BIOINFORMATICS

Vol. 00 no. 00 2004
Pages 1–8

A knowledge based approach for representing and reasoning about signaling networks

C. Baral [1], K. Chancellor [1], N. Tran [1], N.L. Tran [2], A. Joy [2] and M. Berens [2]

[1]Department of Computer Science and Engineering, Ira A. Fulton School of Engineering, Arizona State University, Tempe, AZ 85281, USA
[2]Translational Genomics Research Institute, 400 N. Fifth Street, Suite 1600, Phoenix, AZ 85004, USA

ABSTRACT

Motivation: In this paper we propose to use recent developments in knowledge representation languages and reasoning methodologies for representing and reasoning about signaling networks. Our approach is different from most other qualitative systems biology approaches in that it is based on reasoning (or inferencing) rather than simulation. Some of the advantages of our approach are: we can use recent advances in reasoning with incomplete and partial information to deal with gaps in signal network knowledge; and can perform various kinds of reasoning such as planning, hypothetical reasoning and explaining observations.

Results: Using our approach we have developed the system BioSigNet-RR for representation and reasoning about signaling networks. We use a NFκB related signaling pathway to illustrate the kinds of reasoning and representation that our system can currently do.

Availability: The system is available on the Web at http://www.public.asu.edu/~cbaral/biosignet.

Contact: baral@asu.edu

1 INTRODUCTION

Our goal in this paper is to make progress towards developing a system (and the necessary representation language and reasoning algorithms) that can be used to represent signal networks and pathways associated with cells and reason with them. We are interested in qualitative representation and reasoning, which can emulate the reasoning done by biologists (or biochemists) when they analyze and navigate a signaling pathway, but at a much larger scale and with respect to larger networks. This is in contrast to the approaches where signalling and transformation between various compounds are expressed using differential equations (O.Voit, 2000; Mishra, 2002; Antoniotti et al., 2003; de Jong, 2002).

In recent years several qualitative approaches have been proposed in systems biology (Priami, 2003). Most of these approaches are concerned with modelling and analysis of the model is done via simulation (Peleg et al., 2002; Regev et al., 2001) and perturbation. While certain questions about cell behavior can be easily answered using such an approach (such as the impact of a particular event), it is computationally expensive to answer questions about explaining a particular observation, or planning to alter the cell behavior in a particular way using simulation. In both these cases a simulation based approach would entail doing a large number of simulations to find the right explanation or the right plan.

Our approach in this paper is a knowledge based approach. We consider the cellular signal network as a knowledge base which can be asked many different kinds of queries. For the various kinds of queries the knowledge base is augmented with various reasoning mechanisms that allow the answering of the queries. This approach is more general than answering standard database queries with respect to a signal network database (Karp et al., 2000; Ogata et al., 1999), as the later is limited to answering queries that can be expressed using a database query language. For example, no existing database query language can express a query whose answer is a plan or an explanation.

An important dimension of our approach is that it allows for reasoning mechanisms that gracefully handle incomplete or partial information. This is extremely important as existing signal networks and pathways often have missing or suspected interaction links, or proven interactions whose outputs are uncertain, for example the yeast 2-hybrid interactions mentioned by Sambrano (2003). Besides being able to handle such incomplete information, our approach also allows for easy updating (referred to as 'elaboration tolerance') of the knowledge base when new knowledge becomes available. This avoids significant overhauling of the old model or scrapping of the old model and making a new model from scratch. Note that simple frame based approaches (Karp et al., 2000), which are a sub-class of classical logic, are monotonic and hence not elaboration tolerant.

The above mentioned issues of reasoning with incomplete information, elaboration tolerant representation, being able to predict, explain and plan are some of the main topics in the sub-area of 'Knowledge Representation' (KR) in the area of 'Artificial Intelligence' (AI).

The rest of the paper is organized as follows. In Section 2, we discuss several related works and point out their differences from our approach. Then we briefly describe the representation and reasoning system we have developed. It is followed by examples of reasoning with NFκB related signaling pathways. Finally, we discuss the scope of this approach and future works. Due to lack of space we do not describe our system implementation in detail here. We present some of the details in the companion paper (Tran & Baral, 2004).

