Analysis of Signalling Pathways Using Continuous Time Markov Chains

Muffy Calder,[1] Vladislav Vyshemirsky,[2] David Gilbert,[2]
Richard Orton[2]

[1] Department of Computing Science, University of Glasgow
[2] Bioinformatics Research Centre, University of Glasgow

25th May 2006

Keywords signalling pathways; stochastic processes; continuous time Markov chains; model checking; continuous stochastic logic.

Abstract

We describe a quantitative modelling and analysis approach for signal transduction networks.

We illustrate the approach with an example, the RKIP inhibited ERK pathway [CSK$^+$03]. Our models are high level descriptions of continuous time Markov chains: proteins are modelled by synchronous processes and reactions by transitions. Concentrations are modelled by discrete, abstract quantities. The main advantage of our approach is that using a (continuous time) stochastic logic and the PRISM model checker, we can perform quantitative analysis such as *what is the probability that if a concentration reaches a certain level, it will remain at that level thereafter?* or *how does varying a given reaction rate affect that probability?* We also perform standard simulations and compare our results with a traditional ordinary differential equation model. An interesting result is that for the example pathway, only a small number of discrete data values is required to render the simulations practically indistinguishable.

1 Introduction

Signal transduction pathways allow cells to sense an environment and make suitable responses. External signals detected by cell membrane receptors activate a sequence of reactions, allowing the cell to recognise the signal and pass it into the nucleus. The cellular response is then activated inside the nucleus. This signalling mechanism is involved in a number of important processes, such as proliferation, cell growth, movement, apoptosis, and cell communication. The pathways include feedback and may be embedded in more complex networks, some form of automated analysis required.

Our aim is to develop quantitative techniques for signal transduction pathway[1] modelling and analysis, based on continuous time and semiquantitative data. Our models are distinctive in two ways. First, we model populations (rather than individuals), that is we model molar concentrations (rather than molecules). Second, we model semiquantitative data. Contemporary methods for biochemical experiments do not, in general, permit the measurement of absolute or continuous values of concentrations. Consequently, some quantitative models are over constrained. We avoid this by considering discrete, abstract concentrations. Thus our treatment of time is continuous, but concentrations are discrete. Our analysis is (model) checking for quantitative, temporal, biological queries.

Our modelling approach is motivated by the observation that signalling pathways have *stochastic*, *computational* and *concurrent* behaviour. Our models are continuous time Markov chains (CTMCs). They are defined, in a natural way, by high level descriptions of concurrent processes: the processes correspond to proteins (the reactants in the pathway) and the transitions to reactions. Concentrations are modelled by discrete, abstract quantities. We use a continuous stochastic logic and the probabilistic symbolic model checker PRISM [KNP02] to express and check a variety of temporal queries for both transient behaviours and steady state behaviours. We can also perform standard simulations and so we compare our results with a traditional ordinary differential equation model. Throughout, we illustrate our approach with an example pathway: the RKIP inhibited ERK pathway [CSK$^+$03].

The paper is organised as follows. In section 2 we present the example pathway. The CTMC model is developed in section 3. In the following section, we discuss analysis by model checking, we present four types of probabilistic, temporal query and give instances for the example pathway. We express the queries in the continuous stochastic logic CSL [BHHK00, ASSB00], and check their validity. In section 5 we discuss how simulations in the stochastic setting compare with simulations in the deterministic setting: in a MATLAB® implementation of ordinary differential equations. In section 6 we discuss our results and we review related work in section 7. We conclude in section 8.

2 RKIP and the ERK pathway

The example system we consider is the RKIP inhibited ERK pathway. We give only a brief overview, further details are presented in [CSK$^+$03, CGH04].

The ERK pathway (also called Ras/Raf, or Raf-1/MEK/ERK pathway) is a ubiquitous pathway that conveys mitogenic and differentiation signals from the cell membrane to the nucleus. The kinase inhibitor protein RKIP inhibits activation of Raf and thus can “dampen” down the ERK pathway.

[1] In this paper we use the terms pathway and network synonymously.

