Reasoning about the ERK signal transduction pathway using BioSigNet-RR

Carron Shankland[a], Nam Tran[b], Chitta Baral[b], and Walter Kolch[c]

[a]Department of Computing Science and Mathematics, University of Stirling, UK,
[b]Department of Computer Science and Engineering, Arizona State University, USA,
[c]Beatson Institute for Cancer Research, and Institute of Biomedical and Life Sciences, University of Glasgow, UK.

Abstract. We present a simple signal transduction pathway from the literature, modelled using a knowledge-based approach in the tool BioSigNet-RR. This allows qualitative analysis of the pathway in three ways.
Keywords: signal transduction pathways, knowledge representation.

1 Introduction

Signal transduction, the correct receiving and processing of external and internal signals, is essential for the proper functioning of cells and ultimately the organism. Mammalian cells have more genes for signal transduction than they have to control metabolism or maintain the integrity of the genome [Vea01]. The importance of signal transduction is further documented by the fact that almost any disease can be described in terms of aberrations of signalling processes and that many important diseases are caused by a breakdown in communication between and within cells. The classical examples are cancer and autoimmune diseases, where either cells proliferate uncontrollably or attack other cells in the body, respectively. Thus, there is great interest in signal transduction research, and we have witnessed enormous progress being made in the last two decades, in particular regarding the identification of the components. Deciphering the function of these signal transduction pathways has turned out much harder, mainly because they are not organised as linear pathways as originally perceived, but in large networks where components are linked to many other components featuring extensive crosstalk and crossregulation. Further, many components are re-used in different functional contexts.

Specific biological responses to stimuli are generated in a combinatorial fashion by integrating both qualitative and quantitative aspects of signalling. These aspects are generated through biochemical reactions, often phosphorylation, that change the activity of signal transduction enzymes and ultimately specify a change in the biological behaviour of the cell. While biochemistry and genetics are very powerful in analysing single biochemical reactions and individual steps in signal transduction networks, they fail us in understanding the results of many reactions and steps and ultimately the behaviour of the network. Therefore, there is a need for mathematical modelling of signalling networks in order to improve our understanding of how specific network behaviour is generated in response to defined inputs.

The emerging area of systems biology yields many computational approaches to such pathways [SK04,VAB04]. Most models are quantitative, and analysis predicts quantities of various species, given initial conditions. Choosing parameters (such as reaction rates) is non-trivial. The experimental determination of such rates is tedious and usually not feasible for all reactions in a pathway model. As such measurements have to be carried out using purified enzymes in vitro, it is unclear whether they really reflect the activities in their physiological environment in intact cells. Additionally, in models with large numbers of variables, considering all interesting initial conditions (combinations of initial concentrations of proteins) is intractable.

An alternative approach is to ignore quantitative data, and concentrate on qualitative aspects of the problem. In this paper we present a preliminary report on describing a a simple network (the ERK pathway of [CSK$^+$03]) using the declarative language of the tool BioSigNet-RR [BCT$^+$04] (BSN for short), and carrying out qualitative analyses. This work is at an early stage; currently we are exploring the requirements of the problem area and different modelling approaches within BSN. BSN itself is very much under current development; exploratory work at this stage will influence future implementation developments. The paper is structured as follows. Section 2 introduces BSN and the ERK pathway. Section 3 explains the construction of the model in BSN. Although we use the ERK pathway for illustration here, the method is general and can be easily applied to other pathways. Section 4 describes the qualitative analyses that can be carried out. A comparison with related work is given in Section 5, and conclusions may be found in Section 6, together with a brief discussion of ongoing work and future directions.

