

Bringing biological networks to life with ANIMO

Stefano Schivo^{1,*}, Jetse Scholma^{2,*}, Brend Wanders³, Ricardo A. Urquidi Camacho², Paul E. van der Vet³, Marcel Karperien², Rom Langerak¹, Jaco van de Pol^{1,†} and Janine N. Post^{2,‡}

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

ABSTRACT

Motivation: Computational modelling is a vital part of the empirical cycle, formalizing existing knowledge and experimental data in a systematic way. Existing methods and tools are not accessible enough to the biological community, hampering widespread application in biological research.

Results: Here we present an intuitive modeling approach, supported by a user-friendly software tool, ANIMO (Analysis of Networks with Interactive MOdeling). An ANIMO model of signal transduction events downstream of TNF α and EGF in HT-29 human colon carcinoma cells is described as a showcase.

Availability and implementation: ANIMO is implemented as a plug-in to the network visualization tool Cytoscape and can be downloaded from <http://fmt.cs.utwente.nl/tools/animo>. Additional information and a step-by-step installation guide can be found in Supplementary Section 1. A user manual is provided in Supplementary Section 2.

Contact: {j.c.vandepol, j.n.post}@utwente.nl

1 INTRODUCTION

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 obtaining a thorough insight in the functioning of these networks, which is a challenging task. Feedback loops and crosstalk between pathways lead to an intricate wiring of the network. Hence, it is necessary to study the ensemble of molecules involved, because the behaviour of individual molecules is not sufficient for a complete understanding. Since the human brain is ill-suited to grasp the non-linear dynamics of these complex networks and the entailed emergent properties, computational support has a growing role in molecular biology.

The systems biology approach to understanding biological systems starts off from a scientific question and then follows an empirical cycle - or rather a positive spiral - of knowledge/theory → model → hypotheses → experiments → observations → update and/or refinement of knowledge/theory, until an answer to the original question is found (Figure 1). A model plays a pivotal role in this cycle:

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

An *in silico* model is always a simplified representation of biological reality and is never the aim in itself. Rather, it is a powerful means in the process of gaining an understanding of the biological system. Given its role in the empirical cycle, the process of modelling is especially effective when applied by the experts with respect to a certain biological system. Biologists usually have a good sense of cause-and-effect relationships of molecular interactions. In addition, they are the most knowledgeable on the network topology and the dynamics of the biological system they are studying. Since they also benefit most from the generation of hypotheses and from an efficient experimental design, biologists would be the primary candidates to construct models of their research topic.

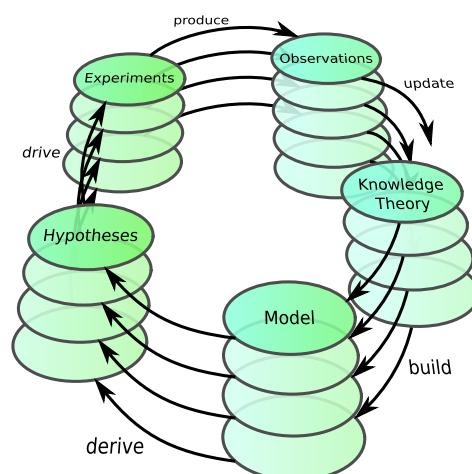


Fig. 1: The empirical spiral.

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 (Mendoza et al., 1999; Shmulevich and Zhang, 2002), ordinary differential equations (ODEs, reviewed by de Jong, 2002),

* These authors contributed equally to this work.

† To whom correspondence should be addressed.

interacting state machines (Fisher et al., 2005; Efroni et al., 2003), process calculi (Dematté et al., 2008; Ciocchetta and Hillston, 2009), Timed Automata (Siebert and Bockmayr, 2008; Bartocci et al., 2009; Batt et al., 2007) and Petri nets (Reisig, 2011; Bonzanni et al., 2009). 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 have been developed to model the dynamic behaviour of systems with processes running in parallel (Alur and Dill, 1994). As such, Timed Automata have been applied in communication protocols and industrial control engineering (Bengtsson et al., 2002; Hessel and Pettersson, 2007; Marques Jr. et al., 2013). The parallels between these application areas and regulatory processes in cells have triggered the step towards their use in biology. Siebert and Bockmayr (2008) use Timed Automata to extend a classical modelling paradigm (Thomas, 1973), allowing to add temporal dynamics to gene network models. Bartocci et al. (2009) describe a model of biological oscillators and test synchronization properties in this dynamic system. A discretization of ODEs to Timed Automata is proposed by Batt et al. (2007), 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 are presented by Nakano and Yamaguchi (2011), who also address the important issue of state space explosion in their paper. Finally, Man et al. (2011) propose an *ad hoc* Timed Automata model of a radiation treatment system, which is then validated through UPPAAL.

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.

Mastery of most existing modelling tools requires training and experience in mathematical modelling. In this respect, a lack of tradition in quantitative reasoning and formal methods within the biological community at large is still a stumbling block for widespread application of modelling of biological systems. Here, we present an intuitive method for the construction of formal *in silico* models of the dynamics of molecular networks, supported by a novel, user friendly modelling tool, ANIMO (Analysis of Networks with Interactive MOdelling, Schivo et al., 2012). Timed Automata are used as the underlying mathematical formalism.

In the Methods section, we will explain how choosing a suitable abstraction level can make the construction of models more intuitive. We will then show how ANIMO is designed to support the modelling process following this approach. Construction of a small model based on experimental data will exemplify the method that we propose. In the Results section, we first show an ANIMO model of the genes and proteins that constitute the circadian clock network in *Drosophila Melanogaster*. The remainder of that section is dedicated to illustrate how a single modelling iteration in the empirical cycle is used to compile prior knowledge and experimental data into a model, perform *in silico* simulations and derive meaningful testable hypotheses. These hypotheses are supported by literature on interactions in different cell types.

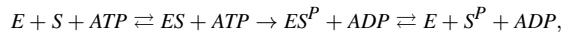
2 METHODS

2.1 Modelling abstractions

In molecular networks in the cell, cascades of chemical and physical interactions enable propagation of signals through the network. In this process, the activity of upstream molecules induces a change in the concentration or activity of downstream molecules. For many reactions, the values of the kinetic parameters are unknown or difficult to collect. This lack of knowledge hampers the feasibility of computational models that describe molecular networks in fine mechanistic detail, especially for larger networks. As a solution to this problem, we propose the construction of models at a higher level of abstraction, thereby reducing the number of parameters involved. In choosing a suitable abstraction level, it is important to retain enough descriptive power to be an asset in the study of biological networks.

As a first abstraction in ANIMO models, the active and inactive forms of each network component are represented together by a single node in the network. Each of these nodes is characterized by its *activity level*, which is a representation of the fraction of active molecules of that species. Activity levels are discretized into integer variables with a user-defined granularity, ranging from Boolean (2 levels) to near-continuous (100 levels).

Detailed biochemical reaction mechanisms are abstracted to *interactions*. These interactions preserve cause-and-effect relationships between molecules and can represent either activations (\rightarrow) or inhibitions (\dashv). Each occurrence of the interaction raises or lowers the activity level of the downstream node by one discrete step, respectively. This aggregation of elementary reactions into single interaction steps, reducesing the number of kinetic parameters involved, while preserving cause-and-effect relationships. For example, consider a reaction in which enzyme E activates substrate kinase S , transferring a phosphate group from a molecule of ATP to a molecule of S . In biochemical terms, the reaction can be represented as



with conservation condition $S_{tot} = S + S^P$.

The same reaction is abstracted in our model by the corresponding interaction



Each occurrence of the interaction $E \rightarrow S$ will increase the activity level of S by one discrete step. The rate R of occurrence of an interaction is defined by the user, who chooses a simplified interaction scenario and a scale factor k . The three available scenarios are based on abstract kinetic formulae:

1. $R = k \times [E]$: the rate of occurrence depends only on the activity level of the upstream node
2. $R = k \times [E] \times [S_{tot} - S^P]$: the rate depends on the activity levels of both participants (note the dependence on the concentration of inactive substrate)
3. $R = k \times [E_1] \times [E_2]$: the rate depends on the activity levels of two user-selected reactants. This formula can be used to represent the so-called *AND gate* kinetics, i.e. the case when the activity of a downstream node depends on the simultaneous presence of two upstream nodes.

The value of the constant k can be either given numerically, or chosen among a pre-defined set of qualitative values, choosing from “very slow”, “slow”, “normal”, “fast” and “very fast”.

We will show in Section 3 that the abstraction proposed here preserves ample descriptiveness to capture experimental data in meaningful models.

2.2 Modelling interactions with Timed Automata

The formal model we propose here is based on Timed Automata (TA, Alur and Dill 1994), which is a formalism developed for modelling dynamic concurrent systems in computer science where timing plays a central role. Being compositional in nature, TA have been shown to be a suitable solution for modelling biological processes (Siebert and Bockmayr, 2008; Bartocci

et al., 2009; Batt et al., 2007). Our model is based on the abstraction presented in Sect. 2.1 and uses the clock mechanism of TA to represent interaction rates: a faster interaction will take less time than a slower interaction to perform one activation/inhibition step.

Figure 2 contains a simplified TA model of the activation of extracellular regulated kinase (ERK) by MAPK ERK kinase (MEK) that illustrates the basic properties of TA.

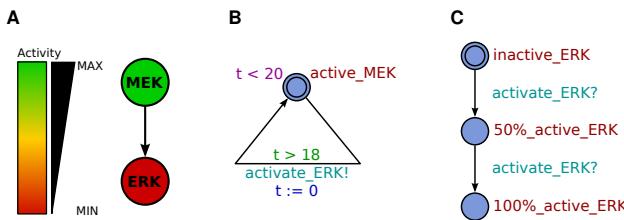


Fig. 2: Parallel between an abstract activation interaction and its TA model. (A) Classical depiction of a well-studied intracellular signal transduction interaction: protein MAPK-ERK kinase (MEK, active) activates downstream protein extracellular-regulated kinase (ERK, initially inactive). In this example, MEK activity is not regulated and ERK has three activity levels, completely inactive, halfway active and completely active. (B) A TA model of active MEK, consisting of one location (circle) and one transition (arrow). $t < 20$ is termed an invariant on the location, allowing residence in this location as long as local clock time t is smaller than 20 units. $t > 18$ 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 in this example ensure that the transition must take place within the (continuous) time interval $18 < t < 20$. When the transition takes place, the action `activate_ERk!` is performed (thus allowing the ERK automaton to reach the `50%_active_ERK` location) and the local clock coupled to this automaton is reset, $t := 0$. This allows the reaction to occur another time, fully activating ERK. (C) A TA model of ERK, consisting of three locations, `inactive_ERK` (the starting location), `50%_active_ERK` and `100%_active_ERK`, and two transitions between the locations. 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.

2.3 ANIMO

The modelling approach described in Section 2.1 is implemented in the software tool ANIMO (Analysis of Networks with Interactive MOdelling, Schivo et al. 2012), a plug-in to the network visualization tool Cytoscape (Killcoyne et al., 2009). The visual interface of ANIMO is designed to allow the user to intuitively construct activity-based models, making the expansion and rewiring of a network a fast and user-friendly process (see Suppl. Video 1). An ANIMO model can be analysed through simulation, with the results automatically plotted as a graph. The dynamic behaviour of a model can then be interactively explored by acting on a slider to highlight time points in a simulation. The selected simulation point defines the appearance of the network: each node will be coloured depending on its activity level at the selected simulation instant. Moreover, experimental data can be matched against the predictions of a model, superposing time-based activity series to a graph produced from the model.