2 RELATED WORKS

Petri nets: Originally developed for modelling concurrent systems, Petri nets and their extensions have been recently used in modelling biological processes (for example see Reddy et al., 1996; Peleg et al., 2002). Petri nets are intuitively appealing as a graphical modelling approach and have a solid mathematical foundation. As mentioned in (Peleg et al., 2002), several properties of Petri nets are relevant to biological systems. These include: liveness, boundedness, soundness and reachability. Despite the basic Petri nets being extended to allow time, hierarchies and stochasticness, due to its origin Petri nets are more of a modelling formalism than a representation and reasoning formalism. This leads to shortcomings such as: (i) It is not clear how to represent and reason with incomplete knowledge using Petri nets. (ii) It is not clear how to do explanation and diagnosis of unexpected observations using a Petri net model. (iii) Although reachability corresponds to plan existence, and finding reachability may be used for finding simple plans, it is not clear how to construct more general plans (for example, ones involving conditional statements and sensing) using a Petri net model.

π-calculus: The π-calculus is a formal language for specifying concurrent computational systems. In the biological π-calculus model (Regev et al., 2001), molecules and their individual domains are viewed as computational processes, whereas the residues of domains correspond to communication channels. Molecular interaction and modification are modelled as communication and channel transmission. The π-calculus modelling approach supports computer execution and analysis and formal verification. Similar to Petri nets, this approach is more suitable for modelling and simulation than elaboration tolerant representation and reasoning. In particular, it is not clear how to deal with incompleteness, and how to formulate explanation, diagnosis, and planning.

Pathway logic: Pathway logic (Eker et al., 2002; Talcott et al., 2004) is an algebraic formalism for modelling and analysis of signaling pathways at an abstract level higher than simulation. Biological structures are represented through algebraic expressions. A biological process corresponds to a set of rewriting rules transforming algebraic expressions from one to another. The possible analyses include reachability, prediction, and explanation. It is not clear how this approach deals with biological exceptions; how it can represent static causal relationships between biological properties, and how it supports planning or diagnosis.

Model checking: Model checking is used to compute temporal logic queries about properties of a concurrent system. Chabrier & Fages (2003) proposed a framework for querying and validating formal models of systems biology, which is based on: modelling biological processes as concurrent transition systems; using temporal logic such as CTL as a query language; and applying model checking techniques to evaluate CTL queries. This approach addresses both quantitative and qualitative aspects of biological systems. Besides reachability analysis, it supports queries about stable states, durations and concentrations of proteins. However, it is not clear how qualitative reasoning such as explanation, planning and diagnosis is supported in this framework. Also, it is not clear if this approach is able to deal with incomplete information.

3 OVERVIEW OF THE BIOSIGNET-RR SYSTEM

We have an initial implementation of a system for representing and reasoning about signal networks. We refer to our system as BioSigNet-RR, where BioSigNet means 'biological signal networks' and RR denotes 'representation and reasoning'.

The input to our system is a knowledge base representing the knowledge about a signal network. It includes information such as what are the various properties (termed **fluents**) of a network, the various **actions** that can be performed or triggered, the impact of these actions on the fluents, and when an action is triggered. Using this knowledge our reasoning system can perform various kinds of reasoning such as prediction, explanation, and planning. The following paragraph illustrates the above with respect to an example.

In cell, when a ligand binds to an appropriate receptor it activates the receptor. The activated receptor then triggers a series of events. The binding of a ligand to the receptor can be thought of as an *action*. The conditions under which a ligand can bind to a receptor is similar to the *executability conditions* of an action in a planning domain. In the context of a cell, by representing the cell behavior using actions, determining the effects and side effects of a drug (that manifests as a ligand) would correspond to *prediction*. Similarly, figuring out what interventions will change the cell behavior in a particular way corresponds to *planning*; and determining the causes behind some unexpected observations about the cell corresponds to *explanation* and *diagnosis*.

3.1 The fundamentals behind BioSigNet-RR

Before we continue on and describe our system, we give a brief overview of the fundamentals behind our system and its implementation. *First*, we need to precisely define what 'prediction' means? When is a series of 'intervention' (or action) a plan that achieves a desired goal? What is an explanation of an observation? We do this in the next subsection. *Next*, we need to implement the reasoning corresponding to prediction, planning and explanation. We do it using a declarative programming language called AnsProlog, which is not only useful for reasoning about prediction, planning and explanation, but is a good language for reasoning with incomplete information.

(Note that AnsProlog is a declarative language different from Prolog. Prolog is a programming language with roots in logic. But it has many non-logical features, and it is not declarative. This makes it unsuitable for knowledge representation.)