2 Background

2.1 BioSigNet-RR

BioSigNet-RR [BCT$^+$04] is a system for representing and reasoning about signal networks. The name derives from 'biological signal networks' and 'representation and reasoning'. The input to BSN is a knowledge base and observations. A knowledge base encodes knowledge about a signal network, including logical statements based on symbols termed *fluents* and *actions*. Fluents represent the various properties of the cell while actions denote biological processes (e.g. biochemical reactions, protein interactions) or external interventions. The logical statements describe the impact of these actions on the fluents, how actions can be triggered or inhibited inside the cell. The logical statements are also called *rules* of the knowledge base. Let us consider a simple knowledge base as follows.

$$bind(ligand, receptor)\ \textbf{causes}\ bound(ligand, receptor) \tag{1}$$

$$high(ligand)\ \textbf{triggers}\ bind(ligand, receptor) \tag{2}$$

$$bound(another, receptor)\ \textbf{inhibits}\ bind(ligand, receptor) \tag{3}$$

Here, $high(ligand)$ is a fluent representing the that the concentration level of the ligand is high; $bound(ligand, receptor)$ represents that ligand is being bound to receptor and $bind(ligand, receptor)$ is an action denoting the association of ligand with receptor. (1) describes that the association of $ligand$ and $receptor$ results in $ligand$ being bound to $receptor$; (2) describes that the association occurs when the level of $ligand$ is high; and (3) describes that the association is blocked when $receptor$ is bound to another molecule.

Observations are statements about the truth value of properties or about action occurrences at time points. For example, the statement f **at** t is the observation that a formula f is true at time t; and the statement a **occurs_at** t' is the observation that action a occurs at time t'.

Given a knowledge base and observations, BSN's reasoning engine can perform various kinds of reasoning such as prediction, explanation, and planning. Let $a_1, \ldots, a_n$ be a sequence of actions and F be a logical formula. Asking if F will be true given the sequence $a_1, \ldots, a_n$ is referred to as *prediction*. Now if F is given, but the reasoning system has to find the appropriate $a_1, \ldots, a_n$ so that F is achieved then we have *planning*. Finally, if a set of observations is given and we need to find particular action occurrences and/or facts about intermediate states of the world that explain the observations, then we do *explanation*. All the above kinds of reasoning may have to be done with partial or incomplete information in the knowledge base. In BSN, the formula F is expressed using linear temporal logic (LTL) [Pnu71]. LTL allows us to express properties over paths, including cyclic properties, such as reaching a particular state infinitely often. Currently the BSN implementation does not support cyclic queries, but this is planned for future work.

In the context of signalling networks, *predict* is equivalent to saying: given some initial concentrations of proteins show that it is possible to achieve some particular output. *explain* allows us to state the desired outcome and discover the initial conditions giving rise to that outcome. Lastly, given some initial concentrations of proteins, and assuming there is some external intervention, *plan* tells us when the intervention should occur to get the desired outcome.

The knowledge representation language of BSN has a precise semantics, defining the models of a knowledge base and observations [TB04]. A model is a trajectory of the form $s_0 A_0 s_1 A_a \ldots s_n A_n$, where A_i's are sets of actions and s_i's are logical interpretations of fluents (i.e. *states* of the cell). A_i consists of actions occuring at state s_i, which can be either triggered by the cell (internally) or carried out by the environment (externally). The actions in A_i transform the cell from state s_i into state s_{i+1}. Thus a model describes a possible evolution of the cell in time. Note that time is abstract: we have a notion of ordering, but not of real time.

BSN is implemented using the declarative programming language AnsProlog [Bar03]. This stands for programming in logic using *answer sets*, or stable models. This approach is useful for non-monotonic reasoning, especially in the context of incomplete information. Note that AnsProlog is a declarative language different from Prolog. Prolog has many non-logical features, and is not purely declarative. This makes Prolog unsuitable for knowledge representation [1].

BSN can be used in three main modes. There is a graphical interface which makes initial description of the model easy and less error-prone; however, only simple queries may be expressed in this interface. The system can also be used from the command-line, taking as arguments a text file containing the knowledge base and one containing the query. This allows more complex queries, but is of course rather error-prone. Finally, the user can access the components of BSN (the tools lparse and smodels [NSS00]) to get more information about the underlying models constructed.