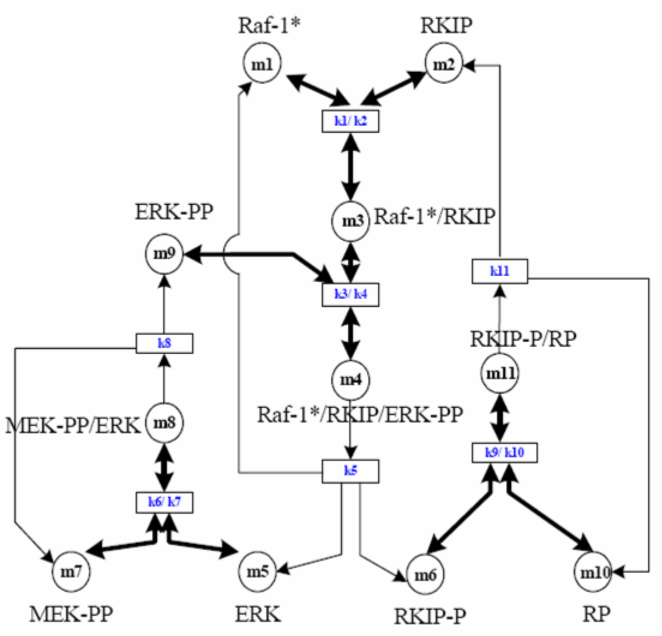

Figure 1: RKIP inhibited ERK pathway

We consider the pathway as given in the graphical representation of Figure 1. This figure is taken from [CSK$^+$03], where a number of nonlinear ordinary differential equations (ODEs) representing the kinetics are given. We take Figure 1 as our starting point, and explain informally, its meaning. Each node is labelled by a protein (or species). For example, Raf-1*, RKIP and Raf-1*/RKIP are proteins, the last being a complex built up from the first two. A suffix -P or -PP denotes a (single or double, resp.) phosphorylated protein, for example MEK-PP and ERK-PP. Each protein has an associated concentration, given by $m1$, $m2$ etc. *Reactions* define how proteins are built up and broken down. In Figure 1, bi-directional arrows correspond to both forward and backward reactions; uni-directional arrows to forward reactions. Each reaction has a rate given by the rate constants $k1$, $k2$, etc. These are given in the rectangles, with $kn/kn+1$ denoting that kn is the forward rate and $kn+1$ the backward rate. Initially, all concentrations are unobservable, except for m_1, m_2, m_7, m_9, and m_{10} [CSK$^+$03].

The dynamic behaviour of the pathway is quite complex, because proteins are involved in more than one reaction and there are several feedbacks. In the next section we develop a model which captures that dynamic behaviour. We note that the example system is part of a larger pathway which can be found elsewhere [KDMH99, SEJGM02].

3 Modelling signalling networks by CTMCs

In this section we describe how we model concentrations of proteins by discrete variables, and the dynamic behaviour of proteins by computational processes.

3.1 Discrete concentrations

Each protein defined in a network has a molar concentration which changes with time, i.e. $m = f(t)$, where m is a concentration of the protein and t is time. As we have indicated earlier, there is a difficulty in obtaining precise and/or continuous concentration values using the methods of contemporary biochemistry. We therefore make discrete abstractions as follows. When the maximum molar concentration is M, then for a given N, the abstract values $0 \ldots N$ represent the concentration intervals $[0, 1{*}M/N), [1{*}M/N, 2{*}M/N), \ldots [N-1{*}M/N, N{*}M/N]$. We refer to $0 \ldots N$ as *levels* of concentration.

We note that we could define a different N for each protein (depending on experimental accuracy for that species) but in this paper, without loss of generality, we assume the same N, for all proteins.

3.2 Proteins as processes

We associate a concurrent, computational process with each of the proteins in the network and define these processes using the PRISM modelling language. This language allows the definition of systems of concurrent processes which when synchronised, denote continuous time Markov chains (CTMCs). In sections 3.3 and 3.4 we discuss in detail how CTMCs provide a natural semantics for signalling networks; in this section we focus on the way in which the proteins are represented by PRISM processes (modules) and reactions are represented by transitions.

Below, we give a brief overview of the language, illustrating each concept with a simple example; the reader is directed to [KNP02] for further details of PRISM.

