see the description of cant_react() above), and to respond to changes in those reactants which can influence the value of r_A. Should one of such reactants change its activity level, the conditions are updated and the local clock is reset, moving to updating to check whether the reaction can now occur.

Computations on the fly

Reaction rates are all computed at run-time by function update(), and are based on the user-chosen scenarios, their kinetic constant k and the activity level of

[3] The term *unwanted* refers to a non-determinism that is not related to actual biological concepts, but to the semantics of the modeling paradigm.

the involved reactants. As such computations require floating-point precision but UPPAAL only provides integer variables and operators, we use a significand-and-exponent notation with 4 significant figures, which allows for an error in the order of 0.1% while avoiding integer overflow in UPPAAL's engine. As an example of such a notation, consider the two floating-point numbers $a = 1.23456$ and $b = 0.00023456$, and their sum $c = a + b = 1.23479456$. The corresponding representations in our integer-based notation will be:

$$\begin{aligned} a = \langle 1235, -3 \rangle &= 1235 \times 10^{-3} \\ b = \langle 2346, -7 \rangle &= 2346 \times 10^{-7} \\ a + b = \langle 1237, -3 \rangle &= 1237 \times 10^{-3} \end{aligned}$$

In this case, the number representing the sum $a + b$ differs from the exact result by $c - 1237 \times 10^{-3} = 0.001456$. The interested reader can find the UPPAAL definitions and functions needed to compute rate and time values for the Timed Automata templates inside any UPPAAL model file generated by ANIMO with a reactant-centered model type.

Notes on the Timed Automata model

We note that replacing all the existing reactions with the net reactions relative to each reactant in a model does change the behavior of the model in the sense that continuous oscillations of activity levels around an equilibrium point are less frequent. However, this does not mean that the behavior of two models using the two different approaches will diverge: the equilibrium points are still the same. From the user's point of view, graphs will look smoother while still exhibiting the same qualitative behavior, as can be seen in Figure 8. The Timed Automata templates in Figure 7 show that the main difference between the two approaches is indeed due to the choice of lumping all the reactions influencing A into the single reaction R_A. In the next session we will show that having less oscillations leads in practice to better analysis performances.

Experimental setup for simulations and model checking

All experiments described in the Results section were carried out on an Intel® Core™ i7 CPU at 2.80GHz equipped with 4 Gb RAM and running Ubuntu GNU/Linux 12.10 64bit. UPPAAL version 4.1.11 was used to compute the result of the queries, asking for "some trace" with random depth-first search order when an execution trace was expected to be produced. For the simulation queries using the statistical model checking engine, we left all options at their default values.

Competing interests
The authors declare that they have no competing interests.

Author's contributions
S.S. designed and performed the experiments, developed the Cytoscape integration, wrote the manuscript; J.S. conceived, designed and performed the experiments, wrote the manuscript; B.W. developed the Cytoscape integration, and designed and developed the time slider UI; P.E.vdV. initiated the study, conceived the Cytoscape implementation, supervised the project; R.A.U.C. performed experiments; M.K. designed experiments, analyzed data and wrote the manuscript; R.L. Contributed to the methodology and supervised the project; J.vdP. contributed to the strategy and methodology in the manuscript, in particular the connection with formal methods; J.N.P. designed experiments, analyzed data sets, contributed in particular to the application of ANIMO for large biological data, wrote the manuscript, and supervised the project.

Acknowledgements

We would like to thank Christof Francke for valuable discussions.

Author details

[1]Formal Methods and Tools, Faculty of EEMCS, University of Twente, 7522NH Enschede, The Netherlands. [2]Developmental BioEngineering, MIRA Institute for Biomedical Technology and Technical Medicine, University of Twente, 7522NH Enschede, The Netherlands. [3]Human Media Interaction, Faculty of EEMCS, University of Twente, 7522NH Enschede, The Netherlands.