2.2 The features of pathways

The example we use in the rest of the paper is that of the Ras/Raf-1/MEK/ERK pathway presented in [CSK+03] (ERK pathway for short). The ERK pathway controls a number of fundamental cellular processes including cell survival, proliferation, motility and differentiation. Briefly, the pathway consists of a core of three kinases, Raf, MEK and ERK. Raf is activated by growth factor receptors on the cell surface via the small G-protein Ras. Raf phosphorylates and activates MEK, which in turn phosphorylates and activates ERK. The interaction betwenn Raf and MEK is negatively regulated by RKIP, which can bind to Raf and prevent it from interacting with MEK, thereby interfering with MEK phosphorylation by Raf [YSL+99]. ERK has more than 80 known substrates which it regulates by phosphorylation [PRB+01]. One substrate of ERK is RKIP. ERK inactivates RKIP by phosphorylation resulting in the dissociation of RKIP from Raf-1, permitting Raf to interact with MEK. When RKIP is dephosphorylated it can bind to Raf again. Thus, ERK mediated RKIP phosphorylation constitutes a positive feedback loop. A graphical representation of the pathway is presented in Figure 1. Phosphorylation is indicated by the suffix -P; e.g. ERK-P is monophosphorylated ERK, ERK-PP is double phosphorylated ERK.

Although such networks are many and varied, they can all be described by kinetic equations, allowing standard features to be exposed. In the Figure, reactants are represented as circles and reaction rates as rectangles. A reaction then is a collection of arrows between reactants and a reaction rate. For example, $k1$ is the association of Raf1 and RKIP to produce Raf1*/RKIP. Reactions have direction; $k2$ is the dissociation of Raf1* and RKIP, the reverse of reaction $k1$.

Our model is abstract and purely qualitative, therefore we represent neither the precise amount of any compound nor the numerical value of reaction rates. Instead we use a binary representation: either we have the compound (and assume that this is a sufficient amount to participate in a reaction), or we have none of the compound (alternatively, an insufficient amount to participate in a reaction). Taking a simple view of the model makes it much easier to describe and analyse, but yet it is still possible to make useful observations about the network, as will be demonstrated in Section 4.

[1] Please refer to [BCT+04,TB04] for more details.

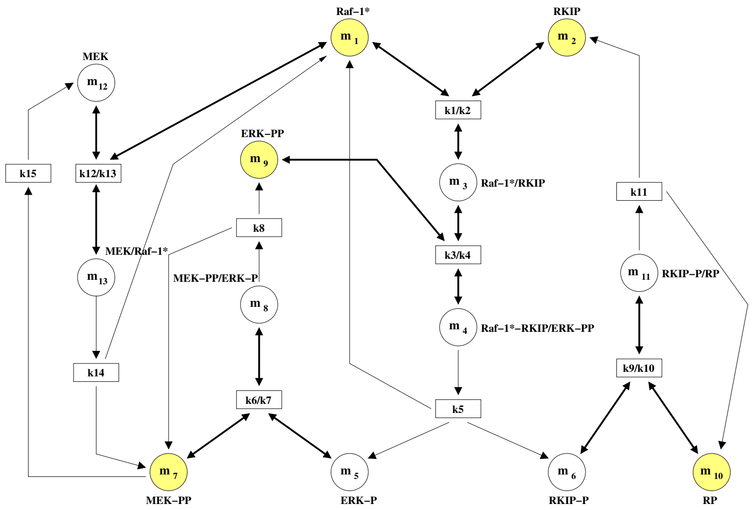

Fig. 1. RKIP inhibited ERK pathway

Before constructing any model, it is important to ask, what is the model for? In this case, what biological insights do we hope to gain by analysing our model? Such insights can then be backed up by wet lab experimentation. At a topological level, we want to be able to determine which parts of a network are important. For example, if we cut one part of the network out (perhaps using a chemical treatment), how does that affect the rest of the network? A related question might be that if there are two paths both achieving a particular outcome then how and when can we switch between those paths? Conversely, we might identify parts of the network which may not be omitted. There may also be patterns in the way in which the network is traversed, e.g. cycling between particular states. Suitable queries can be formulated for all these questions.