Transitions are labelled with performance rates and (optional) names. For each transition, the performance rate is defined as the parameter λ of an exponential distribution of the transition duration. A key feature is synchronisation: concurrent processes are synchronised on transitions with common names (i.e. the transitions occur simultaneously). Transitions with distinct names are not synchronised. The performance rate for the synchronised transition is the *product* of the performance rates of the synchronising transitions. For example, if process A performs α with rate λ_1, and process B performs α with rate λ_2, then the performance rate of α when A is synchronised with B is $\lambda_1 \cdot \lambda_2$.

As an example, consider the simple single reaction of Figure 2 which describes the binding of (active) Raf-1*, called RAF1 henceforth, and RKIP. Call this reaction $r1$.

The PRISM model for this system is listed as Model 1. The model begins with the keyword *stochastic* and consists of some preliminary constants (N and

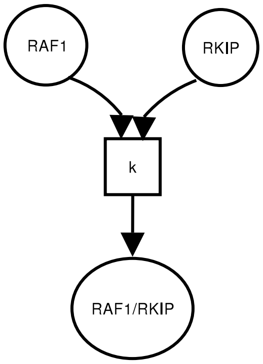

Figure 2: Simple biochemical reaction $r1$

R), four modules: $RAF1$, $RKIP$, $RAF1/RKIP$, and $Constants$, and a system description which states that the four modules should be run concurrently. The constant N defines the concentration levels, as discussed in section 3.1; R is simply an abbreviation for N^{-1}. Consider the first three modules which represent the proteins $RAF1$, $RKIP$ etc. Each module has the form: a state variable which denotes the protein concentration (we use the same name for process and variable, the type can be deduced from context) followed by a single transition named $r1$. The transition has the form *precondition* → *rate: assignment*, meaning when the precondition is true, then perform the assignment at the given rate. The rate for transitions of the first two modules is protein concentration multiplied by R, the rate for the third is 1. The assignments in the first two modules decrease the protein level by 1; the level is increased by 1 in the third module. These correspond to the fact that the rate of the reaction is determined by the concentrations of the reactants, and the reactants are *consumed* in the reaction to *produce* $RAF1/RKIP$. But, we must not forget that there is a fourth module, $Constants$; this simply defines the constants for reaction kinetics. In this case the module contains a “dummy” state variable called x, and one (always) enabled transition named $r1$ which defines the rate (i.e. $0.8/R$) for the transition $r1$. For readabilty, our variable and process names include the character ‘/’, strictly speaking, this is not allowed in PRISM.

Since all four transitions have the same name, they will all *synchronise*, and when they do, the resulting transition has rate $\frac{N \cdot R \cdot N \cdot R \cdot 0.8}{R} = 2.4$ ($RAF1$ and $RKIP$ are initialised to N, $RAF1/RKIP$ is initialised to 0, $N = 3$ and $R = 1/3$).

In this simple PRISM model, all the proteins are involved in only one reaction. The reaction can occur three times, until all the $RAF1$ and $RKIP$ has been consumed. Thus in the underlying CTMC, there are 3 transitions (plus a loop) over 4 states, as indicated in Figure 3. The states are labelled by tuples representing the (molar) concentrations of $RAF1$, $RKIP$ and $RAF1/RKIP$, respectively. The edges are labelled with the transition rates.

Model 1 RAF1 binding to RKIP

```
stochastic

const int N = 3;
const double R = 1/N;

module RAF1
    RAF1: [0..N] init N;
    [r1] (RAF1>0) -> RAF1*R:
           (RAF1' = RAF1 - 1);
endmodule

module RKIP
    RKIP: [0..N] init N;
    [r1] (RKIP>0) -> RKIP*R:
           (RKIP' = RKIP - 1);
endmodule

module RAF1/RKIP
    RAF1/RKIP: [0..N] init 0;
    [r1] (RAF1/RKIP < N) -> 1:
           (RAF1/RKIP' = RAF1/RKIP + 1);
endmodule

module Constants
    x: bool init true;
    [r1] (x=true) -> 0.8/R:
           (x'=true);
endmodule

system
    RAF1 || RKIP || RAF1/RKIP || Constants
endsystem
```

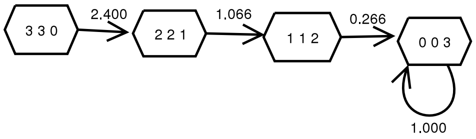

Figure 3: CTMC for Model 1

In the RKIP inhibited ERK pathway, each protein is involved in several reactions. We model this quite easily by introducing different names $(r1, r2, \ldots)$ for each reaction (and the corresponding transitions). We use the convention that reaction rx has rate parameter kx.