Each network built with ANIMO is automatically translated to a TA model, which is then simulated with the model checking tool UPPAAL (Larsen et al., 1997), translating back the results in the proper user-friendly format (all parts of Figure 3, as well as similar figures in the rest of the paper, were taken from ANIMO's user interface). No training or prior knowledge on the use of TA or UPPAAL is needed in order to benefit from ANIMO. Nevertheless, the translation process can be followed in a transparent manner if desired by the user.

The technical foundations of ANIMO have been described by us elsewhere (Schivo et al., 2012). An abstract overview on the TA model can be found in Supplementary Section 3.

2.4 Using ANIMO to build a model based on data

To illustrate the process of model building in ANIMO, we consider now an example based on a literature compendium of signal transduction events in HT-29 human colon carcinoma cells (Gaudet et al., 2005). This data set comprises triplicate measurements of 11 different protein activities or post-translational modification states at 13 time points after treatment with different combinations of tumour necrosis factor- α (TNF α), epidermal growth factor (EGF) and insulin. The data set contains relative protein levels and activities, and no absolute quantities, which is the typical situation in biochemistry.

As a first step, we normalized measurements for each protein to the maximum value in the complete experiment, resulting in a nondimensionalized data set that is suitable for use with ANIMO.

In Figure 3, we show the stepwise construction of a model of a small part of the network that is able to account for measured variations in activity of inhibitor of nuclear factor kappa-B kinase (IKK), c-Jun N-terminal kinase-1 (JNK1), mitogen-activated protein kinase-activated protein kinase 2 (MK2), Caspase 8 (Casp8) and Caspase 3 (Casp3) upon stimulation with 100 ng/ml TNF α . In this example we aimed for inclusion of a minimum number of nodes in the network, while preserving biological relationships. Multi-step cascades were aggregated into a single step when possible. Parameters for all reactions were set manually, resulting in a close match between the model and the patterns observed in the dataset.

A more comprehensive version of this model is presented in Section 3.2.

3 RESULTS

3.1 Expressivity of ANIMO models

While allowing for a low degree of freedom in setting the parameters of models, ANIMO is still precise enough to describe the behavior of complex networks. The results obtained with ANIMO are comparable to the ones obtainable with other modeling approaches, but the modeling process in ANIMO requires considerably less expertise about the formal foundations. To demonstrate this, Figure 4A represents an ANIMO model of the drosophila circadian clock based on the work by Fathallah-Shaykh et al. (2009), 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 (cycle/clock) protein complex. This is in turn regulated by a series of proteins which are produced as a consequence of CYC/CLK formation. The CWO (clockwork orange) protein is central to the network working, as it degrades the mRNA for most of the involved proteins, thus acting as an inhibitor counterbalancing the action of CYC/CLK. The positive influence of the light-regulated cryptochromes CRY on the degradation of TIM is a consequence of the passage between light and dark, allowing the circadian clock to synchronize to a time zone (see Suppl. Sect. 4.1).

We see in Figures 4B and 4C that the modeling power of ANIMO is sufficient to reproduce the cyclic behavior of the model. In particular, the oscillations exhibited by the two models show the same periods and phases. Moreover, the ANIMO user interface allows for intuitive *in silico* knock-out experiments, by manually disabling a node in the model. Such experiments have been done before (Fathallah-Shaykh et al., 2009) and give similar results in our model.

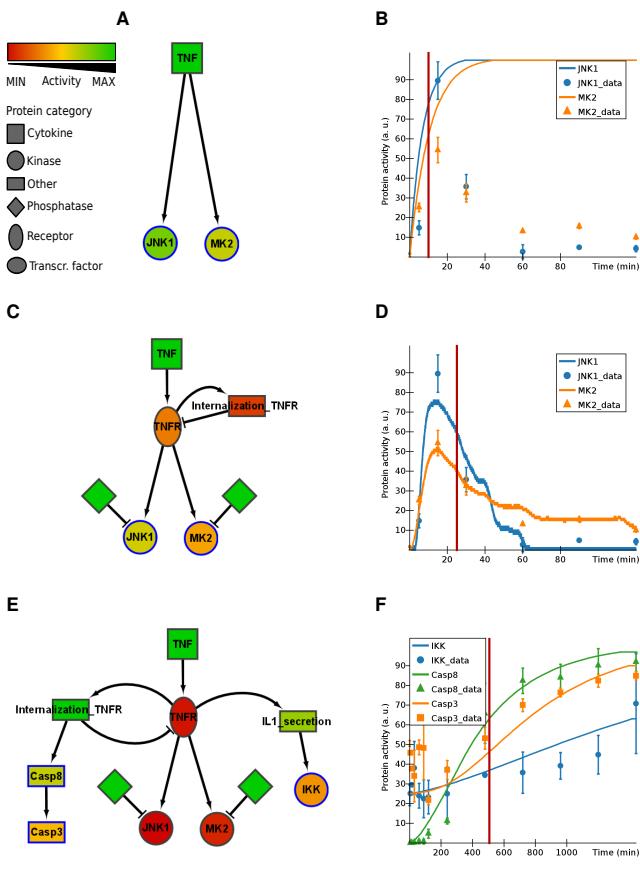


Fig. 3: Incremental construction of an ANIMO model of signal transduction events in human colon carcinoma cells upon stimulation with 100 ng/ml TNF α . Each construction step (top to bottom) is simulated in ANIMO, giving intermediate feedback useful for the piecewise refinement of the model. The graphs on the right show the dynamic behaviour of the corresponding models on the left, comparing it to the measured activity values by Gaudet et al. (2005) (error bars represent the standard deviation). On the vertical axis, “100” represents the maximum protein activity in the complete experiment. A red vertical line in each graph highlights a selected time point in the time course: nodes in the corresponding model are coloured according to their activity at that time point. **(A, B)** Basic model showing direct activation of JNK1 and MK2 by TNF α . 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 (TNF receptor). Note that adding TNF 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 Janes et al., 2006), Casp8 and Casp3, showing the late response to TNF α signalling. As the data set did not contain values for cleaved caspase-3, we computed the Casp3.data series as 100% – [pro-Casp3].

3.2 Case study: using ANIMO to generate hypotheses

In order to validate our modeling approach, we construct a larger model of the signaling network downstream of TNF α and EGF, and formalize crosstalk between the pathways. We first modeled the two pathways in isolation (Figs. S9, S10), using information on protein interactions from the KEGG (Kanehisa and Goto, 2000) and phosphosite (Hornbeck et al., 2004) databases. These models were fitted to the data from the works by Gaudet et al. (2005); Janes et al. (2006). We then merged the two pathways into a single model and added autocrine crosstalk between the pathways that has

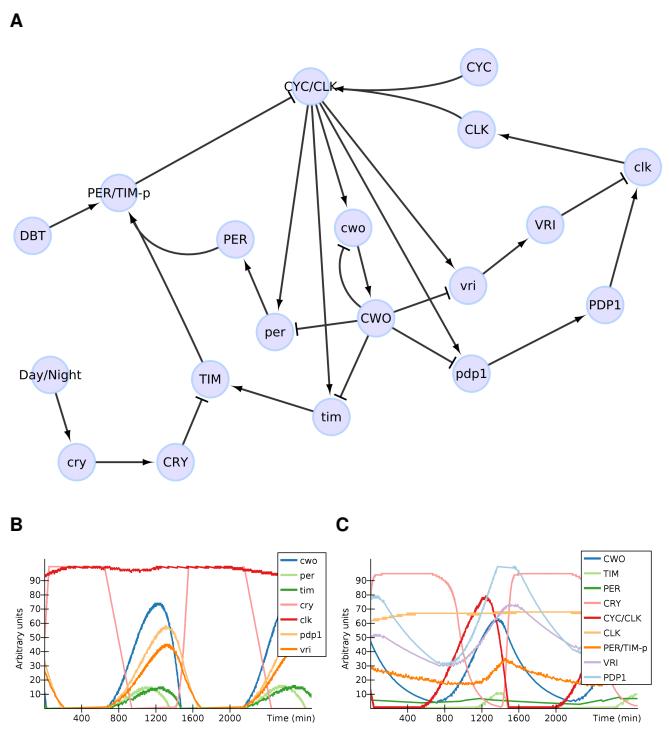


Fig. 4: ANIMO model of the drosophila circadian clock **(A)**. Negative auto-loops on the nodes of the circadian clock network are also present in the model, following the example of the model by Fathallah-Shaykh et al. (2009), but are not represented here only for cosmetic purposes. Naming conventions follow the same rule as in the original model, with lower-case names representing mRNA, and upper-case names indicating proteins.

ANIMO plots of the concentration of mRNA **(B)** and proteins **(C)** over a period of 48 hours. Oscillations are shown here on an absolute scale, while the graphs in (Fathallah-Shaykh et al., 2009) are based on relative scales, where each series was normalized considering local minimum and maximum values.

been reported previously (Janes et al., 2006). Briefly, stimulation with TNF α leads to a rapid release of TGF α (TGF α in the model), which activates the EGF receptor (EGFR). This activation causes secretion of IL-1 α (IL-1 α) at later time points. The effect of IL-1 α is down-regulated by the secretion of IL-1 receptor antagonist (IL-1ra) downstream of TNF α . The resulting model (Fig. S11) was compared to the experimental data for treatments with 100 ng/ml TNF alone and 100 ng/ml EGF alone (data not shown) (Gaudet et al., 2005).

At this point, the behavior of the model deviated from the data for some of the nodes. This is an interesting situation, as it requires modifications to the model, that 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 only TGF α does not lead to secretion of IL-1 α , for which TNF α is also required (Janes et al., 2006). However, in the first version of the model TGF α treatment was sufficient for IL-1 α expression (Fig. 6A). Given the time delay until secretion of IL-1 α , it can be expected that *de novo* synthesis of IL-1 α is required and that both TNF α and TGF α are needed to activate transcription of the IL-1 α gene. JNK1 and ERK signal downstream of TNF α and TGF α , respectively, and

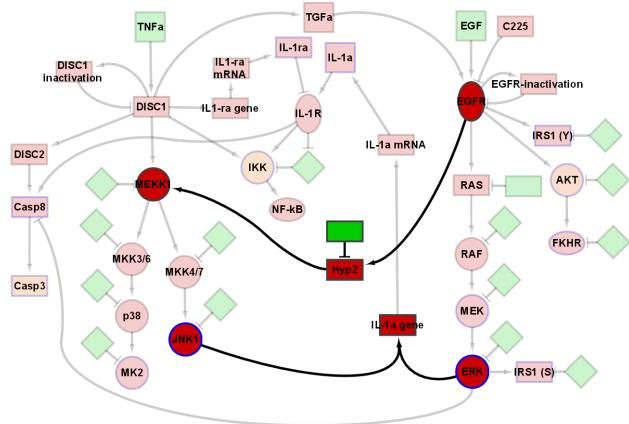
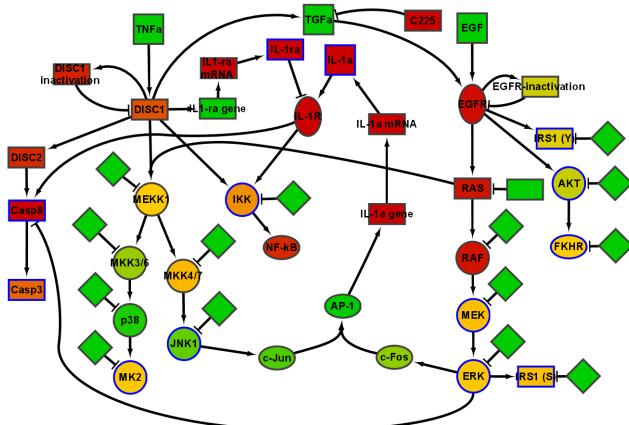
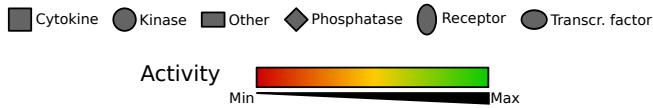
A**B****Protein category**