References

1. Mendoza, L., Thieffry, D., Alvarez-Buylla, E.R.: Genetic control of flower morphogenesis in arabidopsis thaliana: a logical analysis. Bioinformatics **15**(7), 593–606 (1999)
2. Shmulevich, I., Zhang, W.: Binary analysis and optimization-based normalization of gene expression data. Bioinformatics **18**(4), 555–565 (2002)
3. de Jong, H.: Modeling and simulation of genetic regulatory systems: A literature review. Journal of Computational Biology **9**, 67–103 (2002)
4. Fisher, J., Piterman, N., Hubbard, E.J.A., Stern, M.J., Harel, D.: Computational insights into caenorhabditis elegans vulval development. Proceedings of the National Academy of Sciences of the United States of America **102**(6), 1951–1956 (2005)
5. Efroni, S., Harel, D., Cohen, I.R.: Toward rigorous comprehension of biological complexity: Modeling, execution, and visualization of thymic t-cell maturation. Genome Research **13**(11), 2485–2497 (2003)
6. Dematté, L., Priami, C., Romanel, A.: Modelling and simulation of biological processes in BlenX. SIGMETRICS Perform. Eval. Rev. **35**, 32–39 (2008)
7. Ciocchetta, F., Hillston, J.: Bio-PEPA: A framework for the modelling and analysis of biological systems. Theor. Comput. Sci. **410**, 3065–3084 (2009)
8. Siebert, H., Bockmayr, A.: Temporal constraints in the logical analysis of regulatory networks. Theor. Comput. Sci. **391**(3), 258–275 (2008)
9. Bartocci, E., Corradini, F., Merelli, E., Tesei, L.: Model checking biological oscillators. Electronic Notes in Theoretical Computer Science **229**(1), 41–58 (2009). Proceedings of the Second Workshop From Biology to Concurrency and Back (FBTC 2008)
10. Batt, G., Salah, R.B., Maler, O.: On timed models of gene networks. In: Proceedings of the 5th International Conference on Formal Modeling and Analysis of Timed Systems. FORMATS'07, pp. 38–52. Springer, Berlin, Heidelberg (2007)
11. Reisig, W.: Petri nets. In: Koch, I., Reisig, W., Schreiber, F., Dress, A., Vingron, M., Myers, G., Giegerich, R., Fitch, W., Pevzner, P.A. (eds.) Modeling in Systems Biology. Computational Biology, vol. 16, pp. 37–56. Springer, London (2011)
12. Bonzanni, N., Krepska, E., Feenstra, K.A., Fokkink, W., Kielmann, T., Bal, H.E., Heringa, J.: Executing multicellular differentiation: quantitative predictive modelling of c.elegans vulval development. Bioinformatics/computer Applications in The Biosciences **25**, 2049–2056 (2009)
13. Alur, R., Dill, D.L.: A theory of timed automata. Theor. Comput. Sci. **126**(2), 183–235 (1994)
14. Bengtsson, J., Griffioen, W.O.D., Kristoffersen, K.J., Larsen, K.G., Larsson, F., Pettersson, P., Yi, W.: Automated verification of an audio-control protocol using UPPAAL. The Journal of Logic and Algebraic Programming **52-53**(0), 163–181 (2002)