3 Encoding pathways in BioSigNet-RR

We present a model of the ERK signalling pathway in BSN to illustrate the language of the tool. Many modelling choices are made in constructing even this simple example. These are not the focus of this paper; we simply present one particular set of decisions. More detail may be found in Section 6 and in a follow-up paper [SK05]. The modelling technique is manual but also deliberately mechanistic; a future development may be to automatically generate BSN descriptions from existing network descriptions, e.g. in SBML.

3.1 Fluents

A fluent in the BioSigNet-RR system represents a reactant in the pathway. Fluents represent real world facts and can have the value *true* or *false*. We indicate presence of the protein by `high(raf1)` and absence by `-high(raf1)`, where - is logical negation. The choice of name is arbitrary; we could have used `is_present(raf1)` instead.

The Appendix A shows the full knowledge base input to BSN. The first 13 lines are the complete set of declarations of fluents for the ERK pathway. Note that this simply declares the names to be used, and the presence or absence of proteins for a particular query will be set up in its initial conditions.

3.2 Actions

An action in the BioSigNet-RR system represents a reaction, or possibly a combination of reactions, in the network. For each reaction, specify an action name. Here we're only giving names to actions; the causes and consquences of actions are expressed using rules in the following sections.

For example, for the ERK pathway of Figure 1, we declare:

```
<action> k1k2.
<action> k5.
```

and so on. For bidirectional reactions we model the combined reaction, hence `k1k2` above. Recall that this name has no numerical meaning in our model.

3.3 Dynamic Rules

Given the ingredients of our system, the reactants/proteins and the reactions, we describe how each reaction affects the reactants. The dynamic rules describe the results of each action. For example, for the ERK pathway of Figure 1, we declare reaction `k1k2` to have the following effects:

```
k1k2 <causes> high(raf1_rkip).
k1k2 <causes> -high(raf1).
k1k2 <causes> -high(rkip).
```

That is, the reaction `k1k2` produces the new protein composed of `raf1` and `rkip`, but in the process uses up the available `raf1` and `rkip`. These values remain until further actions change them, e.g. `k3k4` depletes Raf1*/RKIP to produce Raf1*/RKIP/ERK-PP.

3.4 Trigger Rules

The dynamic rules only describe the results of an action; we must also describe the causes of actions. For example, for the ERK pathway of Figure 1, we declare the trigger for reaction `k1k2` as:

```
high(raf1) ; high(rkip) ; -high(raf1_rkip) <triggers> k1k2.
```

The semicolon can be read as logical and.

3.5 Inhibiting Rules

Lastly, there may be situations in which the action cannot occur, even if the triggering rules described above are satisfied, i.e. we explicitly over-ride the normal triggering mechanisms. This facilitates non-monontonic reasoning and elaboration tolerance. For example, for the ERK pathway of Figure 1, we declare the inhibitions for reaction `k1k2` as:

```
high(raf1_rkip) <inhibits> k1k2.
```

That is, if the result compound is already present the action will be blocked.

Inhibition becomes important in situations where one component can participate in several reactions or pathways. For instance, ERK can translocate from the cytosol into the cell nucleus and regulate transcription factors. Preventing ERK from entering the nucleus precludes it from activating transcription factors, but still allows it to phosphorylate cytosolic substrates. In contrast, inhibition of ERK by MKP phosphatases terminates all ERK signalling, as MKPs remove the activating phosphates from ERK causing it to switch back into the inactive conformation.

4 Reasoning about pathways

For each kind of analysis, queries must be formed. These are expressed using linear temporal logic. Queries can use the path quantifiers **always** and **eventually**, standard logical operators (and, or, not) and fluent literals.

4.1 Predict

Essentially, this form of analysis can be expressed as

Given some initial conditions, prove formula F holds.

In other words, this is rather similar to standard model checking.