Fig. 5: Signaling network downstream of TNF α and EGF in human colon carcinoma cells. (A) The merged model for the TNF α -EGF pathway in which the two hypotheses are highlighted. Node colors represent initial activity levels. (B) The TNF α -EGF model with the modified hypotheses. Node colors represent the activity levels 15 minutes after stimulation of 100 ng/ml TNF α + 100 ng/ml EGF.

are known to affect the activity of many transcription factors. We altered the model to make activation of IL-1 α expression dependent on both JNK1 activity and ERK activity (Fig. 5A, arrows linking JNK1 and ERK to IL-1 α gene). After this modification to the model, IL-1 α was no longer secreted upon stimulation with TGF α alone, which greatly improved the fit between the measured IL-1 α levels and the model (see Fig. 6B). This hypothesis of IL-1 α as a target of combined JNK1 activity and ERK activity is supported 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 (Davis, 2000; Bannister et al., 1994). AP-1 has been reported to bind to the promoter of IL-1 α , supporting its role in the

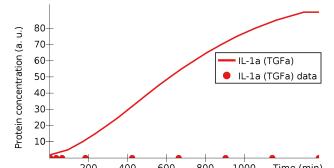
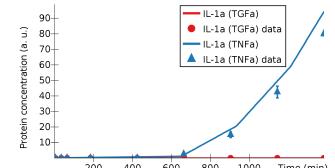
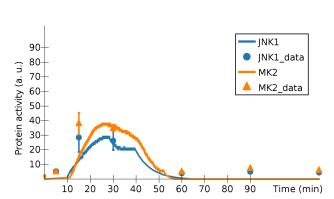
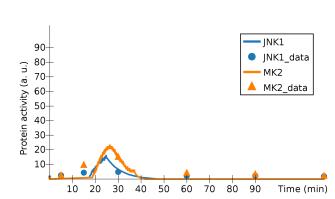
A**B****C****D**

Fig. 6: Comparison of simulation data obtained from the ANIMO model with experimental data. (A) Modeled production of IL-1 α after stimulation with 100 ng/ml TGF α (24 hours). (B) After the addition of the first hypothesis (activation of IL-1 α production depending both on JNK1 and ERK): production of IL-1 α after stimulation with 100 ng/ml TNF α (series IL-1a (TNF α)) compared with stimulation with 100 ng/ml TGF α (series IL-1a (TGF α)) (24 hours).

After the addition of the second hypothesis (activation of MEKK1 downstream of EGFR); (C) stimulation with 5 ng/ml TNF α (2 hours), (D) stimulation with 5 ng/ml TNF α + 10 μ g/ml C225 (2 hours). Without the addition of the second hypothesis, the model behaves in both conditions as in (C).

Suppl. Sect. 4.2 explains how the dosage of 5 ng/ml TNF α was represented in the model.

regulation of IL-1 α expression (Bailly et al., 1996). Based on these findings in literature we included c-Jun and c-Fos in our model as transcriptional activators of IL-1 α (Fig. 5B).

As a second example we considered the behavior of JNK1 and MK2. In the model, both proteins were located downstream of TNF α but not TGF α 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 TNF α . However, experimental data show that C225 strongly reduces activation of JNK1 and MK2 upon stimulation with TNF α (Janes et al., 2006). This fact is indicative of a role for EGFR in activation of JNK1 and MK2. Since both JNK1 and MK2 are located downstream of MAPK/ERK kinase kinase 1 (MEKK1), we hypothesized that activation of MEKK1 is dependent on both TNF α -signaling and TGF α -signaling. In the model we added a new hypothetical node Hyp 2 (hypothesis 2) to link EGFR to MEKK1 (Fig 5A). This addition led to an improved fit of the model to the data upon treatment with TNF α + C225, while preserving the fit for the other conditions (Figs. 6C, 6D). Activation of both MK2 and JNK1 was then strongly suppressed by C225 (Fig. 6D). 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 GTPase-activating protein (Ras) has been reported as a direct activator of MEKK1 (Russell et al., 1995). EGFR is a well-known and potent activator of Ras, which is why it was already in our network (Kanehisa and Goto, 2000). 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 (Bannister et al., 1994; Dérjard et al., 1994). Based on these findings, we adapted our model by removing the Hyp 2 node and creating a direct interaction between Ras and MEKK1 (Fig. 5B). 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 S12, S13 and S14. A complete deciphering of the signaling events in this biological system is outside the scope of this paper. Instead, we illustrated how interactive modeling of the dynamic behavior of a signal transduction network can be used to extend previous pathway topologies and can lead to the generation of novel hypotheses.

4 DISCUSSION

In research projects, the ultimate aim is to solve a problem or get the answer to a scientific question. An in-depth understanding of the biological system is often an important step towards this goal. In this process, the role of modelling is not to produce a model, but to assist in gaining more insight in the system. We developed a new modelling approach and show here how ANIMO allows effective use of this approach in the intuitive construction of formal models.

The novelty of ANIMO is not limited to providing an interface for a modelling formalism: that is already successfully done in many other tools. ANIMO is made to make modelling easier for the biology experts without requiring them to acquire additional training. This is done by providing a visual interface that represents a network similarly to the networks normally found in biology books. Moreover, the lack of precise knowledge on biochemical parameters is tackled by modelling reactions in a less detailed way, while keeping enough precision to provide useful feedback on the evolutions of a modelled network. In Supplementary Section 5 we show a comparison between ANIMO and other tools for modelling biological networks.

In Section 3.1, we described the construction of an ANIMO model with a behaviour similar to an ODE model that had been published previously (Fathallah-Shaykh et al., 2009). The biggest difference between the construction of these models is that the latter is constructed by writing a system of mathematical equations, together with an algorithm for simulation. Drawing a number of network nodes for the molecules involved and linking these nodes with directed cause-and-effect relationships is a more intuitive way of model construction. A single parameter then defines the strength of each interaction. 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 3.2, we showed the construction of an executable model of signalling events downstream of TNF α and EGF in human colon carcinoma cells. The same data set has been used for previous modelling studies, based on partial least-squares regression and fuzzy logic (Janes et al., 2005; Aldridge et al., 2009). 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 insight in dynamic behaviour. 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 TNF α is high AND time is low, then JNK1 is high” and “if TNF α is high AND time is high, then JNK1 is low” were used to describe the dynamic behaviour of JNK1. Although this leads to a useful and representative description of the dynamic behaviour of JNK1, it gives no insight in the dynamics of activation and inactivation. Furthermore, in many cases peaks in protein activity were not reproduced by the fuzzy logic model.

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 (Janes et al., 2006). We then introduced two additional layers of crosstalk, at the signal transduction and transcriptional level, to improve the fit of the model to the data. These modifications can be interpreted as novel testable hypotheses, and are supported in literature. Together, our model sheds more light on the intricate entanglement between the TNF α and EGF pathways at multiple cellular levels and will provide a starting point for further investigation.

ANIMO is leading to a new paradigm for interactive representation of biological networks. Networks in digital textbooks and articles can be displayed as animations amenable to modifications by readers. Repositories of formal descriptions of signalling modules could be used to put together executable signalling networks. A more user-friendly way of interacting with dynamic network models will lead to a more thorough understanding of biological networks and will accelerate hypothesis-driven research.

Acknowledgements

Funding: we acknowledge support from Netherlands Organization for Scientific Research (NWO) CASIMIR to J.S.

Conflict of interest: none declared.

REFERENCES

- Aldridge, B. B., Saez-Rodriguez, J., Muhlich, J. L., Sorger, P. K., and Lauffenburger, D. A. (2009). Fuzzy logic analysis of kinase pathway crosstalk in TNF/EGF/Insulin-induced signaling. *PLoS Comput Biol*, 5(4):e1000340+.
- Alur, R. and Dill, D. L. (1994). A theory of timed automata. *Theor. Comput. Sci.*, 126(2):183–235.
- Bailly, S., Fay, M., Israël, N., and Gougerot-Pocidalo, M. A. (1996). The transcription factor AP-1 binds to the human interleukin 1 alpha promoter. *European Cytokine Network*, 7(2):125–128.
- Bannister, A. J., Brown, H. J., Sutherland, J. A., and Kouzarides, T. (1994). Phosphorylation of the c-Fos and c-Jun HOB1 motif stimulates its activation capacity. *Nucleic Acids Res*, 22(24):5173–6+.
- Bartocci, E., Corradini, F., Merelli, E., and Tesei, L. (2009). Model checking biological oscillators. *Electronic Notes in Theoretical Computer Science*, 229(1):41–58.