3.1.1 Short overview of reasoning about actions Reasoning about actions (Gelfond & Lifschitz, 1993; Rieter, 2001) is concerned with representing and reasoning about dynamic worlds. Research issues in this field involve:

- developing languages that allow for (a) succinct representation of the world, (b) succinct representation of causal relations between properties of the world, (c) succinct representation of the impact of actions (and their executability) on the world, (d) succinct representation of the way the world behaves, or the way we want the world to behave, and (e) representation of action-plans; and
- reasoning algorithms that can answer a query expressed using (d) and (e) with respect to a knowledge base expressed using (a), (b) and (c)

Different kinds of queries mentioned above lead to different kinds of reasoning. For example, given a sequence of actions $a_1, \ldots, a_n$ and a property p, asking if p will be true during (or after) the execution of $a_1, \ldots, a_n$ is referred to as *prediction*. Now if p is given, but the reasoning system has to find the appropriate $a_1, \ldots, a_n$ so that p 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 or diagnosis*. All the above kinds of reasoning may have to be done with partial or incomplete information in the knowledge base.

We now briefly discuss our use of a declarative programming language in implementing the above mentioned reasoning mechanisms.

3.1.2 Declarative Programming using AnsProlog In declarative programming the intent of a program ('what') is described in a particular declarative programming language and the interpreter figures out the 'how' part. For example, a declarative program for sorting a set of numbers just describes what sorting is (and does not say how to sort a set of numbers), and the interpreter when given a set of numbers, figures out the 'how' part and does the sorting. Although this approach could be inefficient (in terms of run-time) for certain kind of programs, it has acceptable performance for query answering and reasoning programs. Its strong point is that it is much quicker and easier (and is often more robust) to write a declarative program than a procedural one. For these reasons declarative programming is used for database querying and we will use it for the reasoning associated with our system. (For example, SQL is a declarative language as in SQL we specify what we want, and not how the system should look and search the files to find the answer to our queries.)

Earlier we mentioned that the kinds of queries we would like to ask are beyond the ability of database query languages. Moreover, we would like our system to be able to handle incomplete information, and be able to express normative and default statements. Among the various declarative and/or knowledge representation languages (most of which are declarative) that have been proposed one language that has the largest body of support structure (both theoretical results, efficient interpreters and developed applications) is the language of logic programming with the answer set semantics – simply referred to as AnsProlog in (Baral, 2003). Let us consider an example of AnsProlog encoding of signal networks.

$$holds(occ(A), T+1) \leftarrow action(A), time(T),$$
$$n\_triggers(S, A), holds(S, T), not\ holds(ab(A), T).$$

According to this rule, if A is an action that can be normally triggered by property S, and if at time point T, S is true but $ab(A)$ is false, then the action A will happen in the time $T+1$. The fluent $ab(A)$ encodes that "there is an exception to the rule". Hence, we can easily elaborate the biological exceptions by manipulating fluents such as $ab(A)$. Note that in the AnsProlog rule we use the operator *not*, which is different from the classical negation operator $\neg$. While $not\ f$ in the body of a rule intuitively means that f can not be proven to be true, $\neg f$ means that f is known to be not true. The operator *not* is very useful and powerful, and is one of the main features that make AnsProlog one of the most suitable knowledge representation languages.

3.2 Overview of the biological domain

In this work we represent and reason about some aspects of nuclear factor (NF)-κB signaling; namely its activation by the degradation of inhibitory κB (IκB), the tumor necrosis factor (TNF) induced activation of NFκB, pro- and anti-apoptotic signaling via TNF receptor 1 (TNF-R1), and the regulation of apoptotic signaling by FLICE-inhibitory protein (FLIP).

Biological signals within the cell are dependent upon association, interaction and binding of various proteins. For example,

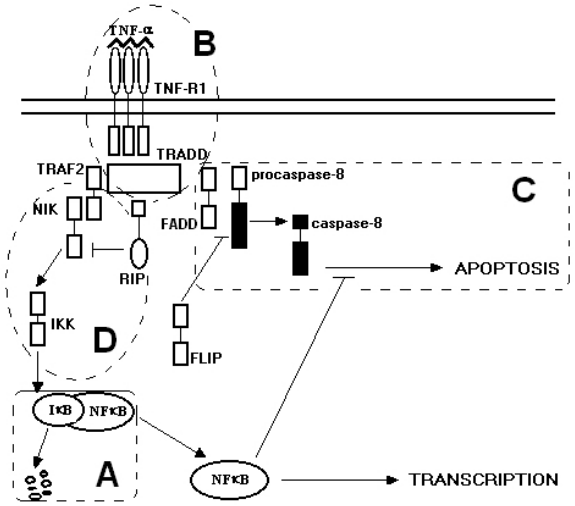

Fig. 1. NFκB-dependent signaling

NFκB is kept in the inactive state in the cytoplasm by association with IκB. IκB can be phosphorylated (at serine residuals Ser-32 and Ser-36), which leads to the ubiquitination and degradation of IκB by the proteosome. Once IκB has degraded, NFκB is free and becomes active. It then translocates to the nucleus and activates transcription of important genes (Sen & Baltimore, 1986; Verma et al., 1995; Baldwin, 1996). (Figure 1-A).