In the context of signalling networks, this is equivalent to saying: given some initial concentrations of reactants show that it is possible to achieve some particular output. For the example above, a predict query might be:

```
initial(0).
observed(high(raf1),0).
observed(high(rkip),0).
observed(high(mek_pp),0).
observed(high(erk_pp),0).
observed(high(rp),0).
the_prediction(evnt(high(erk_p)),0).
evnt(high(erk_p)).
:- predict(true), not holds(evnt(high(erk_p)),T), match(0,T), time(T).
:- predict(false), holds(evnt(high(erk_p)),T), match(0,T), time(T).
prediction.
```

(This is the text file representation of the query. This particular query could have been carried out in the GUI.)

This example says, given that we have high concentrations of `raf1`, `rkip`, `mek_pp`, `erk_pp` and `rp` at time 0, show that we can get high `erk_p` eventually (the operator `evnt`) (node m5 in Figure 1). For this to be true we must find a positive model (line 9), and no negative models (line 10). This query is an abstract version of the claim of Cho *et al* [CSK+03].

Other properties are proved in a similar fashion. For example, given high `raf1`, `rkip`, `mek`, `erk_pp` and `rp`, then we can show

```
not(evnt(wedge(high(mek_raf1), high(raf1_rkip)))).
```

where `wedge` represents logical and. This is a sort of mutual exclusion formula and can only be true if `k1k2` and `k12k13` can proceed together.

LTL allows us to express cyclic properties, such as reaching a particular state infinitely often. Currently the BSN implementation does not support such queries, but this is planned for future work.

4.2 Explain

This can be regarded as the reverse of Predict:

> *Assuming formula F holds at some point in the future, deduce possible initial conditions.*

In the context of signalling networks, this allows us to state the desired outcome and discover the initial conditions which might give rise to that outcome. In an analog of the example above, we can determine initial conditions resulting in high `erk_p`:

```
initial(0).
unknown(high(raf1)).
unknown(high(rkip)).
unknown(high(rkip_p)).
unknown(high(mek_pp)).
unknown(high(erk_pp)).
unknown(high(rp)).
unknown(high(mek)).
to_be_explained(evnt(high(erk_p)),0).
evnt(high(erk_p)).
:- not holds(evnt(high(erk_p)),T), match(0,T), time(T).
explanation.
```

Here we declare the unknown reactants (i.e. their initial concentrations are unknown), and then ask for our formula to be explained. All reactants not mentioned default to low concentration.

This yields one model satisfying the formula. More usefully, we can ask the system to tell us *all* possible models satisfying this formula. This requires recourse to the command line. For this example, 16 are generated. Manual inspection of the output allows us to categorise the results into two groups: either we must have high `raf1`, `rkip` and `erk_pp` initially, or we must have high `raf1`, `rkip_p`, `rp` and `erk_pp`. What varies in the models is the concentrations of other reactants. This is easily verified by examination of the network of Figure 1. A possible future direction for the work may be to carry out some post-processing of the models generated to make some quantitative judgements about the relative importance of particular reactants or paths in the system.

4.3 Plan

This last form of analysis is more concerned with the actions carried out, and can be expressed as

> *Given some initial conditions, and some intervention action, determine at what point the intervention must occur to ensure formula F holds at some point in the future.*

Here we have partial information about the state, not enough to show that F holds, therefore some intervention must be made which changes the state.

In the context of signalling networks, this is equivalent to saying, given some initial concentrations of reactants, and assuming we will add some new element to the system, for example a drug treatment, when should it be added to get the desired outcome. Continuing the example above, we might want to find out at what point we can add `erk_pp` to the system to still produce `erk_p`.

```
intervention(intro(erk_pp)).
initial(0).
observed(high(raf1),0).
observed(high(mek),0).
observed(high(rkip),0).
the_goal(evnt(high(erk_p)),0).
evnt(high(erk)).
:- not holds(evnt(high(erk)),T), match(0,T), time(T).
planning.
```

Here we have to add a new action `intro(erk_pp)`. This is declared to have the single effect of making `erk_pp` high. The plan produced by BSN states that this action must occur at time 2.