15. Hessel, A., Pettersson, P.: Model-based testing of a wap gateway: an industrial case-study. In: Proceedings of the 11th International Workshop, FMICS 2006 and 5th International Workshop, PDMC Conference on Formal Methods: Applications and Technology. FMICS'06/PDMC'06, pp. 116–131. Springer, Berlin, Heidelberg (2007)
16. Marques Jr., A.P., Ravn, A.P., Srba, J., Vighio, S.: Model-checking web services business activity protocols. International Journal on Software Tools for Technology Transfer **15**(2), 125–147 (2013)
17. Thomas, R.: Boolean formalization of genetic control circuits. J. Theor. Biol. **42**(3), 563–585 (1973)
18. Nakano, S., Yamaguchi, S.: Two modeling methods for signaling pathways with multiple signals using UPPAAL. In: Proceedings of the 2nd International Workshop on Biological Processes & Petri Nets (BioPPN2011), pp. 87–101. CEUR-WS.org, Aachen (2011)
19. Man, K.L., Krilavicius, T., Wan, K., Hughes, D., Lee, K.: Modeling and analysis of radiation therapy system with respiratory compensation using UPPAAL. In: Proceedings of the 9th IEEE International Symposium on Parallel and Distributed Processing with Application - ISPA 2011, pp. 50–54 (2011)
20. Larsen, K.G., Pettersson, P., Yi, W.: UPPAAL in a nutshell. International Journal on Software Tools for Technology Transfer (STTT) **1**, 134–152 (1997)
21. Schivo, S., Scholma, J., Wanders, B., Camacho, R.A.U., van der Vet, P.E., Karperien, M., Langerak, R., van de Pol, J., Post, J.N.: Modelling biological pathway dynamics with Timed Automata. IEEE Journal of Biomedical and Health Informatics **XX**(XX), (2013)
22. Fathallah-Shaykh, H.M., Bona, J.L., Kadener, S.: Mathematical model of the drosophila circadian clock: Loop regulation and transcriptional integration. Biophysical Journal **97**(9), 2399–2408 (2009)
23. Kanehisa, M., Goto, S.: KEGG: Kyoto Encyclopedia of Genes and Genomes. Nucleic Acids Research **28**(1), 27–30 (2000)
24. Hornbeck, P.V., Chabra, I., Kornhauser, J.M., Skrzypek, E., Zhang, B.: PhosphoSite: A bioinformatics resource dedicated to physiological protein phosphorylation. Proteomics **4**(6), 1551–1561 (2004)
25. Gaudet, S., Janes, K.A., Albeck, J.G., Pace, E.A., Lauffenburger, D.A., Sorger, P.K.: A compendium of signals and responses triggered by prodeath and prosurvival cytokines. Mol Cell Proteomics **4**(10), 1569–1590 (2005)
26. Janes, K.A., Gaudet, S., Albeck, J.G., Nielsen, U.B., Lauffenburger, D.A., Sorger, P.K.: The response of human epithelial cells to TNF involves an inducible autocrine cascade. Cell **124**(6), 1225–1239 (2006)
27. Davis, R.J.: Signal transduction by the JNK group of MAP kinases. Cell **103**(2), 239–252 (2000)