- Proceedings of the Second Workshop From Biology to Concurrency and Back (FBTC 2008).
- Batt, G., Salah, R. B., and Maler, O. (2007). On timed models of gene networks. In *Proceedings of the 5th international conference on Formal modeling and analysis of timed systems, FORMATS'07*, pages 38–52, Berlin, Heidelberg. Springer-Verlag.
- Bengtsson, J., Griffioen, W. D., Kristoffersen, K. J., Larsen, K. G., Larsson, F., Pettersson, P., and Yi, W. (2002). Automated verification of an audio-control protocol using uppaal. *The Journal of Logic and Algebraic Programming*, 52-53(0):163 – 181.
- Bonzanni, N., Krepska, E., Feenstra, K. A., Fokkink, W., Kielmann, T., Bal, H. E., and Heringa, J. (2009). Executing multicellular differentiation: quantitative predictive modelling of *C.elegans* vulval development. *Bioinformatics/computer Applications in The Biosciences*, 25:2049–2056.
- Chauvière, C., Remy, E., Mossé, B., and Thieffry, D. (2003). Qualitative analysis of regulatory graphs: A computational tool based on a discrete formal framework. In Benvenuti, L., De Santis, A., and Farina, L., editors, *Positive Systems*, volume 294 of *Lecture Notes in Control and Information Sciences*, pages 830–832. Springer Berlin / Heidelberg.
- Ciocchetta, F., Duguid, A., Gilmore, S., Guerriero, M. L., and Hillston, J. (2009). The Bio-PEPA Tool Suite. *International Conference on Quantitative Evaluation of Systems*, pages 309–310.
- Ciocchetta, F. and Hillston, J. (2009). Bio-PEPA: A framework for the modelling and analysis of biological systems. *Theor. Comput. Sci.*, 410:3065–3084.
- Davis, R. J. (2000). Signal transduction by the JNK group of MAP kinases. *Cell*, 103(2):239–252.
- de Jong, H. (2002). Modeling and simulation of genetic regulatory systems: A literature review. *Journal of Computational Biology*, 9:67–103.
- de Jong, H., Geiselmann, J., Hernandez, C., and Page, M. (2003). Genetic Network Analyzer: qualitative simulation of genetic regulatory networks. *Bioinformatics*, 19(3):336–344.
- Dematte, L., Priami, C., and Romanel, A. (2008). Modelling and simulation of biological processes in BlenX. *SIGMETRICS Perform. Eval. Rev.*, 35:32–39.
- Dérjard, B., Hibi, M., Wu, I.-H., Barrett, T., Su, B., Deng, T., Karin, M., and Davis, R. J. (1994). 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.
- Efroni, S., Harel, D., and Cohen, I. R. (2003). Toward rigorous comprehension of biological complexity: Modeling, execution, and visualization of thymic t-cell maturation. *Genome Research*, 13(11):2485–2497.
- Elowitz, Michael B., Leibler, S. (2000). A synthetic oscillatory network of transcriptional regulators. *Nature*, 403(6767):335.
- Fathallah-Shaykh, H. M., Bona, J. L., and Kadener, S. (2009). Mathematical model of the drosophila circadian clock: Loop regulation and transcriptional integration. *Biophysical Journal*, 97(9):2399 – 2408.
- Fisher, J., Piterman, N., Hubbard, E. J. A., Stern, M. J., and Harel, D. (2005). Computational insights into *caenorhabditis elegans* vulval development. *Proceedings of the National Academy of Sciences of the United States of America*, 102(6):1951–1956.
- Gaudet, S., Janes, K. A., Albeck, J. G., Pace, E. A., Lauffenburger, D. A., and Sorger, P. K. (2005). A compendium of signals and responses triggered by prodeath and prosurvival cytokines. *Mol Cell Proteomics*, 4(10):1569–1590.
- Hessel, A. and Pettersson, P. (2007). 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*, pages 116–131, Berlin, Heidelberg. Springer-Verlag.
- Hornbeck, P. V., Chabra, I., Kornhauser, J. M., Skrzypek, E., and Zhang, B. (2004). PhosphoSite: A bioinformatics resource dedicated to physiological protein phosphorylation. *Proteomics*, 4(6):1551–1561.
- Janes, K. A., Albeck, J. G., Gaudet, S., Sorger, P. K., Lauffenburger, D. A., and Yaffe, M. B. (2005). A systems model of signaling identifies a molecular basis set for cytokine-induced apoptosis. *Science*, 310(5754):1646–1653.
- Janes, K. A., Gaudet, S., Albeck, J. G., Nielsen, U. B., Lauffenburger, D. A., and Sorger, P. K. (2006). The response of human epithelial cells to TNF involves an inducible autocrine cascade. *Cell*, 124(6):1225–1239.
- Kadener, S., Stoleru, D., McDonald, M., Nawathean, P., and Rosbash, M. (2007). Clockwork orange is a transcriptional repressor and a new drosophila circadian pacemaker component. *Genes & Development*, 21(13):1675–1686.
- Kanehisa, M. and Goto, S. (2000). KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Research*, 28(1):27–30.
- Killcoyne, S., Carter, G. W., Smith, J., and Boyle, J. (2009). Cytoscape: a community-based framework for network modeling. *Methods in molecular biology (Clifton,* N.J.), 563:219–239.
- Larsen, K. G., Pettersson, P., and Yi, W. (1997). UPPAAL in a nutshell. *International Journal on Software Tools for Technology Transfer (STTT)*, 1:134–152.
- Man, K., Krilavicius, T., Wan, K., Hughes, D., and Lee, K. (2011). 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*, pages 50–54.
- Marques Jr., A. P., Ravn, A. P., Srba, J., and Vighio, S. (2013). Model-checking web services business activity protocols. *International Journal on Software Tools for Technology Transfer*, 15(2):125–147.
- Mendes, P., Hoops, S., Sahle, S., Gauges, R., Dada, J., and Kummer, U. (2009). Computational modeling of biochemical networks using COPASI systems biology. volume 500 of *Methods in Molecular Biology*, chapter 2, pages 17–59. Humana Press, Totowa, NJ.
- Mendoza, L., Thieffry, D., and Alvarez-Buylla, E. R. (1999). Genetic control of flower morphogenesis in *arabidopsis thaliana*: a logical analysis. *Bioinformatics*, 15(7):593–606.
- Nagasaki, M., Saito, A., Jeong, E., Li, C., Kojima, K., Ikeda, E., and Miyano, S. (2011). Cell illustrator 4.0: a computational platform for systems biology. *Stud Health Technol Inform*, 162:160–81.
- Nakano, S. and Yamaguchi, S. (2011). Two modeling methods for signaling pathways with multiple signals using uppaal. In *Proceedings of the 2nd International Workshop on Biological Processes & Petri Nets (BioPPN2011)*, pages 87–101. CEUR-WS.org.
- Reisig, W. (2011). Petri nets. In Koch, I., Reisig, W., Schreiber, F., Dress, A., Vingron, M., Myers, G., Giegerich, R., Fitch, W., and Pevzner, P. A., editors, *Modeling in Systems Biology*, volume 16 of *Computational Biology*, pages 37–56. Springer London. 10.1007/978-1-84996-474-6_3.
- Russell, M., Lange-Carter, C. A., and Johnson, G. L. (1995). Direct interaction between Ras and the kinase domain of mitogen-activated protein kinase kinase kinase (MEKK1). *Journal of Biological Chemistry*, 270(20):11757–11760.
- Schivo, S., Scholma, J., Wanders, B., Urquidi Camacho, R. A., van der Vet, P. E., Karperien, M., Langerak, R., van de Pol, J., and Post, J. N. (2012). Modelling biological pathway dynamics with Timed Automata. In *2012 IEEE 12th International Conference on Bioinformatics & Bioengineering (BIBE)*, pages 447–453, Los Alamitos, CA, USA. IEEE Computer Society.
- Shmulevich, I. and Zhang, W. (2002). Binary analysis and optimization-based normalization of gene expression data. *Bioinformatics*, 18(4):555–565.
- Siebert, H. and Bockmayr, A. (2008). Temporal constraints in the logical analysis of regulatory networks. *Theor. Comput. Sci.*, 391(3):258–275.
- Thomas, R. (1973). Boolean formalization of genetic control circuits. *J. Theor. Biol.*, 42(3):563–585.

Bringing biological networks to life with ANIMO

Additional Materials

CONTENTS

1 Requirements and installation	1
1.1 Java	1
1.2 Cytoscape	2
1.3 UPPAAL	2
1.4 Installing ANIMO	3
2 ANIMO user's manual	4
2.1 Modelling a small network	4
2.2 Additional tips	8
3 ANIMO and Timed Automata	11
3.1 Timed Automata model	11
3.2 Granularity of an ANIMO network node	13
4 Additional notes	14
4.1 Simulating the day-night cycle	14
4.2 Note on the parameters in the TNF α -EGF model	14
5 Comparison between ANIMO and other modelling tools	15
6 Supplementary figures	17

1 REQUIREMENTS AND INSTALLATION

In order to run ANIMO, a desktop or laptop computer is needed with the following software installed:

- Java: see Section 1.1
- Cytoscape (Killcoyne et al., 2009): see Section 1.2
- UPPAAL (Larsen et al., 1997): see Section 1.3

Each works under Windows () Mac-OS () and all most common GNU/Linux () distributions. If the requirements are already met, ANIMO can be directly installed following the instructions in Section 1.4.

Note: when required to type something, the text to input will be represented “like this”: the quotation marks are not intended to be typed.

1.1 Java

1. In order to check that Java is installed, open a console

 Windows 7: press Windows button and type “cmd”, then press Return. Previous versions: in the Start menu find *All programs* → *Accessories* → *Command Prompt*.

 Go to *Applications* → *Utilities* → *Terminal*.

 Under Gnome, press Alt-F2, type “gnome-terminal”, then press Return. Under KDE, press the KMenu button, type “konsole” and click Konsole. Under Unity, press the home button (the one with the Ubuntu logo: ) type “terminal” and click the Terminal icon.

2. Type “java” and press Return. If a brief error message like “unknown command” is shown, Java needs to be installed: please proceed to step 3. Otherwise, please continue to Section 1.2.

3.  Point your web browser to java.sun.com.

Click *Downloads*, then on *Java SE* (can be usually found under *Popular Downloads*, or by performing a search inside the text of the web page).

Choose *Java Platform (JDK)*, and select *Accept License Agreement*.

Choose the correct package to download, based on your platform: if you have a 32 bit installation, choose the version named *Windows x86*, otherwise choose the *Windows x64* package.

After the download, double click the installer and install Java. You don’t need to register with Oracle, so you can close the browser when it opens at the end of the installation.

To check that Java has been correctly installed, open the console (see the instructions of step 1 in Section 1.1) and type “java”, followed by Return. If you get the message “unknown command”, read on. Otherwise, skip the rest of this section and continue with the installation of Cytoscape (Sec. 1.2).

Open *My Computer*, and enter the local disc (C). Usually, Java is installed in “Program Files\Java”. In that directory, there should be a directory named *jdk1.6.nnnn* with nnnn = some numbers. Inside this directory, there is a *bin* directory. Copy the complete path to that *bin* directory (it should be something similar to “C:\Program Files\Java\jdk1.6.0_26\bin”). Go to Control Panel (Start → Settings → Control Panel) and look for the *System* icon. In the System properties window, select the *Advanced* tab, and then click the *Environment variables* button. In the upper part of the new window, look for a

property named *PATH*. If present, double click it, otherwise click the button *New* (pay attention that you focus on the **upper** part of the window). Check that *Variable name* says *PATH*. If you have clicked *New*, you must insert the word “*PATH*” yourself. *Variable value* can correspond to an empty field, and in that case you have to paste the path you copied at the previous step. If there is something else in that field, go to the end of the field, add a “;” and paste the path after the “;”.

Click *OK* in both the *Environment variables* and *System properties* windows.

Open a **new** console (using one that is already open will not update the *PATH* variable), and try again to type “*java*” and press *Return*. This time, a longer message should be shown instead of the “*unknown command*” message. Java has been successfully installed.



Go to *Applications* → *Utilities* → *Java Preferences*.

If the Java Preferences window is shown, Java is already installed, otherwise the system will prompt you to install it. Follow the instructions and Java will be correctly installed at the end of the procedure.



If you run Ubuntu, open the Software centre, search for “*java*” and select *OpenJDK Java 6 Runtime*. An *Install* button will appear next to the name of the package: click that button and Java will be correctly installed.

If you run another distribution, use your package manager in a similar way. If you cannot find OpenJDK, there may be the possibility to install *Sun Java Development Kit (JDK)* instead.

1.2 Cytoscape

Cytoscape can be found at the address www.cytoscape.org/download.html: an automatic installer program can be downloaded. Please note that you need to accept Cytoscape’s terms of use before being able to start the download. Choose the latest version (at least 2.8.3), possibly using a platform specific installer. For Windows, you can choose *64bit* only if you know that your computer can run 64bit programs, otherwise it is safe to choose *32bit*.

1.3 UPPAAL

1. Point your browser to www.uppaal.org.

Note: UPPAAL is free only for academic use. Information and contacts for commercial licenses can be found on the web site.

2. Click the *Download* link, and choose the latest *development* version (at least 4.1) for your operating system.

3. Fill in the required contact information and click the *Accept and download* button to download UPPAAL.

Note: problems with the registration on UPPAAL website have been reported when using some versions of Internet Explorer. If the registration is unsuccessful, please consider changing your web browser.

4. Unzip the downloaded file to a known location: UPPAAL will be installed there.

5. Complete UPPAAL installation.



Open the UPPAAL installation location in Finder, drop the *UPPAAL.App* icon in your *Applications* folder, and copy the *verifyta* executable file to a known location. The installation of UPPAAL is complete: go to Section 1.4.



Open a console (this was done in Sec. 1.1, step number 1), type “cd PATH_TO_THE_UPPAAL_DIRECTORY” and press Return; PATH_TO_THE_UPPAAL_DIRECTORY is the path to the directory where you installed UPPAAL. It can be for example “c:\Users\myuser\Desktop\uppaal-4.1.7”, or “/home/myuser/programs/uppaal-4.1.7”.

Note: some Windows users may have access only to specific partitions (D:, Z:,...): in that case, please first change to the corresponding drive letter where the downloaded file was extracted. For example: if UPPAAL is located in d:\myuser\Programs\uppaal-4.1.7, the two commands to be entered are
“d:”
“cd \user\Programs\uppaal-4.1.7”

6. Type “java -jar uppaal.jar” and press Return.
7. The license for UPPAAL will be automatically acquired, and the main window of UPPAAL user interface will appear: you may now close that window.