Tumor necrosis factor-α (TNF-α) can trigger both cell death and cell survival signals. By association with TNF-α ligand, the TNF-R1 undergoes trimerization. The death domains on TNF-R1 then recruit an adapter protein, TNF-R-associated death domain (TRADD). TRADD serves as a platform to recruit adaptor proteins, including the Fas-associated death domain protein (FADD), TNF-receptor-associated factor 2 (TRAF2), and receptor-interacting protein (RIP) (Aggarwal, 2003; Gaur & Aggarwal, 2003; Legler et al., 2003). (Figure 1-B).

The adapter protein FADD contains an N-terminal death effector domain (DED) that can bind to the homologous DEDs of procaspase-8. FADD recruits procaspase-8 to form a death inducing signal complex (DISC). Procaspase-8 is then cleaved to yield active caspase-8, which in turn triggers apoptosis (Nagata, 1999). (Figure 1-C).

IKK is an IκB kinase, which can induce NFκB activation by phosphorylating IκB. TRAF2 is capable of interacting with downstream signaling molecules such as NFκB-inducing kinase (NIK). NIK phosphorylates IKK at serine 176 thus activating IKK, which results in NFκB activation. RIP has been implicated to inhibit the NIK activity thus, inhibiting NFκB activation (Woronicz et al., 1997; Ling et al., 1998; Lin et al., 1998). (Figure 1-D).

4 DESCRIPTION OF BIOSIGNET-RR

Our system includes two components: (1) a script language by which users can conveniently describe signal network knowledge and queries; (2) an AnsProlog program to encode the knowledge and to compute answers for queries.

The system expects two input scripts from users: the first script encodes a signaling knowledge base and the second encodes a query. In the following we describe the script language through the example of NFκB signaling. Note that the keywords of the script language are in bold font.

4.1 The knowledge base script

The alphabet - the set of fluent and action symbols - of the knowledge base is defined by the following constructs

fluent $some\_property.$

action $some\_action.$

The negation of a fluent $some\_property$ is denoted by $-some\_property$. A conjunction of fluents is written as a list of the fluents separated by ';' .
The effect of an action is described in the general form

$$a \textbf{ causes } f \textbf{ if } f_1; \ldots ; f_k$$

where a is an action symbol, and $f_1, \ldots, f_k$ and f are fluent symbols.
A causal relationship between fluents is written as

$$g_1; \ldots ; g_l \textbf{ causes } g$$

where $g_1, \ldots, g_l$ and g are fluent symbols.
There are two types of triggers, which are written in the following forms

$$h_1; \ldots ; h_m \textbf{ n\_triggers } b$$
$$h'_1; \ldots ; h'_n \textbf{ triggers } b'.$$

Here, b and b' are actions; and h_i and h'_j are fluent symbols, where $1 \leq i \leq m$, $1 \leq j \leq n$.
We use **n_triggers** to state that a particular action is *normally* triggered. When an action is being normally triggered, it will not occur if it is also inhibited at the same time. On the other hand, a (not normally) triggered action will occur even if it is inhibited.
Finally, inhibitions of actions are described in the form

$$\phi_1; \ldots ; \phi_l \textbf{ inhibits } c$$

where $\phi_1, \ldots, \phi_l$ are fluents and c is an action. Note that the **n_triggers** construct allows one to be able to deal with incomplete knowledge about inhibitions in an elaboration tolerant manner. As more knowledge about additional inhibitory interactions become known one just has to add the new information.

Let us consider examples taken from the knowledge base script of NFκB signaling as follows.
A NFκB related signal starts with the binding of TNF-α ligand to TNF-R1 receptor. Upon the binding, TNF-R1 is trimerized. Hence, we define an action for the binding and a fluent for the receptor being trimzerized:

action $bind(tnf, rec)$.

fluent $trimerized(rec)$.

$bind(tnf, rec)$ **causes** $trimerized(rec)$.