28. Bannister, A.J., Brown, H.J., Sutherland, J.A., Kouzarides, T.: Phosphorylation of the c-Fos and c-Jun HOB1 motif stimulates its activation capacity. Nucleic Acids Res **22**(24), 5173–6 (1994)
29. Bailly, S., Fay, M., Israël, N., Gougerot-Pocidalo, M.A.: The transcription factor AP-1 binds to the human interleukin 1 alpha promoter. European Cytokine Network **7**(2), 125–128 (1996)
30. Russell, M., Lange-Carter, C.A., Johnson, G.L.: Direct interaction between Ras and the kinase domain of mitogen-activated protein kinase kinase kinase (MEKK1). Journal of Biological Chemistry **270**(20), 11757–11760 (1995)
31. Dérijard, B., Hibi, M., Wu, I.-H., Barrett, T., Su, B., Deng, T., Karin, M., Davis, R.J.: JNK1: A protein kinase stimulated by uv light and ha-ras that binds and phosphorylates the c-Jun activation domain. Cell **76**(6), 1025–1037 (1994)
32. Clarke, E.: Model checking. In: Ramesh, S., Sivakumar, G. (eds.) Foundations of Software Technology and Theoretical Computer Science. Lecture Notes in Computer Science, vol. 1346, pp. 54–56. Springer, Berlin / Heidelberg (1997)
33. Santos, S.D.M., Verveer, P.J., Bastiaens, P.I.H.: Growth factor-induced MAPK network topology shapes Erk response determining PC-12 cell fate. Nature Cell Biology **9**(3), 324–330 (2007)
34. Schivo, S., Scholma, J., Wanders, B., Urquidi Camacho, R.A., van der Vet, P.E., Karperien, M., Langerak, R., van de Pol, J., Post, J.N.: Modelling biological pathway dynamics with Timed Automata. In: 2012 IEEE 12th International Conference on Bioinformatics & Bioengineering (BIBE), pp. 447–453. IEEE Computer Society, Los Alamitos, CA, USA (2012)
35. Emerson, E.A., Clarke, E.M.: Using branching time temporal logic to synthesize synchronization skeletons. Science of Computer Programming **2**(3), 241–266 (1982)
36. Monteiro, P.T., Ropers, D., Mateescu, R., Freitas, A.T., de Jong, H.: Temporal logic patterns for querying dynamic models of cellular interaction networks. Bioinformatics **24**(16), 227–233 (2008)
37. Janes, K.A., Albeck, J.G., Gaudet, S., Sorger, P.K., Lauffenburger, D.A., Yaffe, M.B.: A systems model of signaling identifies a molecular basis set for cytokine-induced apoptosis. Science **310**(5754), 1646–1653 (2005)
38. Aldridge, B.B., Saez-Rodriguez, J., Muhlich, J.L., Sorger, P.K., Lauffenburger, D.A.: Fuzzy logic analysis of kinase pathway crosstalk in TNF/EGF/Insulin-induced signaling. PLoS Comput Biol **5**(4), 1000340 (2009)
39. Tretmans, J.: Model-based testing and some steps towards test-based modelling. In: Bernardo, M., Issarny, V. (eds.) Formal Methods for Eternal Networked Software Systems. Lecture Notes in Computer Science, vol. 6659, pp. 297–326. Springer, Berlin / Heidelberg (2011)
40. David, A., Larsen, K., Legay, A., Mikučionis, M., Wang, Z.: Time for statistical model checking of real-time systems. In: Gopalakrishnan, G., Qadeer, S. (eds.) Computer Aided Verification. Lecture Notes in Computer Science, vol. 6806, pp. 349–355. Springer, Berlin / Heidelberg (2011)
41. Dalsgaard, A.E., Laarman, A.W., Larsen, K.G., Olesen, M.C., van de Pol, J.C.: Multi-core reachability for timed automata. In: Jurdzinski, M., Nickovic, D. (eds.) 10th International Conference on Formal Modeling and Analysis of Timed Systems, FORMATS 2012, London, UK. Lecture Notes in Computer Science, vol. 7595, pp. 91–106. Springer, London (2012)
42. Daigle, B.J., Srinivasan, B.S., Flannick, J.A., Novak, A.F., Batzoglou, S.: Current progress in static and dynamic modeling of biological networks. In: Choi, S. (ed.) Systems Biology for Signaling Networks. Systems Biology, vol. 1, pp. 13–73. Springer, New York (2010)
43. Ciocchetta, F., Duguid, A., Gilmore, S., Guerriero, M.L., Hillston, J.: The Bio-PEPA Tool Suite. International Conference on Quantitative Evaluation of Systems, 309–310 (2009)
44. Nagasaki, M., Saito, A., Jeong, E., Li, C., Kojima, K., Ikeda, E., Miyano, S.: Cell illustrator 4.0: a computational platform for systems biology. Stud Health Technol Inform **162**, 160–81 (2011)
45. Mendes, P., Hoops, S., Sahle, S., Gauges, R., Dada, J., Kummer, U.: Computational modeling of biochemical networks using COPASI systems biology. Methods in Molecular Biology, vol. 500, pp. 17–59. Humana Press, Totowa, NJ (2009). Chap. 2
46. Chaouiya, C., Remy, E., Mossé, B., Thieffry, D.: Qualitative analysis of regulatory graphs: A computational tool based on a discrete formal framework. In: Benvenuti, L., De Santis, A., Farina, L. (eds.) Positive Systems. Lecture Notes in Control and Information Sciences, vol. 294, pp. 830–832. Springer, Berlin / Heidelberg (2003)
47. de Jong, H., Geiselmann, J., Hernandez, C., Page, M.: Genetic Network Analyzer: qualitative simulation of genetic regulatory networks. Bioinformatics **19**(3), 336–344 (2003)