1.4 Installing ANIMO

1. ANIMO is **free only for academic use**. For commercial licenses, please contact us.
2. Run Cytoscape.
3. Click the menu command *Plugins → Manage Plugins*: the *Manage Plugins* window will open.
4. Select the *Settings tab* and press the *Add* button.
5. Insert this *Name*: “ANIMO”, and this *URL*: “<http://fmt.cs.utwente.nl/tools/animoplugins.xml>” (please note: the “<http://>” is required), then confirm with *OK*.
6. From the *Download Sites* list in the upper part of the *Manage Plugins* window, select *ANIMO* (you may need to scroll down: it should appear after *Cytoscape*).
7. The panel on the left shows a smaller list of plugins: under *Available for Install → Analysis*, select *ANIMO v1.28*.
8. Click the *Install* button. The ANIMO tool will be automatically downloaded and installed.
9. The tool will ask you to indicate the position of the *verifyta* executable, which is the tool to verify Timed Automata models. You can find it in the *bin* (*bin-Linux*, *bin-Win32*, ...) depending on your operating system) directory inside the UPPAAL installation directory, or where it was copied at step 5 for in Section 1.3.
10. Click the *Close* button to close the *Manage Plugins* window.
11. ANIMO is correctly installed and ready to be used.

2 ANIMO USER'S MANUAL

We will now present a step-by-step sequence to obtain an example model with ANIMO, which will allow us to illustrate the main features of the tool.

2.1 Modelling a small network

1. Run Cytoscape.
2. If Cytoscape is already running and there are open documents, please make sure that the current work is saved before proceeding.
3. From the *File* menu, select *New → Session*. Answer positively to the question “*Current session (all networks/attributes) will be lost. Do you want to continue?*”.
4. From the *File* menu, select *New → Network → Empty Network*.
5. In the *Control Panel* find the *Editor* tab. If you cannot find it, click the black arrows on the top right of the panel to search through the available tabs. Click the name of the tab to activate it.
6. Add 5 nodes to the empty network by Ctrl-clicking on empty areas of the *Network* window.
Note: *Ctrl-clicks* are obtained as follows. While holding the *Ctrl* key down, click with the left mouse button, then release the *Ctrl* key. The *Ctrl* key is usually located in the lower left or lower right corner of the keyboard. Apple keyboards may have the *⌘* symbol instead of *Ctrl*.
7. The *Edit reactant* dialogue window is opened when a new node is added, or when you right click an existing node and then select the *[ANIMO] Edit reactant...* item from the menu. Use that window to set the properties of the nodes as indicated in Table S1, taking the setting in Figure S1 for node A as reference. When the properties of a node have been inserted, confirm the choice with the *Save* button.

Table S1. The settings for the nodes (signalling network components) in the example.

Name	Total act. levels	Initial act. level	Molecule type	Enabled?	Plotted?
A	15	15	Cytokine	Yes	No
B	15	0	Receptor	Yes	No
C	15	0	Phosphatase	Yes	No
D	100	0	Kinase	Yes	Yes
E	1	1	Phosphatase	Yes	No

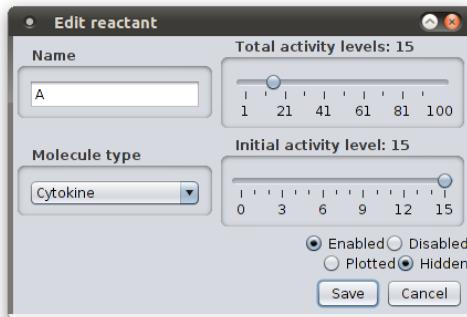


Figure S1: The *Edit reactant* window: modifying the properties of node A.

8. In order to add edges to the network, make sure that the *Editor* tab is still active in the *Control Panel*, and add the following edges by Ctrl-clicking the source and then clicking the target: A → B, B → C, C → B, B → D, E → D.
9. The *Edit reaction* dialogue window is opened when you add a new edge, or when you right click an existing edge and then select the [*IANIMO*] *Edit reaction...* item from the menu. Use that window to set the parameters of the edges as indicated in Table S2. The settings for the edge A → B should reflect the ones shown in the *Edit reaction* window in Figure S2.

Note: In order to insert a qualitative parameter like the ones required by the example network, click once the slider in the *parameter* box to activate it, and then move the slider to match the requested value.

Table S2. The settings for the edges (reactions) in the example.

Reaction	Scenario	Parameter value	Influence
A → B	1	Fast	Activation
B → C	1	Medium	Activation
C → B	1	V. Fast	Inhibition
B → D	1	Medium	Activation
E → D	2	0.00015	Inhibition

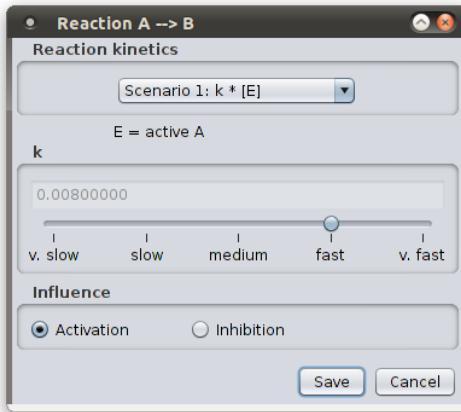


Figure S2: The *Edit reaction* window: modifying the properties of edge $A \rightarrow B$.

10. In the *Control Panel* activate the *ANIMO* tab by clicking its title.
11. Click the *Choose seconds/step* button. A new dialogue window will appear: you can safely choose a time resolution of 12 seconds per step and click *OK*.
12. Click the *Analyse network* button.
13. After a few seconds the *Results Panel* should appear on the right, showing a plot of the activity level of reactant D over a time course of 240 minutes. Figure S3 shows the resulting network and graph plot.

2.1.1 Managing simulation data and activity levels plots Each time a simulation result is obtained, a new tab is added to the *Results Panel* (see the right part of Fig. S3) in which we identify three buttons, a plot of the activity levels of the selected reactants and a time slider.

Clicking on the button *Change title* allows to select a new title for the tab: this can be useful e.g. when comparing different simulations made on similar configurations of the same network. Button *Save simulation data...* allows to save the simulation data of the current tab on a *.sim* file, which can then be loaded and inspected in the future. The *Load simulation data...* button in the *Control Panel* above the *Simulation* box can be used for this purpose. Please note that the best results are obtained only when loading a *.sim* file when the *Network* window contains the same network on which the simulation data are based. If no network is currently opened, a network will *not* be opened by loading a *.sim* file. The *Close* button is used to close the currently displayed results tab.

Right clicking inside the graph area will bring up a menu that allows to perform some basic operations with the graph and its data:

- *Add data from CSV*: superpose the graph with other data series found in a *.csv* (comma separated values) file. This file type can be obtained for example by exporting data from the default Excel format. If you want the data in the *.csv* file to be rescaled so that its maximum Y value coincides with the maximum in the plot, the data file needs to contain a column named (exactly) *Number_of_levels*,

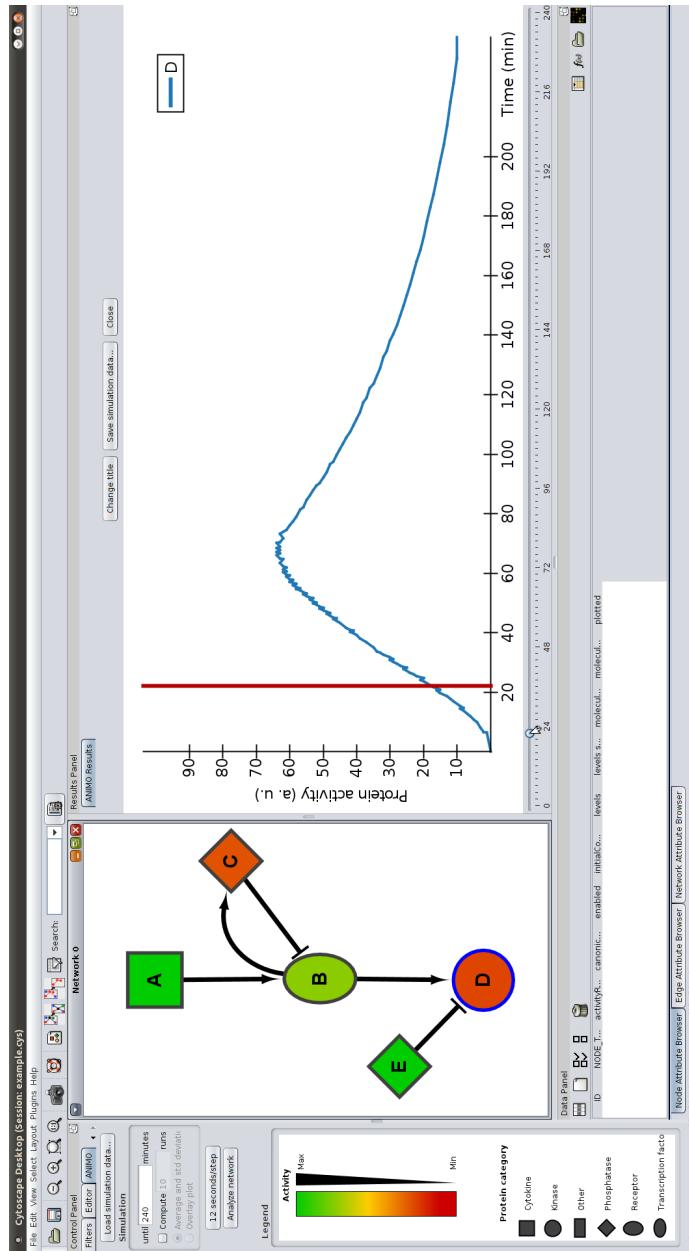


Figure S3: The completed example, where also the feature that allows to view the activity levels of reactants at chosen simulation times is demonstrated: the vertical red bar in the graph on the right can be moved through the slider under the graph, and indicates the point in the time series on which the colouring of the nodes in the *Network* window is based. The legends for colours and shapes can be found in the *Legend* panel.

on the first row of which the maximum of the scale for the *.csv* data needs to be put. For example, if the data in the *.csv* file are on a 0-100 scale, the value for *Number_of_levels* will be 100.

- *Save as PNG*: save the graph as it is shown in a *.png* image file. This file format can be opened by most image editors.
- *Export visible as CSV*: export to a *.csv* file all the series that are currently visible (i.e., not hidden) in the graph.
- *Clear Data*: clear the contents of the graph, removing all series. This can be useful for plotting a *.csv* file without superposing it to the current graph, or for loading a file in which all hidden data were removed (exporting the visible graph to a *.csv* with the previous command).
- *Graph interval*: change the lower and upper bounds for X and Y axes.
- *Zoom rectangle*: zoom the graph around a user-chosen rectangular area. After selecting this command the shape of the mouse cursor changes into a cross. The area of the plot to be zoomed can then be selected by dragging a rectangular selection around it (see the definition of *rectangular selection* on page 8).
- *Zoom extents*: bring the zoom level back to default, cancelling the effects of any *Zoom rectangle* command.

Whenever the result of one or more simulations is shown as a graph, it is possible to use the slider under the graph to move through the entire simulation, showing the activity levels of all reactants represented with different node colouring in the *Network* window on the left. For an example, see Figure S3: the vertical red line in the graph represents the time instant on which the colours of the nodes in the *Network* window are based, and can be moved with the slider over which the mouse cursor is drawn.

2.2 Additional tips

2.2.1 Editing a network in Cytoscape Nodes and arcs can be placed in the network as shown previously: with the *Editor* tab selected in the *Control Panel*, Ctrl-click (click while holding the **Ctrl** or **⌘** key) in an empty place to add a node; Ctrl-click the source node and click the target node to add an arc. It is also possible to drag and drop the node/arc icons from the *Control Panel* into the *Network* window. *Note*: to perform *drag and drop* move the mouse cursor over the icon in the *Control Panel*, click with the left mouse button and, without releasing the button, drag the mouse cursor on the *Network* window where you want to place the symbol; then release the left mouse button.

In order to delete a node/edge, select it by clicking or grouping them in a larger rectangular selection, and then press the *Delete* key on the keyboard or select the *Edit → Delete Selected Nodes and Edges* menu command.