Notice that we can describe all the transitions of the processes independently of the number of concentration levels: we simply make the appropriate comparison (in the precondition). The size of the complete underlying CTMC depends on N, some examples are:

- when $N = 3$ there are 273 states and $1,316$ transitions;
- when $N = 5$ there are $1,974$ states and $12,236$ transitions;
- when $N = 9$ there are $28,171$ states and $216,282$ transitions.

The full PRISM model for the RKIP inhibited pathway is given in Appendix A.1. In the full model, R is calibrated by the initial concentration(s), i.e. $R = 2.5/N$.

3.3 Continuous Time Markov Chains as models

The PRISM description allows us to focus on the overall structure of the stochastic system, whilst saving us from the detail of defining the large and complex underlying CTMC.

In this section, we give more detail of the underlying CTMCs and why they provide good, sound models for signalling networks. We assume some familiarity with Markov chains, for completeness we give the following definition of a CTMC.

Definition Given a finite set of atomic propositions AP, a continuous time Markov chain (CTMC) is a triple (S,R,L) where

- S is a finite set of states
- L: $S \rightarrow 2^{AP}$ is labelling of states
- R: $S \times S \rightarrow \Re_{\geq 0}$ is a rate matrix.

For a given state s, when $R(s, s') > 0$ for more than one s', then there is a *race* between the outgoing transitions from s. The rates determine *probabilities* according to the "memoryless" negative exponential: when $R(s, s') = \lambda$, then the probability that transition from s to s' completes within time t is $1 - e^{-\lambda t}$.

A path through a CTMC is an alternating sequence $\sigma = s_0 t_0 s_1 t_1 \ldots$ such that $(s_i, s_{i+1}) \in R$ and each time stamp t_i denotes the time spent in state s_i, $\forall i$.

In the case of PRISM system descriptions, the atomic propositions refer to PRISM variables, for example atomic propositions include $RAF1 = 0$, $RKIP = 1$, etc. CTMCs are often presented using a graphical notation, as in Figure 3.

3.4 Soundness of stochastic model

In this section we explain why the underlying CTMCs are sound models for signalling networks; in particular, we show how the rates associated with transitions relate to mass action kinetics.

By way of illustration, consider the simple reaction in Figure 2. Recall that for each reaction, a protein is either a *producer* or a *consumer*; thus in the PRISM representation, producers have their concentrations decreased, and consumers have their concentrations increased. Now consider the equations for standard reaction (mass action) kinetics, given by the following:

$$\begin{cases} \dfrac{dm_1}{dt} = -k \cdot m_1 \cdot m_2 \\ \dfrac{dm_2}{dt} = -k \cdot m_1 \cdot m_2 \\ \dfrac{dm_3}{dt} = k \cdot m_1 \cdot m_2 \end{cases}$$

where m_1,m_2, and m_3 are the concentrations of RAF1, RKIP, and RAF1/RKIP respectively.

In the CTMC denoted by the PRISM model (Model 1), from the initial state, the first transition is the synchronisation of four processes ($RAF1$, $RKIP$, $RAF/RKIP$ and *Constants*). Recall the rates for synchronising actions in PRISM are *multiplied*, so the first transition has rate

$$(RAF1 \cdot R) \cdot (RKIP \cdot R) \cdot (k \cdot N) \tag{1}$$

where $k = 0.8$. The crucial question is is *how does this rate compare with, or relate to, the standard mass action semantics?*