5 Related Work

The abstract approach to modelling signalling pathways was presented using the Performance Evaluation Process Algebra by Calder, Hillston and Gilmore [CHG04]. We have adopted their approach of using high and low to abstract away from concentrations; however, we have taken the abstraction further by also disregarding reaction rate information. Calder *et al* were able to reproduce the result of Cho *et a*l [CSK+03] that RKIP damps down the production of ERK. In addition, properties expressed in CSL (allowing probabilistically quantified logic formulae to be expressed) were verified.

The logic CTL (Computation Tree Logic) has been used to reason about biological pathways in [CRCD+04]. Chabrier-Rivier *et al* develop a specialised language to describe pathways, and then translate that to a Kripke structure so standard model checking tools can be utilised. The approach of [CRCD+04] can carry out analyses in the style of *predict* here (using CTL), but it is not possible to compute *explain* or *plan* queries.

The paper lists interesting properties expressible in CTL that may be of biological significance. CTL is of the same family as LTL, but is incomparable. That is, it is possible to express some properties in CTL which cannot be expressed in LTL, and vice versa. For example, cyclic properties referring to the existence of stable states may be expressed in CTL, but not in LTL. LTL will allow statements to be made about stable states for all paths. Conversely, it is possible to express that something occurs infinitely often in LTL, but not in CTL. Both are qualitative, therefore it is not possible to express properties concerning particular periods of time, or amounts of reactants.

6 Conclusions and Future Directions

We have presented a tool in which abstract, qualitative models of cell signalling networks can be described. Several kinds of analysis can be carried out, allowing the important aspects of the system to be seen clearly. In our simple ERK pathway these analyses could have been carried out by hand since the model is small; however, our approach will scale up to larger, more complex models in a straightforward manner. An advantage of the approach is its simplicity; there is no need to use sophisticated experimental techniques to estimate parameters of the model, since there is no quantitative data. This approach can therefore be seen as complementary to established differential equation approaches, perhaps as a useful first step in analysis, focussing attention on the areas of importance. A particularly useful application could be the mapping of new connections in signalling networks. For instance, it is a well know puzzle that in some cellular systems Ras seems to be upstream of Rac (a Ras related G-protein) while in others Ras appears downstream of Rac. The logical analysis if the signal transduction network topology will quickly reveal which connections are permitted and under which conditions. Another useful feature is to predict how many different types of network behaviour are possible and under which conditions these types of behaviour are realized. This will already allow a critical evaluation of the model and will be useful in guiding experimental work.

Future work includes exploration of more complex models as mentioned above, such as the Schoeberl *et al* model [SEJGM02], and further development of the BSN system both in the modelling language and in the interface. New directions in answer set programming, for example the use of preferences to compute the relative desirability of models, will also be explored.

In this paper we did not discuss modelling decisions, but clearly these are crucial, and tightly linked to the sorts of analysis to be performed. For example, representing the biology of inhibition poses a challenge. RKIP inhibits the interaction of Raf with MEK and thereby prevents MEK phosphorylation. Thus, I think inhibition is necessary to keep. So are negative triggers. For instance, phosphorylation of RKIP by Protein Kinase C (PKC) also will cause RKIP to dissociate from Raf, thereby releasing Raf inhibition. However, the PKC phosphorylated RKIP now can bind to G-protein coupled Receptor Kinase-2 (GRK-2) and inhibit it. Thus, the phosphorylation of RKIP by PKC inactivates RKIP as an inhibitor for Raf, but activates RKIP as an inhibitor for GRK-2 [LLQ03].

Two related questions are about resource and competition for that resource. Here, we assume reactions use up all available reactants. A reasonable alternative is assume there is always some reactant left. This produces similar results for the ERK system for simple queries; however, this means cycles in behaviour will not be detectable since once a reactant is present it is never removed. If resource is finite, then competition becomes important. Consider the system with initial `mek`, `raf1` and `rkip`. If all `mek` is used up in a reaction then clearly either `k1k2` or `k12k13` can proceed, but not both. This can be represented easily in BSN **??**. MEK and RKIP actually can compete for binding to Raf. In a situation where all three proteins are expressed the relative abundance of the proteins will determine the ratio of Raf - MEK and Raf - RKIP complexes. As Raf and MEK are expressed in all cell and RKIP expression is variable, RKIP expression will be the main determinator in most cells.