Note: in order to obtain a *rectangular selection*, left click in the *Network* window where the upper-left corner of the rectangle should be and, without releasing the left mouse button, drag the mouse cursor to where the lower-right corner of the rectangular selection should be; then release the left mouse button. All the entities which were *even partially* touched by the rectangle are now selected.

Navigation inside the *Network* window can be performed by clicking and dragging the centre mouse button, while zooming can be done by either rotating the mouse wheel or clicking and holding the right mouse button while moving the mouse in a vertical direction.

Finally, note that the colours used to represent node activity can be changed using the *VizMapper* interface provided by Cytoscape, changing the setting for *Node colour*, shown in Figure S4. To change the node colours, activate the *VizMapperTM* tab in the *Control Panel*, and find the entry named *Node Colour* in the *Visual Mapping Browser* box. Click the arrow-shaped icon directly to the left of *Node*

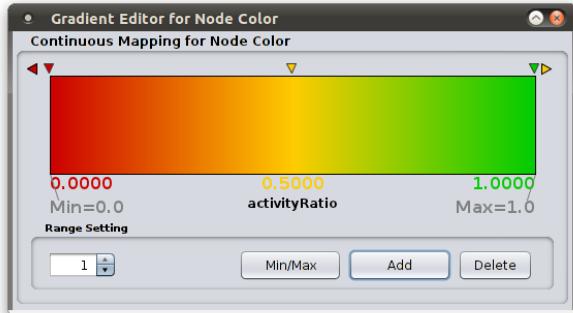


Figure S4: Changing the colours used to represent node activity in an ANIMO model.

Colour: the current setting for the node colours should appear: click the coloured bar to open the window shown in Figure S4. The *activityRatio* on which the node colouring is based is the ratio between the current activity level of a node and its number of activity levels. The coloured arrows pointing downward on the upper border of the coloured rectangle can be dragged along the length of the $[0, 1]$ interval, thus changing the point at which a particular colour appears. New arrows can be added with the *Add* button, while clicking an arrow and pressing the *Delete* button will remove one. To change the colour of a point in the interval, double click the corresponding arrow, and a new window will open, allowing you to choose a new colour: clicking *OK* will accept the new colour. The modifications made in the *Gradient Editor for Node Colour* window should be automatically reflected in the model: when the gradient is as wanted, simply close the window. If the node colours seem not to have been updated, please move the slider under an existing graph in the *Results Panel*.

2.2.2 ANIMO features Nodes and edges (and groups of nodes/edges highlighted via a rectangular selection) can be disabled by choosing *[ANIMO] Enable/disable* from the right-click menu: they will be represented with less saturated colours and can be re-enabled by performing the same action. Moreover, a node can be enabled/disabled directly in its properties window, where it is also possible to add/remove the node from the list of series appearing in the graph resulting from a simulation of the network by selecting *Plotted* or *Hidden* (see also Figure S1): nodes that will be plotted are circled in blue. Every enabled node will be taken into account when computing the evolution of the system, but only nodes marked as *Plotted* will appear in a graph.

Each plot in an *ANIMO Results* tab contains by default a legend, which can be used to modify which series are displayed and how they are displayed. Clicking with the central mouse button on a series name will hide it from the graph, while the same centre-click the coloured line beside the series name will change the colour of that series, cycling through a predefined set of available colours. The entire legend can be hidden by clicking with the central mouse button anywhere on the graph (not inside the legend), or it can be dragged around by clicking and holding the left mouse button, releasing it when the preferred position is reached. Rotating the mouse wheel will allow the thickness of all the graph lines, and the size of the text, to grow or decrease: this feature can be useful when the window containing the graph is very large.

As the model is non-deterministic, i.e. its evolution will not be exactly the same for every single simulation run, it is possible to ask ANIMO to perform a number of simulation runs in a batch, plotting

the averages of the activity levels over the runs together with a standard deviation value, or showing a so-called *overlay plot* where all runs are plotted over each other. The controls that allow to ask for multiple simulation runs can be found in the *Control Panel*, inside the *Simulation* box.

Standard deviation may be represented in the graph: it is normally shown as vertical bars, but its aspect can be cycled through five possibilities (vertical bars, shading, both bars and shading, bars and symbols, none) by right-clicking the corresponding line in the legend. Symbols associated to a representation of standard deviation can be changed by Shift-right-clicking (holding down the *Shift* key, right click) on the corresponding line in the legend. Standard deviation values can be obtained when asking for multiple simulations in the network analysis, but they can also be present in a .csv file, e.g. when the file contains averages of experimental data. In a .csv file, the column containing the standard deviation values for column *A* should be named *A_StdDev* for the program to recognize it and properly display the data series with the associated error values.

2.2.3 Parameter settings The application of some basic strategies when setting the parameters for a network allows the less experienced users to considerably shorten the modelling time. First of all, it is important to proceed in a *top-to-bottom* order, trying to match a component to the corresponding data before inserting the components downstream thereof. Second, when choosing the kinetic parameter for a reaction, we advise to first use the qualitative settings (very slow, slow, medium, fast, very fast): this allows to define the relative speeds of the reactions as soon as possible, leaving the more precise parameter setting procedure as a follow-up step. Finally, as can be seen from the parameter settings of Section 2.1, in order to obtain a peak behaviour it is particularly important that a negative feedback is present (as an example, see the reactions involving B and C in Tab. S2), and that the inactivating reaction in the loop is faster than the ones activating the target node.

A final note on the *seconds/step* button. This button allows to define the time granularity of the simulations, but it is not strictly necessary to choose a very precise value. If the current value for *seconds/step* is too high (or too low) to allow the network to be properly simulated, ANIMO will automatically choose (respectively) the highest (lowest) value that still allows to avoid rounding problems. It will be possible to notice such a change in the value of time scale when the number on the *seconds/step* button changes.

2.2.4 Updating ANIMO To check whether a new version of ANIMO has been published, run Cytoscape and ask for an update of all plug-ins via the menu command *Plugins* → *Update Plugins*. After some seconds during which Cytoscape will contact all the providers of the installed plug-ins, the system should report the list of updatable plug-ins.

Note: a window with the message *Attempting to connect to XYZ...* may appear and disappear multiple times: it is the normal behaviour.

If an updated version of ANIMO is available, it will appear under the category *Updatable Plugins* → *Analysis* → *ANIMO v1.28*. If no plug-in can be updated, a message stating *No updates available for currently installed plug-ins*. will be shown.

3 ANIMO AND TIMED AUTOMATA

3.1 Timed Automata model

The Timed Automata (TA) model underlying an ANIMO network is generated whenever an analysis is requested by the user. Starting from the network represented in the Cytoscape-based user interface, ANIMO automatically generates a TA model to be used with UPPAAL. The analysis result is then parsed and properly presented to the user, for example as a graph of reactant activity levels. This workflow is described in Figure S5.

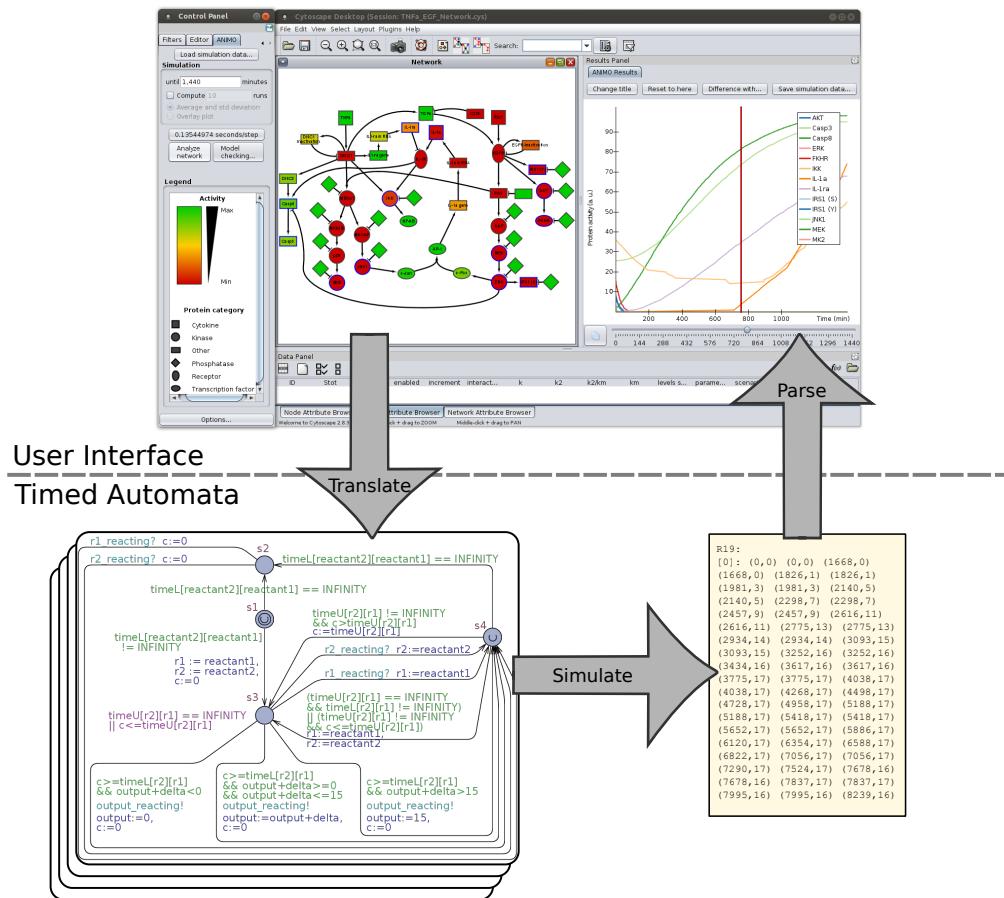


Figure S5: The passages intercurring between the press of the *Analyse network* button and the generation of an activity level graph. A simulation produces, for each selected node, a series of pairs (t, a) , with t a time instant along the simulation and a an activity level. These data are then parsed and translated into a graph.

Each TA model generated by ANIMO contains one automaton for each interaction (activation or inhibition) in the network. A TA representing an interaction performs a cyclic series of steps, continuously updating the target of the interaction it represents, and adapting the timing of the next update according to the user-defined dynamics. Synchronizations between different automata occur when the activity level of a network component (e.g. ERK) changes: this allows the automata depending on that component to update their time settings.

The abstract behaviour of the interaction $\text{MEK} \rightarrow \text{ERK}$ in the TA model used in ANIMO is described in Figure S6. There, the activity levels of MEK and ERK are represented by variables called, respectively, $\text{MEK}_{\text{activity}}$ and $\text{ERK}_{\text{activity}}$. A more detailed description of the TA model underlying ANIMO has been presented at the IEEE conference BIBE 2012 (Schivo et al., 2012).