The trimerization of TNF-R1 triggers the binding (recruitment) of TRADD to the receptor TNF-R1. Let $bind(rec, tradd)$ be the action that TRADD binds to TNF-R1, and let $bound(rec, tradd)$ be the fluent that TRADD is bound to TNF-R1. Then we write

action $bind(rec, tradd)$.

fluent $bound(rec, tradd)$.

$bind(rec, tradd)$ **causes** $bound(rec, tradd)$.

To say that trimerization of TNF-R1 triggers TRADD and TNF-R1 binding, we write

$trimerized(rec); -bound(rec, tradd)$
triggers $bind(rec, tradd)$.

The recruitment of TRAF2 to the TNF-R1 signaling complex *normally* triggers the interaction between TRAF2 and NIK, which causes NIK to be activated. Similarly to the above, the triggering is encoded in the script language as follows.

action $bind(tradd, traf)$.

action $interact(traf, nik)$.

fluent $bound(tradd, traf)$.

fluent $activated(nik, traf)$.

$bind(tradd, traf)$ **causes** $bound(tradd, traf)$.

$interact(traf, nik)$ **causes** $activated(nik, traf)$.

$bound(rec, tradd); bound(tradd, traf);$
$-activated(nik, traf)$ **n_triggers** $interact(traf, nik)$.

In the examples of explanation and planning in the next section, we will deal with some exceptions to this triggering.

4.2 The query script

A query script contains observations and queries. First, we need to define the initial situation and the chronological order of situations in which observations are made. For example, we have the initial situation labelled by t_0 , and some later situation t_1 .

initial t_0.

t_0 **precedes** t_1.

A situation can also be a number, which will be understood as a time point. The initial situation corresponds to the time point 0. Hence, we can use the number 0 instead of t_0 to denote the initial situation.

There are observations about properties of the cellular environment and about occurrences of actions. They take the general forms:

f **at** t.

a **occurs_at** t'.

Here, f is a fluent symbol, a is an action symbol and t and t' are some situations. For example, to state that at the beginning, NFκB is bound to IκB, we write

$bound(nf\kappa b, i\kappa b)$ **at** t_0.

We can write the observation about the initial binding of TNF-α to TNF-R1 receptor as follows.

$bind(tnf, rec)$ **occurs_at** t_0.

Currently BioSigNet-RR supports three kinds of queries, namely *prediction*, *explanation* and *planning*. In prediction, we query whether some property will *certainly* follow from the knowledge base and given observations. We usually want to know if some fluent (or its negation) is true at some situation, or *always* true from some situation, or *eventually* will be true after some situation. An example of prediction query is

predict **always** $-bound(fadd, procasp)$ **from** t_0.

The query asks if the protein FADD is never bound to procaspase-8.

In explanation, we want to find the unknown truth values of fluents in the initial situation, given the knowledge base and some observations. For example,

explain **always** $bound(nf\kappa b, i\kappa b)$ **from** t_0.

The above query asks about the value of fluents in the initial situation that will explain the observation that starting from the initial situation NFκB always remains bound to IκB.

In planning, we want to find a sequence of interventions to achieve a goal property. For example, starting from a well-defined initial situation, we would like to find a sequence of actions to keep NFκB bound to IκB.

plan **always** $bound(nf\kappa b, i\kappa b)$ **from** t_0.

Let us discuss in detail some examples of the reasoning about signaling pathways using the system.

5 REASONING WITH BIOSIGNET-RR

5.1 Prediction

FLIP belongs to the family of viral protein vFLIP and a related cellular protein cFLIP. FLIP contains death effector domains (DEDs) that are similar to the DEDs on FADD and procaspase-8. But the DEDs on FLIP cannot cleave themselves to yield caspase-8 (Gupta, 2002). One would predict that FLIP suppresses TNF-α-induced apoptosis via blocking activation of caspase-8. First, we describe biological facts about FLIP.

action $bind(fadd, flip)$.

fluent $bound(fadd, flip)$.

$bind(fadd, flip)$ **causes** $bound(fadd, flip)$.

$bound(fadd, flip)$ **inhibits** $bind(fadd, flip)$.

$bound(fadd, flip)$ **inhibits** $bind(fadd, procasp)$.

That is, FLIP can bind to FADD. Once FLIP is bound to FADD, the binding of FLIP to FADD as well as the binding of FADD to procaspase-8 is inhibited.