Acknowledgement

C Shankland is grateful for the support of the BioInformatics Research Centre, University of Glasgow, for hosting her sabbatical, and especially to Rainer Breitling and Richard Orton for discussions about signalling pathways, and to M Calder for providing the graphic for Figure 1.

References

[Bar03] C. Baral. *Knowledge representation, reasoning and declarative problem solving*. Cambridge University Press, 2003.

[BCT+04] C. Baral, K. Chancellor, N. Tran, N.L. Tran, A. Joy, and M. Berens. A knowledge based approach for representing and reasoning about signaling networks. *Bioinformatics*, 20:i15–i22, 2004.

[CHG04] M. Calder, J. Hillston, and S. Gilmore. Modelling the influence of RKIP on the ERK signalling pathway using the stochastic process algebra PEPA. In *Proceedings of Bio-CONCUR 2004*, Electronic Notes in Theoretical Computer Science, 2004.

[CRCD+04] N. Chabrier-Rivier, M. Chiaverini, V. Danos, F. Fages, and V. Schächter. Modeling and querying biomolecular interaction networks. *Theoretical Computer Science*, 325:25–44, 2004.

[CSK+03] K.-H. Cho, S.Y. Shin, H.-W. Kim, O. Wolkenhauer, B. McFerran, and W. Kolch. Mathematical modeling of the influence of rkip on the erk signaling pathway. In C. Priami, editor, *Proceedings of Computational Methods in Systems Biology (CMSB'03)*, volume 2602 of *Lecture Notes in Computer Science*, pages 127–141. Springer-Verlag, 2003.

[LLQ03] K. Lorenz, M.J. Lohse, and U. Quitterer. Protein kinase c switches the raf kinase inhibitor from raf-1 to grk-2. *Nature*, 426:574–579, 2003.

[NSS00] I. Niemelä, P. Simons, and T. Syrjänen. Smodels: a system for answer set programming. In *Proceedings of the 8th International Workshop on Non-Monotonic Reasoning*, 2000.

[Pnu71] A. Pnuelli. The temporal logic of programs. In *Proceedings of the 18th Annual Symposiom on Foundations of Computer Science*, pages 46–57. IEEE, 1971.

[PRB+01] G. Pearson, F. Robinson, G.T. Beers, B.E. Xu, M. Karandikar, K. Berman, and M. H. Cobb. Mitogen-activated protein (map) kinase pathways: regulation and physiological functions. *Endocr.Rev.*, 22:153–183, 2001.

[SEJGM02] B. Schoeberl, C. Eichler-Jonsson, E.D. Gilles, and G. Müller. Computational modeling of the dynamics of the MAP kinase cascade activated by surface and internalised EGF receptors. *nature biotechnology*, 20:370–375, 2002.

[SK04] H.M. Sauro and B.N. Kholodenko. Quantitative analysis of signaling networks. *Prog.Biophys.Mol.Biol*, 86:5–43, 2004.

[SK05] Baral Shankland, Tran and Kolch. Qualitative reasoning about the erk signal transduction pathway. Submitted for publication, 2005.

[TB04] N. Tran and C. Baral. Reasoning about triggered actions in AnsProlog and its application to molecular interactions in cells. In *Proceedings of KR 2004*, pages 554–563, 2004.

[VAB04] S.J. Vayttaden, S.M. Ajay, and U.S. Bhalla. A spectrum of models of signaling pathways. *Chembiochem*, 5:1365–1374, 2004.

[Vea01] J.C. Venter and et al. The sequence of the human genome. *Science*, 291:1304–51, Feb 16 2001.

[YSL+99] K. Yeung, T. Seitz, S. Li, P. Janosch, B. McFerran, C. Kaiser, F. Fee, K.D. Katsanakis, D.W. Rose, H. Mischak, J.M. Sedivy, and W. Kolch. Suppression of raf-1 kinase activity and map kinase signalling by rkip. *Nature*, 401:173–177, 1999.