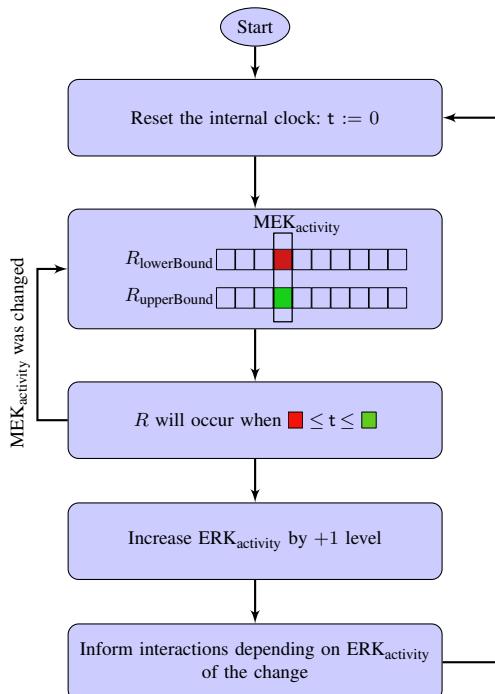


Figure S6: Schematic overview of the steps taken during a simulation run by a Timed Automaton modelling an interaction R that increases $\text{ERK}_{\text{activity}}$ and depends only on $\text{MEK}_{\text{activity}}$. In this example, MEK has 10 activity levels. After resetting the internal clock t , the automaton sets the time constraints for the interaction. $\text{MEK}_{\text{activity}}$ is used as the index inside the time tables $R_{\text{lowerBound}}$ and $R_{\text{upperBound}}$, which contain pre-computed lower- and upper-bounds for the interaction timing. Once the bounds have been identified, R can occur when t reaches a value inside the continuous time interval $[\textcolor{red}{■}, \textcolor{green}{■}]$. When it occurs, R increases the value of $\text{ERK}_{\text{activity}}$ by 1. All interactions that depend on $\text{ERK}_{\text{activity}}$ are notified of the change (via a synchronization on a specific channel), so that the associated time bounds are updated accordingly. After resetting the clock t , the process can restart. If $\text{MEK}_{\text{activity}}$ was changed by another automaton before the occurrence of R , the time bounds are updated according to the new activity level of MEK.

3.2 Granularity of an ANIMO network node

Figure S7 shows the differences between different choices for the number of levels of a node. This allows to adapt a model to the quality of experimental data.

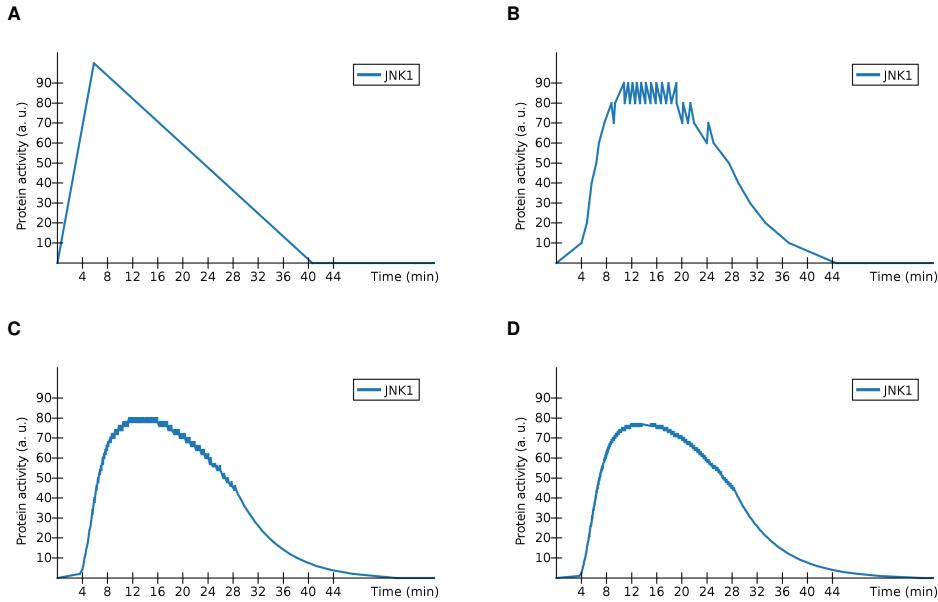


Figure S7: Comparing different reactant granularity settings. (A) 2 levels, (B) 10 levels, (C) 50 levels, (D) 100 levels. The JNK1 series is computed from the model presented in Figure 5B, considering 100 ng/ml TNF α as treatment condition over a period of 60 minutes.

4 ADDITIONAL NOTES

4.1 Simulating the day-night cycle

The model presented in Figure 4A contains a node labelled Day/Night. That node abstracts our representation of the cyclic alternation of day and night, which causes the variations in cryptochrome (*cry*): these oscillations allow the network to synchronize to a time zone. Note that the network oscillates also when the node *cry* is not included in the model.

The alternation between day and night is represented in our model with a repressilator-like (Elowitz, 2000) subnetwork, as can be seen in Figure S8. In the model by Fathallah-Shaykh et al. (2009) a specific function was introduced in the equations to approximate the experimental data from Kadener et al. (2007).

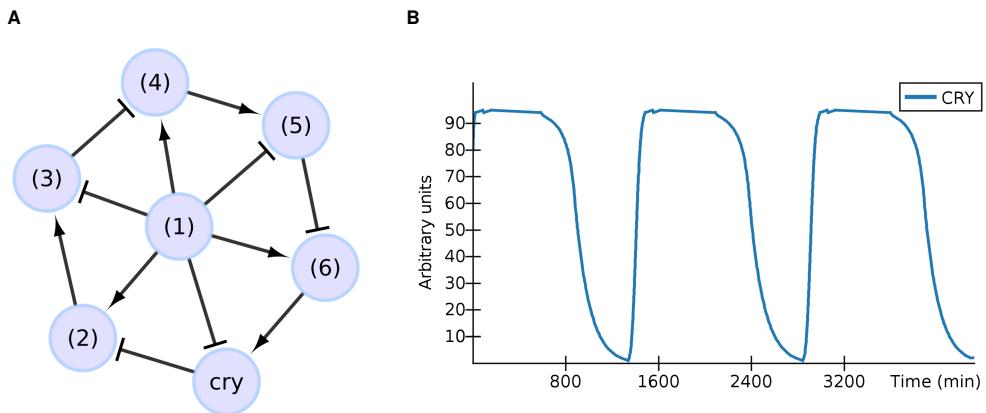


Figure S8: (A) The repressilator-like subnetwork used to represent the alternation between day and night that cause the oscillations in CRY concentrations in the network modelled in Section 3.1. (B) A graph plotting the oscillations in CRY along a period of three days.

4.2 Note on the parameters in the TNF α -EGF model

The parameters in the model in Figure 5B have been set by fitting the model to the experimental data for conditions with 100 ng/ml TNF α . In the model we have set the starting level of TNF α at 100 out of 100 for these conditions. This level is a dimensionless quantity that indicates the maximum activity level in the data set. We found that setting the initial level of TNF α at level 8 out of 100 gave slightly better results for the condition with 5 ng/ml TNF α than level 5 out of 100. We believe that this has to do with the fact that 100 ng/ml is a highly supra-physiological concentration of TNF α , that will rapidly cause activation of all receptors present. Fitting the model to this experimental condition may have resulted in slight deviations in the parameter values. Nevertheless, the modelling results illustrate that building a model with basic kinetic rate laws can give useful predictions over a range of concentrations. Figures 6C and 6D show the modelling results with TNF set at 8 out of 100.

5 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 S3 shows the comparison between ANIMO and the selected tools.

Table S3. 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 (Schivo et al., 2012)	Timed Automata	Yes	Yes	Yes	Yes	Yes
Bio-PEPA Workbench (Ciocchetta et al., 2009)	Bio-PEPA	No	No	No	No	Yes
Cell Illustrator (Nagasaki et al., 2011)	Petri Nets	Yes	Yes	No	Yes	No
COPASI (Mendes et al., 2009)	ODE, stochastic models	No	No	No	No	No
COSBI LAB ¹	BlenX	Yes	Yes	No	Yes	No
GINsim (Chaouiya et al., 2003)	Boolean Networks	No	Yes	Yes	Yes	Yes ²
GNA (de Jong et al., 2003)	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.

6 SUPPLEMENTARY FIGURES

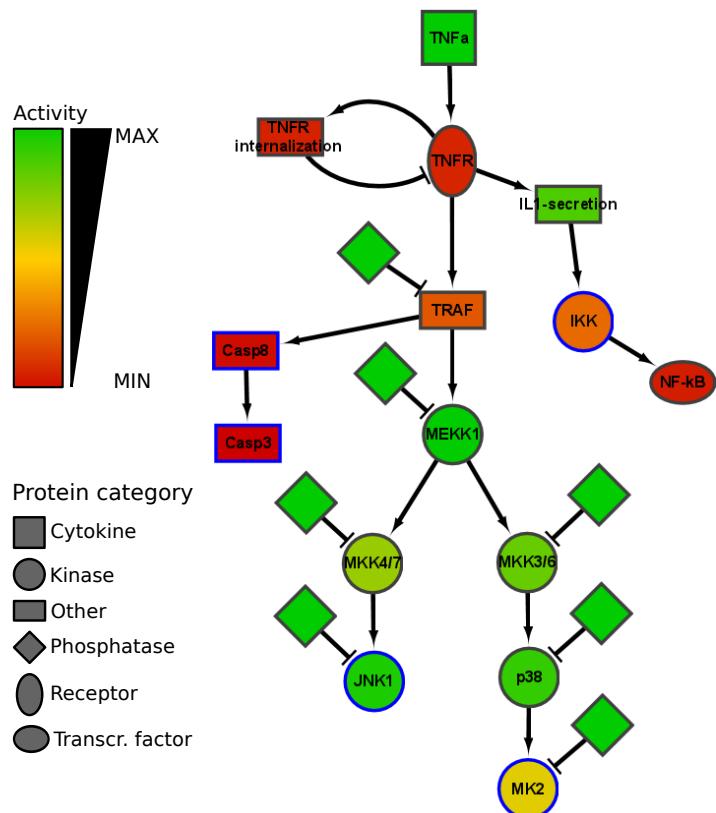


Figure S9: The model for the TNF α pathway in isolation. Node colours represent the activity level of the corresponding modelled reactants at time $t = 10$ minutes after a stimulation of 100 ng/ml TNF α .

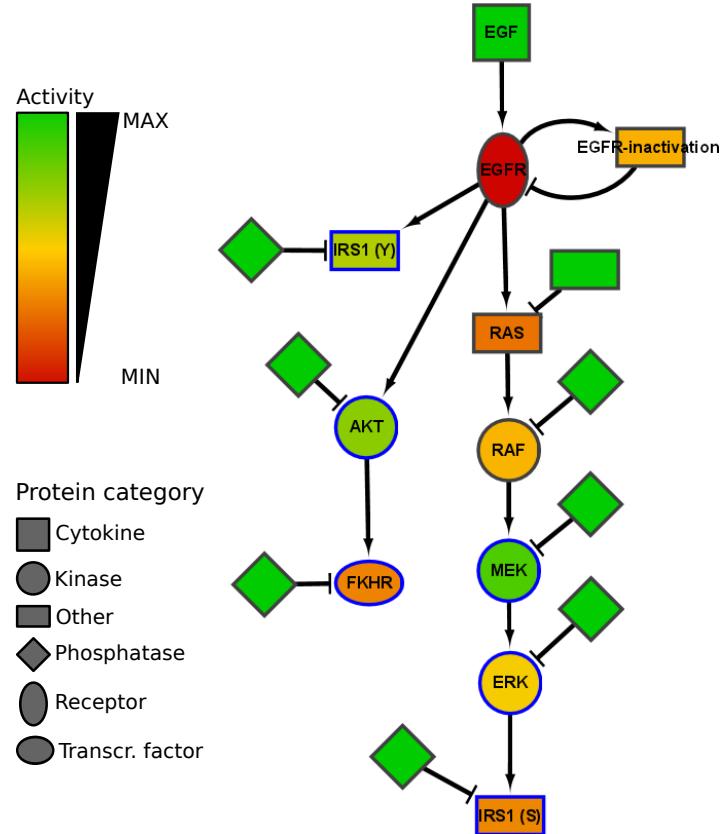


Figure S10: The model for the EGF pathway in isolation. Node colours represent the activity level of the corresponding modelled reactants at time $t = 5$ minutes after a stimulation of 100 ng/ml EGF.