We describe the observation about the recruitment of FADD to TNF-R1 signaling complex. In the initial situation t_0 , it is observed that NFκB is *being bound* IκB and TNF-α *binds* to TNF-R1.

initial t_0.

$bound(nf\kappa b, i\kappa b)$ **at** t_0.

$bind(tnf, rec)$ **occurs_at** t_0.

Then, in some later situation t_1 and t_2, the occurrences of two actions are observed. In situation t_1, TRADD binds to the receptor TNF-R1; in situation t_2, FADD binds to TRADD.

t_0 **precedes** t_1.

t_1 **precedes** t_2.

$bind(rec, tradd)$ **occurs_at** t_1.

$bind(tradd, fadd)$ **occurs_at** t_2.

Finally, we query if the caspase-8 is always inactive after FLIP binds to FADD.

predict **always** $-active(caspase8)$ **after** $bind(fadd, flip)$ **from** t_0.

The output of BioSigNet-RR is as follows.

```
$./signet.pl nfkb-rep.sn prediction.sn
Evaluating prediction ...
Computing positive models ...
Positive model found!
Computing negative models ...
Negative model not found!
Answer: the  prediction is true.
```

5.2 Explanation

This is a hypothetical example adapted from (Heyninck & Beyaert, 2001). Imagine that we observe the recruitment of TRAF2 to TNF-R1 signaling complex. Then we expect that NFκB would be activated (by NIK-induced IKK activation). However, if no activity of NFκB is observed, we suspect a deregulation of the TRAF2-NFκB pathway, such as mutated form of TRAF2 proteins. The mutation of TRAF2 may have altered the ability of TRAF2 either to bind to TNFR or to recruit NIK, which is essential for NFκB activation. To test the hypothesis, we define a new fluent for TRAF2 being mutated and its causal relationship with the recruitment of NIK to TRAF2.

unknown $mutated(traf)$.

$mutated(traf)$ **inhibits** $interact(traf, nik)$.

The observation about the initial situation is the same as that in the previous example.

initial $t0$.

$bound(nf\kappa b, i\kappa b)$ **at** t_0.

$bind(tnf, rec)$ **occurs_at** t_0.

The recruitment of TRAF2 to TNF-R1 signaling complex is observed to happen as usual: first TRADD is recruited to TNFR1, then TRAF2 is recruited to TRADD.

t_0 **precedes** t_1.

t_1 **precedes** t_2.

$bind(rec, tradd)$ **occurs_at** t_1.

$bind(tradd, traf)$ **occurs_at** t_2.

Given the above observations, we query for the explanation why NFκB is always inactive; that is, why it is always bound to IκB.

explain **always** $bound(nf\kappa b, i\kappa b)$ **from** t_0.

The BioSigNet-RR answers as follows.

```
$./signet.pl nfkb-rep.sn explanation.sn
Finding explanation ...
Observations can be explained if ...
TRUE: mutated(traf).
```

NFκB is essential for regulating various cellular processes, namely proliferation, migration and survival. Dysregulation of this pathway has been reported in various diseases such as cancer. We have described one way NFκB pathway can malfunction. The same reasoning can be applied to other explanations such as mutation of IκB, IKK, or NIK.

5.3 Planning

As previously discussed, blocking TRAF2's function, such as the presence of dominant-negative mutant TRAF2 or introduction of small inhibitory peptides or chemical compounds against TRAF2 proteins, results in NFκB inactivation. We would define inhibition of TRAF2 as an intervention to regulate NFκB inactivation. Disruption of TRAF2 function by these effects would result in the inability of TRAF2 to bind to TRADD and to interact with NIK.
We define an action $intro(dnm\_traf)$ for the introduction of *dominant-negative mutant* TRAF2. The action causes the mutant TRAF2 to be present, which is denoted by a fluent $present(dnm\_traf)$. This is described in BioSigNet-RR script language as follows.

fluent $present(dnm\_traf)$.

intervention $intro(dnm\_traf)$.

$intro(dnm\_traf)$ **causes** $present(dnm\_traf)$.

The inhibitory effects of mutant TRAF2 are described by the following inhibition rules.

$present(dnm\_traf)$ **inhibits** $bind(tradd, traf)$.

$present(dnm\_traf)$ **inhibits** $interact(traf, nik)$.

If the mutant TRAF2 is introduced too late, it would not have the desired effect on NFκB activation. If we do not want to interfere with the association of TRADD and TRAF2, then we should not disrupt TRAF2's function at the beginning. Hence, we have to find a plan to intervene at the right time.
The planning query includes *an initial condition* and *a goal*. We can have an initial condition that NFκB is bound to IκB and TNF-α comes to bind to TNF receptor. This is written in BioSigNet-RR as:

$bound(nf\kappa b, i\kappa b)$ **at** 0.