A Complete ERK model as BSN input

```
%erk-forward
% No competition, finite resource, inhibition, -ve triggers,
% single direction reactions
<fluent> high(raf1).
<fluent> high(rkip).
<fluent> high(raf1_rkip).
<fluent> high(raf1_rkip_erk_pp).
<fluent> high(erk_p).
<fluent> high(rkip_p).
<fluent> high(mek_pp).
<fluent> high(mek_pp_erk_p).
```

```
<fluent> high(erk_pp).
<fluent> high(rp).
<fluent> high(rkip_p_rp).
<fluent> high(mek).
<fluent> high(mek_raf1).
<action> k1k2.
<action> k3k4.
<action> k5.
<action> k6k7.
<action> k8.
<action> k9k10.
<action> k11.
<action> k12k13.
<action> k14.
<action> k15.
k1k2 <causes> high(raf1_rkip).
k1k2 <causes> -high(raf1).
k1k2 <causes> -high(rkip).
k3k4 <causes> high(raf1_rkip_erk_pp).
k3k4 <causes> -high(raf1_rkip).
k3k4 <causes> -high(erk_pp).
k5 <causes> -high(raf1_rkip_erk_pp).
k5 <causes> high(raf1).
k5 <causes> high(erk_p).
k5 <causes> high(rkip_p).
k6k7 <causes> high(mek_pp_erk_p).
k6k7 <causes> -high(mek_pp).
k6k7 <causes> -high(erk_p).
k8 <causes> high(erk_pp).
k8 <causes> high(mek_pp).
k8 <causes> -high(mek_pp_erk_p).
k9k10 <causes> high(rkip_p_rp).
k9k10 <causes> -high(rkip_p).
k9k10 <causes> -high(rp).
k11 <causes> high(rkip).
k11 <causes> high(rp).
k11 <causes> -high(rkip_p_rp).
k12k13 <causes> high(mek_raf1).
k12k13 <causes> -high(mek).
k12k13 <causes> -high(raf1).
k14 <causes> high(mek_pp).
k14 <causes> high(raf1).
k14 <causes> -high(mek_raf1).
k15 <causes> high(mek).
k15 <causes> -high(mek_pp).
high(raf1) ; high(rkip) ; -high(raf1_rkip) <triggers> k1k2.
high(raf1_rkip) ; high(erk_pp) ; -high(raf1_rkip_erk_pp) <triggers> k3k4.
high(raf1_rkip_erk_pp); -high(rkip_p); -high(raf1); -high(erk_p) <triggers> k5.
high(mek_pp) ; high(erk_p) ; -high(mek_pp_erk_p) <triggers> k6k7.
high(mek_pp_erk_p) ; -high(erk_pp); -high(mek_pp) <triggers> k8.
```

```
high(rkip_p) ; high(rp) ; -high(rkip_p_rp) <triggers> k9k10.
high(rkip_p_rp) ; -high(rkip); -high(rp) <triggers> k11.
high(mek) ; high(raf1) ; -high(mek_raf1) <triggers> k12k13.
high(mek_raf1) ; -high(raf1) ; -high(mek_pp) <triggers> k14.
high(mek_pp) ; -high(mek) <triggers> k15.
high(raf1_rkip) <inhibits> k1k2.
high(raf1_rkip_erk_pp) <inhibits> k3k4.
high(erk_p) <inhibits> k5.
high(rkip_p) <inhibits> k5.
high(raf1) <inhibits> k5.
high(mek_pp_erk_p) <inhibits> k6k7.
high(erk_pp) <inhibits> k8.
high(mek_pp) <inhibits> k8.
high(rkip_p_rp) <inhibits> k9k10.
high(rkip) <inhibits> k11.
high(rp) <inhibits> k11.
high(mek_raf1) <inhibits> k12k13.
high(mek_pp) <inhibits> k14.
high(raf1) <inhibits> k14.
high(mek) <inhibits> k15.
```