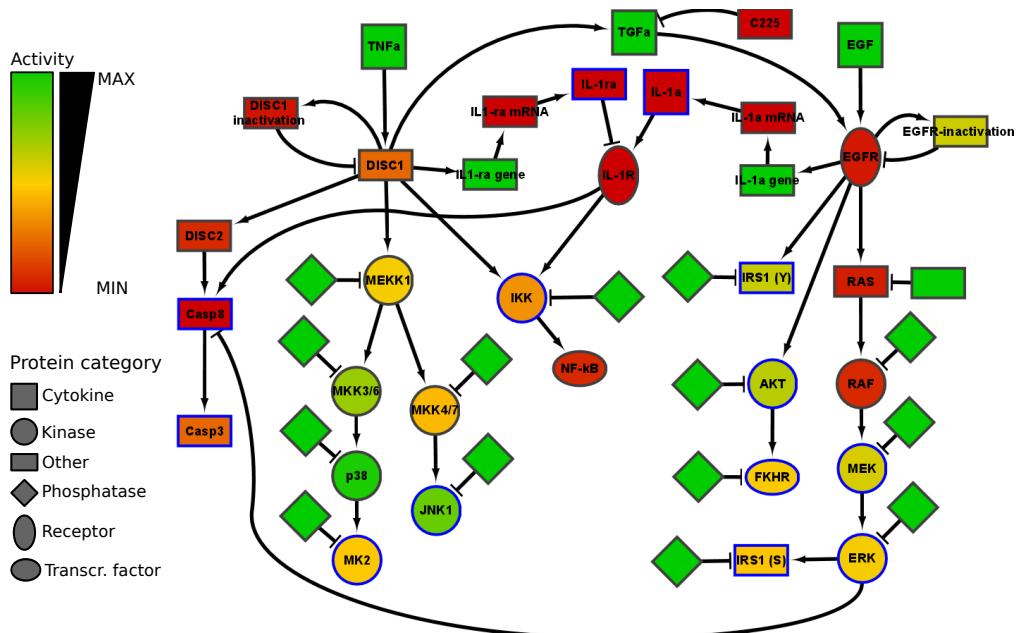


Figure S11: The model for the merged $\text{TNF}\alpha$ and EGF pathways before adding the hypotheses in Figure 5A. Node colours represent the activity level of the corresponding modelled reactants at time $t = 15$ minutes after a stimulation of 100 ng/ml $\text{TNF}\alpha + 100$ ng/ml EGF.

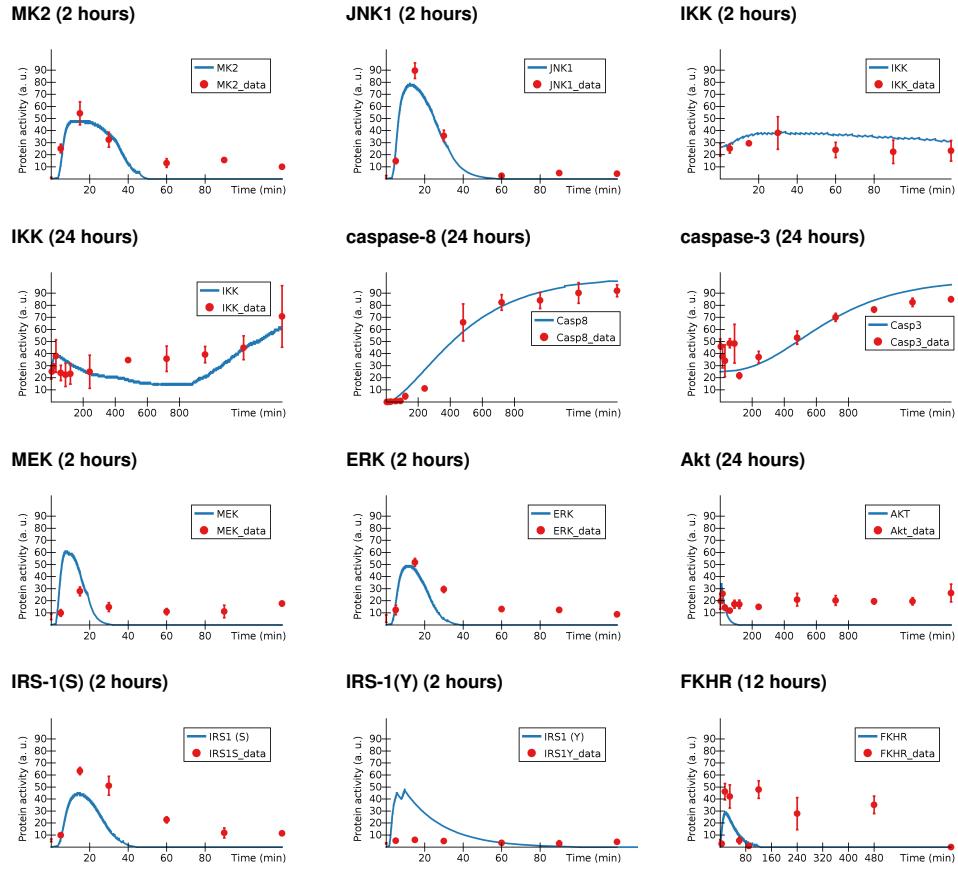


Figure S12: Comparison between the ANIMO model in Figure 5B and experimental data. Treatment condition: 100 ng/ml TNF α . In order to ease the comparison for earlier responses, the time span for those cases is less than 24 hours.

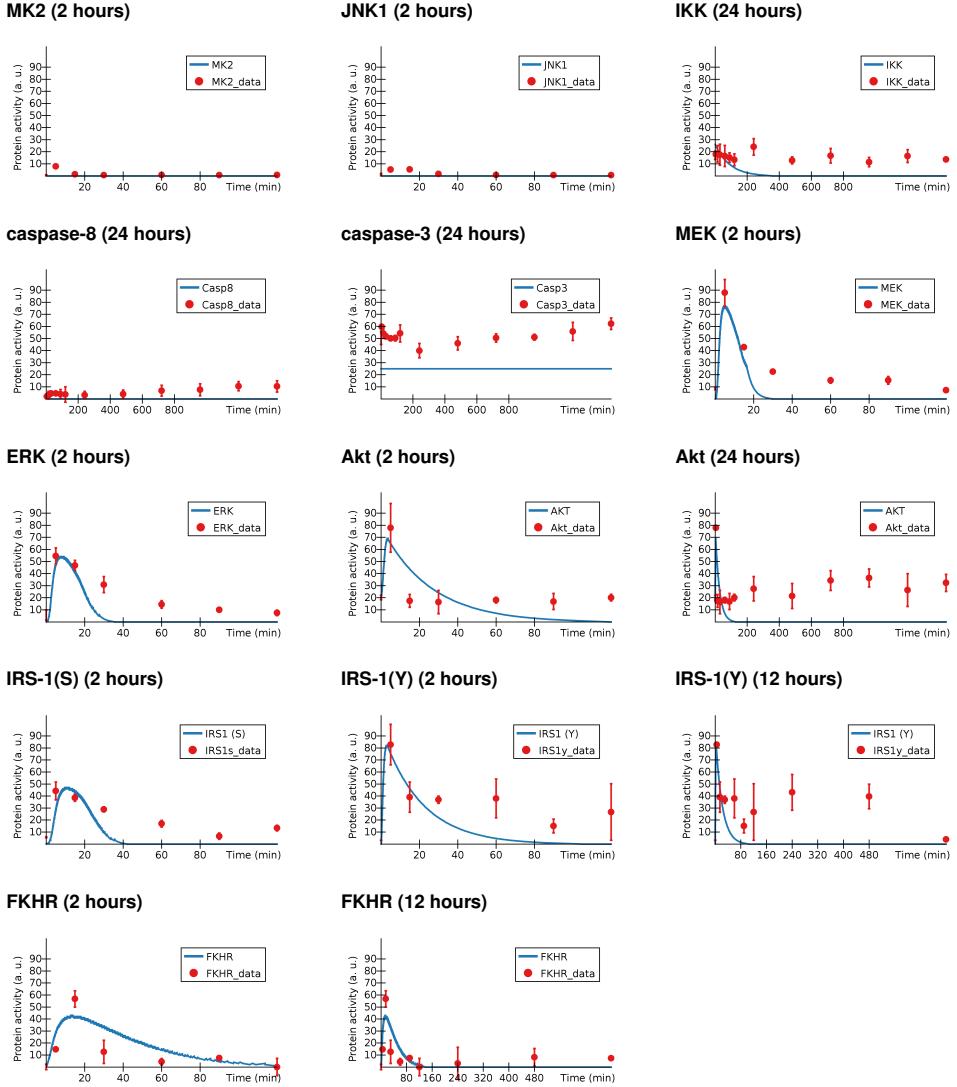


Figure S13: Comparison between the ANIMO model in Figure 5B and experimental data. Treatment condition: 100 ng/ml EGF. In order to ease the comparison for earlier responses, the time span for those cases is less than 24 hours.

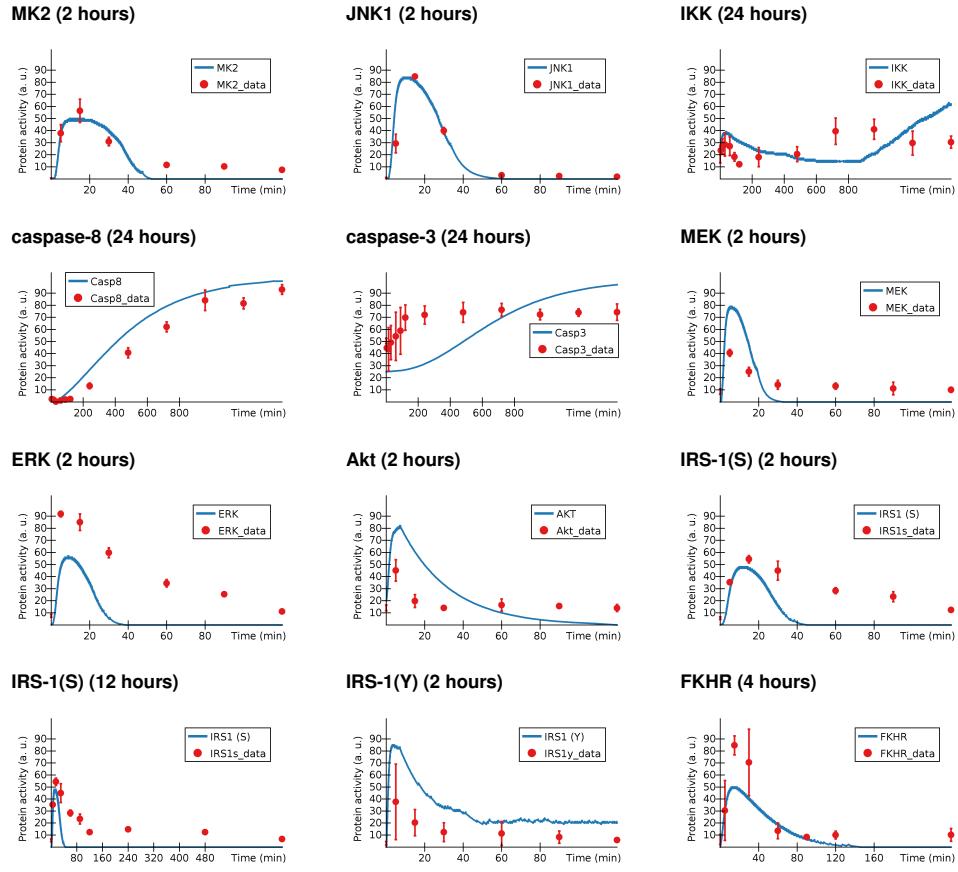


Figure S14: Comparison between the ANIMO model in Figure 5B and experimental data. Treatment condition: 100 ng/ml TNF α + 100 ng/ml EGF. In order to ease the comparison for earlier responses, the time span for those cases is less than 24 hours.