$bind(tnf, rec)$ **occurs_at** 0.

We want to find ways to keep NFκB inactive, which is written in BioSigNet-RR as

plan **always** $bound(nf\kappa b, i\kappa b)$ **from** 0.

The BioSigNet-RR will find such a plan:

```
$./signet.pl nfkb-rep.sn planning.sn
Planning ...
Plan found ...
intro(dnm_traf) at 3
```

The plan corresponds to the scenario where TRADD binds to TNF-R1 at time 1, TRAF2 is recruited to TRADD at time 2 and the disruption of TRAF2 function occurs at time 3. Note that we are talking about logical time (ordering of events), not biological time.

Planning for interventions may have an important application in drug therapy design. For example, a series of interventions aiming at regulation of a NFκB signaling pathway can be considered as part of a cancer drug therapy.

6 CONCLUSION

We have proposed applying knowledge representation and reasoning methodologies to the important problem of representing and reasoning about signaling networks. We have developed a preliminary system called BioSigNet-RR for this goal. Our system is grounded on recent research in reasoning about actions and declarative programming.

We showed how BioSignet-RR can describe a portion of NFκB signaling networks and answer interesting queries on prediction, explanation and planning. We have provided a glimpse into the AnsProlog implementation to illustrate how this approach can deal with elaboration tolerance and incomplete information.

As the next step, we plan to apply our method to represent and reason about bigger networks such as Kohn's map (Kohn, 1999). The biological facts in our examples have been manually extracted from biological and bioinformatic sources. This manual extraction should become automated in the future. We will also address an important issue with logical frameworks in modelling biological systems, namely the ability to model quantitative and resource sensitive information.

Our work is an initial step toward the ultimate goal of modelling signal network knowledge. We believe our approach is feasible and fruitful, because it is based on a solid theoretical basis. Our knowledge representation and reasoning based approach can support many features that are important not only for modelling signaling networks but also for modelling biological systems in general. Those features include elaboration tolerance, and modelling and reasoning with incomplete information.

ACKNOWLEDGEMENTS

The authors would like to thank Professor Subbarao Kambhampati for coining the term BioSigNet. The first three authors acknowledge the support of NSF through grant number 0070463. Chitta Baral acknowledges the support of TGen and the college of Engineering of ASU for supporting him during the summer of 2003, Professor Shankar Subramaniam for introducing him to signal transduction and Professors Charlie Colbourn and Sudhir Kumar for supporting him through Proposition 301 funds during Spring 2003. Karen Chancellor acknowledges the support of an NSF graduate fellowship, and A. Joy and M. Berens acknowledge the support of NIH and NINDS.

REFERENCES

Aggarwal, B. B. (2003). Signalling pathways of the TNF superfamily: A double-edge sword. *Nature Reviews Immunology*, **3**, 745–756.

Antoniotti, M., Park, F., Policriti, A., Ugel, N., & Mishra, B. (2003). Foundations of a query and simulation system for the modeling of biochemical and biological processes. In *Pacific Symposium on Biocomputing 2003 (PSB 2003)*, page 116–127.

Baldwin, A. S. (1996). The NF-κB and IκB proteins: New discoveries and insights. *Annual Review of Immunology*, **14**, 649–681.

Baral, C. (2003). *Knowledge representation, reasoning and declarative problem solving*. Cambridge University Press.

Chabrier, N. & Fages, F. (2003). Symbolic model checking of biochemical networks. In C. Priami, editor, *Computational Methods in Systems Biology (CMSB 2003)*, pages 149–162.

de Jong, H. (2002). Modeling and simulation of genetic regulatory systems: A literature review. *Journal of Computational Biology*, **9**(1), 67–103.

Eker, S., Knapp, M., Laderoute, K., Lincoln, P., Meseguer, J., & Sonmez, K. (2002). Pathway logic: symbolic analysis of biological signaling. In *Pacific Symposium on Biocomputing 2002 (PSB 2002)*, page 400–412.

Gaur, U. & Aggarwal, B. B. (2003). Regulation of proliferation, survival and apoptosis by members of the TNF superfamily. *Biochemical Pharmacology*, **66**, 1403–1408.

Gelfond, M. & Lifschitz, V. (1993). Representing actions and change by logic programs. *Journal of Logic Programming*, **17**(2,3,4), 301–323.