Figures

Tables

Additional Files

Additional file 1 — Sample additional file title
Additional file descriptions text (including details of how to view the file, if it is in a non-standard format or the file extension). This might refer to a multi-page table or a figure.

Additional file 2 — Sample additional file title
Additional file descriptions text.

Figure 1: Formalization of an activation interaction into a Timed Automata model. **(A)** Classical depiction of a well-studied intracellular signal transduction interaction: MEK activates downstream protein ERK. **(B)** A Timed Automaton model of MEK, consisting of a single location (circle, active_MEK) and a single transition (arrow). In this example, MEK activity is not regulated and MEK is always active. $\mathsf{t} < 20$ is termed an invariant on the location, allowing residence in this location as long as local clock time t is smaller than 20 units. $\mathsf{t} > 18$ is termed a guard on the transition, allowing the transition to take place when local clock t is greater than 18 units. Together, the invariant and guard ensure that the transition must take place within the continuous time interval $18 < \mathsf{t} < 20$. When the transition takes place, the action activate_ERK! is performed and the local clock is reset, $\mathsf{t} := 0$. **(C)** A Timed Automaton model of ERK, consisting of three locations, inactive_ERK (the starting location, indicated by a double circle), 50%_active_ERK and 100%_active_ERK, and two transitions between the locations. Here, ERK has three activity levels: completely inactive, halfway active and completely active. A transition will take place when it is possible to synchronize with the corresponding action activate_ERK! in the MEK automaton. Each synchronization on channel activate_ERK represents the occurrence of the activating interaction between MEK and ERK, and allows ERK to eventually become completely active. If we replace the time constraints for the occurrence of activate_ERK! with variables depending on scenario 1 ($R = k \times [MEK]$), the second activation step would have the same time constraints as the first activation step, since the interaction rate only depends on MEK. If we use scenario 2 ($R = k \times [MEK] \times [1 - ERK]$) instead, the time constraints are doubled after the first activation step, because only 50 % of inactive ERK is left. The second activation step would then take twice the time of the first step.

Figure 2: Incremental construction of an ANIMO model of signal transduction events in human colon carcinoma cells upon stimulation with 100 ng/ml TNFα. Each construction step (top to bottom) is simulated in ANIMO, giving intermediate feedback useful for the piecewise refinement of the model. The graphs on the right show the dynamic behaviour of the corresponding models on the left, comparing it to the measured activity values by [25] (error bars represent the standard deviation). On the vertical axis, "100" represents the maximum protein activity in the complete experiment. The red vertical line in each graph indicates a selected time point in the time course. Nodes in the corresponding network representation are coloured according to their activity at that time point. All images in this figure are taken from the ANIMO user interface. **(A, B)** Basic model showing direct activation of JNK1 and MK2 by TNFα. No peak dynamics are observed because no inactivating processes are present. **(C, D)** The model after addition of inactivating phosphatases and a negative feedback loop that down-regulates TNFR. Note that adding TNFR internalization or phosphatases alone would not be enough to reproduce activity peaks. **(E, F)** The model after addition of IKK, IL1-secretion (abstracting the autocrine IL-1 signalling described by [26]), Casp8 and Casp3, showing the late response to TNFα signalling.
The _data suffix identifies experimental data; all other series are computed by ANIMO.
As the data set did not contain values for cleaved caspase-3, but only for its non-cleaved precursor pro-caspase-3, we computed the Casp3_data series as $100\% - [\text{pro-Casp3}]$.