First, consider how the concentration variables relate to each other: the ODEs above refer to continuous concentrations, whereas our model has *discrete* natural number levels. Let m be a continuous variable and let m^d be the corresponding PRISM variable (e.g. $RAF1$, $RAF/RKIP$). Then

$$m = m^d \cdot R = m^d \cdot \frac{1}{N} \tag{2}$$

Second, derive a rate expressed in terms of the PRISM variables. From the continuous rate:

$$\frac{dm_3}{dt} = k \cdot m_1 \cdot m_2 \tag{3}$$

the simplest way to derive a new concentration m_3' from m_3 is by Euler's method thus:

$$m_3' = m_3 + (k \cdot m_1 \cdot m_2 \cdot \Delta t) \tag{4}$$

But the abstract (PRISM) concentrations can only increase in units of 1 level of molar concentration, or $\frac{1}{N}$ molars, so

$$\Delta t = \frac{1}{k \cdot m_1 \cdot m_2 \cdot N} \tag{5}$$

PRISM implements rates as the "memoryless" negative exponential, that is for a given rate λ, $P(t) = 1 - e^{-\lambda t}$ is the probability that the action will be completed before time t. Taking λ as $\frac{1}{\Delta t}$, in this example we have

$$\lambda = k \cdot m_1 \cdot m_2 \cdot N \tag{6}$$

Replacing the continuous variables by their abstract forms, we have

$$\lambda = k \cdot (m_1^d \cdot R) \cdot (m_2^d \cdot R) \cdot N \tag{7}$$

or

$$\lambda = k \cdot (RAF1 \cdot R) \cdot (RKIP \cdot R) \cdot N \tag{8}$$

which is exactly the rate given in (1) above.

We can generalise this to arbitrary reactions as follows. In the PRISM description, for a given reaction r, let C^d and P^d be the set of variables representing consumers and producers, respectively. There is a r-named transition defined for each member of C^d and P^d; together they synchronise as a single r-named transition in the underlying CTMC. Let the transition representing that reaction be described by $\lambda : \bigwedge P$, where P is the set of the atomic propositions holding in the target state. Assume λ is defined as described above. The underlying CTMC is given by:

$$\forall c_l \in C_l.\ \forall p_l \in P_l.\ \bigwedge(c_l > 0) \Rightarrow \lambda : \bigwedge(c_l' = c_l - 1) \bigwedge(p_l' = p_l + 1) \tag{9}$$

Note the use of $\Rightarrow$ (logical implication), as opposed to $\rightarrow$ in the PRISM description of a transition. Note also that in the PRISM model, we include the rate factor $(\cdot N)$ in the *Constants* module, and multiply the concentrations by R in the protein processes.

In the next section we give a more detailed example.

3.5 A more complex example

Consider the network given in Figure 4; this is more complex than the single reaction given in Figure 2, in particular, some of the arrows are bidirectional.

Assume $N = 2$. The CTMC model for this network is given in Figure 5, tuples of concentration $\langle m_1\ m_2\ m_3\ m_4\ m_5 \rangle$ label the states. Since one reaction is reversible, there are simple loops in this topology. (More complicated pathway topologies can produce more nontrivial loops.)

Assume rate constants $k1 = 0.57$, $k2 = 0.02$, and $k3 = 0.31$ and initial concentrations $m_1(0) = m_2(0) = m_3(0) = 2$ and $m_4(0) = m_5(0) = 0$. The performance rates for the transitions are calculated as follows:

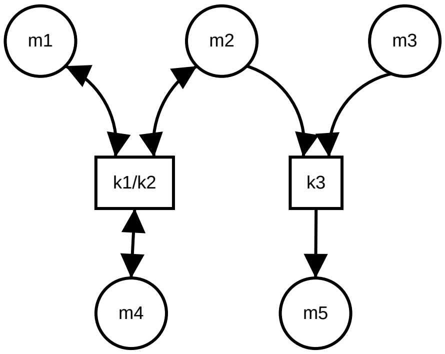

Figure 4: Network with three reactions, 5 proteins

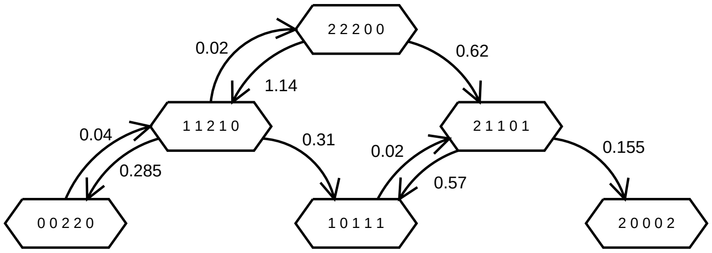

Figure 5: CTMC for network in Figure 4

- $\langle 2\ 2\ 2\ 0\ 0 \rangle \longrightarrow \langle 1\ 1\ 2\ 1\ 0 \rangle$: $\lambda = \frac{k_1 \cdot \frac{m1}{N} \cdot \frac{m2}{N}}{\frac{1}{N}} = \frac{k_1 \cdot \frac{2}{2} \cdot \frac{2}{2}}{\frac{1}{2}} = 2 \cdot k_1 = 1.14$
- $\langle 1\ 1\ 2\ 1\ 0 \rangle \longrightarrow \langle 2\ 2\ 2\ 0\ 0 \rangle$: $\lambda = \frac{k_2 \cdot \frac{m4}{N}}{\frac{1}{N}} = \frac{k_2 \cdot \frac{1}{2}}{\frac{1}{2}} = k_2 = 0.02$
- $\langle 2\ 2\ 2\ 0\ 0 \rangle \longrightarrow \langle 2\ 1\ 1\ 0\ 1 \rangle$: $\lambda = \frac{k_3 \cdot \frac{m2}{N} \cdot \frac{m3}{N}}{\frac{1}{N}} = \frac{k_3 \cdot \frac{2}{2} \cdot \frac{2}{2}}{\frac{1}{2}} = 2 \cdot k_3 = 0.62$

Other performance rates can be calculated exactly in the same way, the results are given in Figure 5.

In conclusion, we propose that CTMCs are good models for networks because they allow us to model performance and nondeterminism explicitly; however, it would be unrealistic to construct a CTMC model manually. The high level modelling abstractions of PRISM allow us to separate system structure from performance and to generate the appropriate underlying rates automatically. Moreover, we can *model check* stochastic properties of CTMCs, a more powerful reasoning mechanism than (stochastic) simulation. For example, consider evaluation of the probability that the state $\langle 2\ 0\ 0\ 0\ 2 \rangle$ is reachable. We cannot determine this probability by simulation, since there are an infinite number of paths leading to this state. However, the PRISM model checking algorithms can compute the probabilities (e.g. in the steady state) *automatically*, as we will consider in the next section.

4 Analysis

Temporal logics are powerful tools for expressing temporal queries which may be generic (e.g. state reachability, deadlock) or application specific (e.g. referring to variables representing application characteristics). For example, we can express queries such as *what is the probability that a protein concentration reaches a certain level, and then remains at that level thereafter?*, or *if we vary the rate of a particular reaction, how does this impact that probability?* Whereas simulation is the exploration of a *single* behaviour over a given time interval, model checking allows us to investigate the truth (or otherwise) of temporal queries over (possibly infinite) sets of behaviours over (possibly) unbounded time intervals.

Since we have a stochastic model, we employ the logic CSL (Continuous Stochastic Logic) [BHHK00, ASSB00], and the symbolic probabilistic model checker PRISM [PNK04] to compute validity. We can not only check validity of logical properties, but using PRISM we can analyse open formulae, i.e. we can perform *experiments* as we vary instances of variables in a formula expressing a property. Typically, we will vary reaction rates or concentration levels.

CSL is a continuous time logic that allows one to express a probability measure that a temporal property is satisfied, in either transient behaviours or in steady state behaviours. We assume a basic familiarity with the logic, it is based upon the computational tree logic CTL. The operators of CSL are given in Table 1, for more details see [PNK04]. The $P_{\bowtie p}[\phi]$ properties are *transient*, that is, they depend on time; $S_{\bowtie p}[\phi]$ properties are *steady state*, that is they hold in the long run. To check the latter properties, we use a linear algebra package in PRISM to generate the steady state solution. Note that in this context steady state solutions are not (generally) single states, rather a network of states (with cycles) which define the probability distributions in the long run.

$\bowtie p$ specifies a bound, for example $P_{\bowtie p}[\phi]$ is true in a state s if the probability that ϕ is satisfied by the paths from state s meets the bound $\bowtie p$. Examples of bounds are > 0.99 and < 0.01. A special case of $\bowtie p$ is no bound, in which case we calculate a probability. We write $P_{=?}[\psi]$ which returns the probability, from the initial state, of ψ. If we don't want to start at the initial state, we can apply a filter thus: $P_{=?}[\psi\{\phi\}]$, which returns the probability, from the state satisfying ϕ, of ψ. (Note: if more than one state satisfies ϕ then the first one in a lexicographic ordering is chosen.)