Gupta, S. (2002). A decision between life and death during TNF-α-induced signaling. *Journal of Clinical Immunology*, **22**(4), 185–194.

Heyninck, K. & Beyaert, R. (2001). Crosstalk between NFκB-activating and apoptosis-inducing proteins of the TNF-receptor complex. *Molecular Cell Biology Research Communications*, **4**(5), 259–265.

Karp, P. D., Riley, M., Saier, M., Paulsen, I. T., Paley, S. M., & Pellegrini-Toole, A. (2000). The ecocyc and metacyc databases. *Nucleic Acids Research*, **28**(1), 56–59.

Kohn, K. W. (1999). Molecular interaction map of the mammalian cell cycle control and DNA repair systems. *Molecular Biology of the Cell*, **10**, 2703–2734.

Legler, D. F., Micheau, O., Doucey, M.-A., Tschopp, J., & Bron, C. (2003). Recruitment of TNF receptor 1 to lipid Rafts is essential for TNFα-mediated NF-κB activation. *Immunity*, **18**, 655–664.

Lin, X., Mu, Y., Cunningham, E. T., Marcu, K. B., Geleziunas, R., & Greene, W. C. (1998). Molecular determinants of NF-κB-inducing kinase action. *Molecular and Cellular Biology*, **18**(10), 5899–5907.

Ling, L., Cao, Z., & Goeddel, D. V. (1998). NF-κB-inducing kinase activates IKK-κ by phosphorylation of Ser-176. In *Proceedings of the National Academy of Sciences(PNSA)*, volume 95, pages 3792–7.

Mishra, B. (2002). A symbolic approach to modelling cellular behaviour. In V. Prasanna, S. Sahni, and U. Shukla, editors, *High Performance Computing–HiPC 2002. LNCS*, volume 2552, pages 725–732. Springer-Verlag.

Nagata, S. (1999). Fas ligand-induced apoptosis. *Annual Review of Genetics*, **33**, 29–55.

Ogata, H., Goto, S., Sato, K., Fujibuchi, W., Bono, H., & Kanehisa, M. (1999). Kegg: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Research*, **27**(1), 29–34.

O.Voit, E. (2000). *Computational Analysis of Biochemical Systems*. Cambridge University Press.

Peleg, M., Yeh, I., & Altman, R. B. (2002). Modelling biological processes using workflow and petri net models. *Bioinformatics*, **18**(6), 825–837.

Priami, C., editor (2003). *Computational Methods in Systems Biology. LNCS*, volume 2602. Springer Verlag.

Reddy, V. N., Liebman, M. N., & Mavrovouniotis, M. L. (1996). Qualitative analysis of biochemical reaction systems. *Computers in Biology and Medicine*, **26**, 9–24.

Regev, A., Silverman, W., & Shapiro, E. (2001). Representation and simulation of biochemical processes using π-calculus process algebra. In *Pacific Symposium on Biocomputing 2001 (PSB 2001)*, pages 459–470.

Rieter, R. (2001). *Knowledge in action : logical foundations for specifying and implementing dynamical systems*. MIT Press.

Sambrano, G. R. (2003). Developing a navigation and visualization system for signaling pathways. *AfCS Reports. http://www.signaling-gateway.org/reports/v1/DA0009.pdf*.

Sen, R. & Baltimore, D. (1986). Inducibility of κ-immunoglobulin enhancer-binding protein NF-κB by a posttranslational mechanism. *Cell*, **47**, 921–928.

Talcott, C., Eker, S., Knapp, M., Lincoln, P., & Laderoute, K. (2004). Pathway logic modeling of protein functional domains in signal transduction. In *Pacific Symposium on Biocomputing 2004 (PSB 2004)*, pages 568–580.

Tran, N. & Baral, C. (2004). Reasoning about triggered actions in AnsProlog and its application to molecular interactions in cells. In *Ninth International Conference on the Principles of Knowledge Representation and Reasoning (KR2004)*.

Verma, I. M., Stevenson, J. K., Schwartz, E. M., Antwerp, D. V., & Miyamoto, S. (1995). Rel/NF-κB/IκB family: intimate tales of association and dissociation. *Genes & Development*, **9**, 2723–2735.

Woronicz, J. D., Gao, X., Cao, Z., Rothe, M., & Goeddel, D. V. (1997). IκB kinase-κ: NF-κB activation and complex formation with IκB kinase-κ and NIK. *Science*, **278**, 866–870.