Figure 3: ANIMO model of the circadian clock in *Drosophila Melanogaster* Autoregulatory negative feedback loops are present on each of the nodes of the network, following the original model by [22]. These feedback loops ensure that protein levels decrease over time when activating inputs are absent. The feedback loops are not represented here for cosmetic reasons and clarity. Naming conventions follow the same rules as in the original model, with lower-case names representing mRNA, and upper-case names indicating proteins.

Figure 4: Signalling network downstream of TNFα and EGF in human colon carcinoma cells. **(A)** The model for the merged TNFα and EGF pathways. Node colours represent the activity level of the corresponding modelled reactants at time $t = 15$ minutes after a stimulation of 100 ng/ml TNFα + 100 ng/ml EGF. **(B)** Modelled production of IL-1α after stimulation with 100 ng/ml TGFα (24 hours). **(C)** Modelled activation of JNK1 and MK2 after stimulation with 5 ng/ml TNFα + 10 μg/ml C225 (2 hours).
The _data suffix identifies experimental data; all other series are computed by ANIMO.

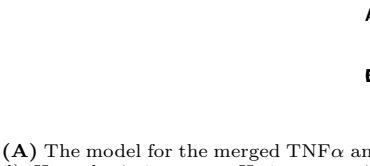

Figure 5: (A) The model for the merged TNFα and EGF pathways after addition of the two hypotheses (highlighted). Hypothesis 1 assumes IL-1α expression to depend on AP-1 activity, which in turn requires both c-Jun en c-Fos to be activated by JNK1 and ERK, respectively. Hypothesis 2 assumes RAS as an activator of MEKK1. Node colours represent the activity levels 15 minutes after stimulation of 100 ng/ml TNFα + 100 ng/ml EGF. **(B)** After the addition of the first hypothesis (activation of IL-1α production depending both on JNK1 and ERK): production of IL-1α after stimulation with 100 ng/ml TNFα (series IL-1a (TNFa)) compared with stimulation with 100 ng/ml TGFα (series IL-1a (TGFa)) (24 hours).
(C) After the addition of the second hypothesis (activation of MEKK1 downstream of EGFR): stimulation with 5 ng/ml TNFα + 10 μg/ml C225 (2 hours). Suppl. Sect. **??** explains how the dosage of 5 ng/ml TNFα was represented in the model.
The _data suffix identifies experimental data; all other series are computed by ANIMO.

Figure 6: The Cytoscape user interface running the ANIMO plug-in (image from [34]). The *Network* panel in the center contains the nodes-edges model of the cross-talking signaling pathways of NGF and EGF, with colors indicating node activity levels and shapes representing different protein categories (see the *Legend* on the left). The *Results Panel* on the right contains a graph plotting activity levels of selected nodes during the first hour of evolution of the model. The slider under the graph allows the user to select the time instant (marked as a vertical red line in the graph) on which the colors nodes in the *Network* are based. The series with the _data suffix is experimental data from [33], considering a treatment with 100 ng/ml EGF.

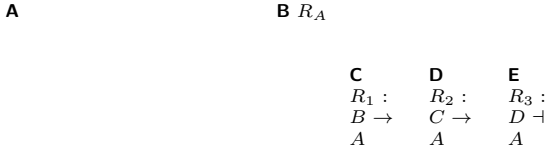

Figure 7: (A) A signaling network where one node is influenced by three distinct reactions. The (virtual) reaction R_A is defined in the reactant-centered model as the algebraic sum of the reactions influencing A. (**B-E**) Timed Automata templates generated by ANIMO for the reactant-centered **(B)** and reaction-centered (**C-E**) models of the network in **(A)**. The templates have been edited for an easier visual comparison. Functions react(), can_react() and cant_react() in (B) have the same content as the corresponding guards and updates in (**C-E**), while function update() in (B) replaces the references to the pre-computed tables tL[], tU[] with on-the-fly computation (see Sect.).

Figure 8: Comparison of two ERK activity graphs generated from the model based on the case study from [34] with treatment of 50 ng/ml NGF. ERK was set to 10 levels of granularity to evidence the oscillations in the graphs.

Figure 9: The interface used in ANIMO to compose a model checking query. The settings on the three lines correspond, from top to bottom, to queries (4), (3) and (6).

Table 1: Comparison between ANIMO and some existing approaches to modelling biological systems. A "Yes" under a column indicates that the modelling tool (mostly) fulfils the parameter, "No" indicates very limited or no fulfilment.

Tool	Formalism	Hidden formalism	Visual modelling	Qualitative parameters	Tight coupling with topology	User-chosen granularity
ANIMO [34]	Timed Automata	Yes	Yes	Yes	Yes	Yes
Bio-PEPA Workbench [43]	Bio-PEPA	No	No	No	No	Yes
Cell Illustrator [44]	Petri Nets	Yes	Yes	No	Yes	No
COPASI [45]	ODE, stochastic models	No	No	No	No	No
COSBI LAB [1]	BlenX	Yes	Yes	No	Yes	No
GINsim [46]	Boolean Networks	No	Yes	Yes	Yes	Yes [2]
GNA [47]	ODE	No	Yes	Yes	Yes	No [3]
Rhapsody [4]	Statecharts	No	Yes	Yes	No [5]	No

[1] COSBILab web page `http://www.cosbi.eu/index.php/research/cosbi-lab`

[2] The user can choose the number of levels for each reactant, allowing to define multi-level models based on Boolean reaction dynamics.

[3] When discretizing an ODE model, the granularity depends on the mathematical features of the model, and not directly on the user's choice.

[4] IBM Rational Rhapsody web page `http://www-01.ibm.com/software/rational/products/rhapsody/designer`

[5] Statecharts represent more closely the so-called *transition system* of the model as opposed to the components and interactions occurring among them.

ANIMO Version	Time (s)	Memory (peak KB)
ANIMO 1.0	37.14	257,928
ANIMO 2.0	2.83	33,464

Table 2: UPPAAL processor time and memory usage for reaction- and reactant-centered modeling approaches when computing the query `simulate 100 [<=36000] { R1, R2, ..., R11 }` on the model from [34] with starting condition NGF = 15/15, EGF = 0/15, corresponding to a treatment with 50 ng/ml NGF. The query asks for 100 time series of the activity levels of all reactants in the model over the first 60 minutes of execution. This means that, on average, it takes 0.37 seconds to perform one simulation of the behavior of the EGF/NGF model with ANIMO 1.0 and 0.028 seconds to compute the same result with ANIMO 2.0.

Query	ANIMO 1.0		ANIMO 2.0		Improvement	
	Computation time (s)	Memory usage (peak KB)	Computation time (s)	Memory usage (peak KB)	Time (n-fold)	Memory (n-fold)
(1)	127.248	351 804	1.415	25 716	90	14
(2)	167.635	269 932	1.932	26 112	87	10
(3)	150.901	318 460	1.439	26 136	105	12
(4)	2.154	216 920	0.113	25 316	19	9
(5)	470.560	311 568	2.223	26 648	212	12
(6)	453.900	357 980	2.229	26 652	204	13

Table 3: UPPAAL processor time and memory usage for ANIMO 1.0 and ANIMO 2.0 when computing the given queries on the case study from [34].

ANIMO template	UPPAAL formula
It is possible for state ϕ to occur	`E<>`ϕ
State ϕ never occurs	`E<>!(`ϕ`)`
If a state ϕ occurs, then it is necessarily followed by a state ψ	ϕ `-->` ψ
A state ϕ can persist indefinitely	`E[]`ϕ
A state ϕ must persist indefinitely	`A[]`ϕ

Table 4: Mapping between queries as presented in ANIMO user interface and the corresponding model checking queries in UPPAAL syntax. State formulas indicated by ϕ and ψ are all in the form $R \bowtie n$, with R the identifier of a reactant in the model, $\bowtie \in \{<, \leq, =, \geq, >\}$ and $n \in [0, g(R)]$ a valid activity level value between 0 and the granularity (number of discrete levels) of R.