Operator	**CSL Syntax**
True	$true$
False	$false$
Conjunction	$\phi \wedge \phi$
Disjunction	$\phi \vee \phi$
Negation	$\neg\phi$
Implication	$\phi \Rightarrow \phi$
Next	$P_{\bowtie p}[\mathbf{X}\phi]$
Unbounded Until	$P_{\bowtie p}[\phi\mathbf{U}\phi]$
Bounded Until	$P_{\bowtie p}[\phi\mathbf{U}^{\leq t}\phi]$
Bounded Until	$P_{\bowtie p}[\phi\mathbf{U}^{\geq t}\phi]$
Bounded Until	$P_{\bowtie p}[\phi\mathbf{U}^{[t_1,t_2]}\phi]$
Steady-State	$S_{\bowtie p}[\phi]$

Table 1: Continuous Stochastic Logic operators

In the next section we use CSL and PRISM to formulate and check a number of biological queries against our model for the RKIP inhibited ERK pathway. We consider four different kinds of temporal property:

1. steady state analysis of stability of a protein, i.e. a protein reaches a level and then remains there, within certain bounds,

2. transient analysis of monotonic decrease of a protein, i.e. the levels of a protein only decrease,

3. steady state analysis of protein stability when varying reaction rates, i.e. a protein is more likely to be stable for certain reaction rates,

4. transient analysis of protein activation sequence, i.e. concentration peak ordering.

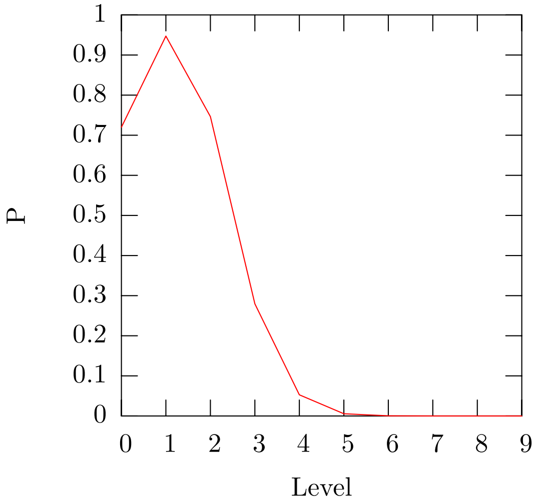

Figure 6: Stability of $RAF1$ wrt C in steady state

4.1 Stability of protein in steady state

This type of property is particularly applicable to the analysis of networks where transient and sustained signal responses can produce markedly different cellular outcomes. For example, a transient signal could lead to cell proliferation, whereas a sustained signal would result in differentiation.

Consider the concentration of $RAF1$. Stability for this protein (within bounds $C-1, C+1$) is expressed by the CSL formula:

$$S_{=?}[(RAF1 \geq C-1) \wedge (RAF1 \leq C+1)] \tag{10}$$

In other words, the level of $RAF1$ is at most 1 increment/decrement away from C. The results are given Figure 6, with C ranging over $0 \ldots 9$ ($N = 9$). They illustrate that $RAF1$ is most likely to be stable at level 1, with a relatively high probability of stability at levels 0 and 2. It is unlikely to be stable at levels 3 or more.

4.2 Monotonic decrease of protein

This type of property expresses the notion that the system does not allow an accumulation of a protein. To decide it, consider two properties. The first property is:

$$P_{\geq 1}[(true)\mathbf{U}((Protein = C) \wedge (P_{\geq 0.95}[\mathbf{X}(Protein = C-1)]))] \tag{11}$$

This property expresses "Is it possible to reach a state in which $Protein$ concentration is at level C and after the next step this concentration is $C-1$

with the probability $\geq 95\%$?". Figure 7 illustrates evaluating this property for $RAF1$, with $N = 9$ and C ranging over $1 \ldots 9$: the result is *false* for levels 1 to 4 and *true* for levels 5 to 9.

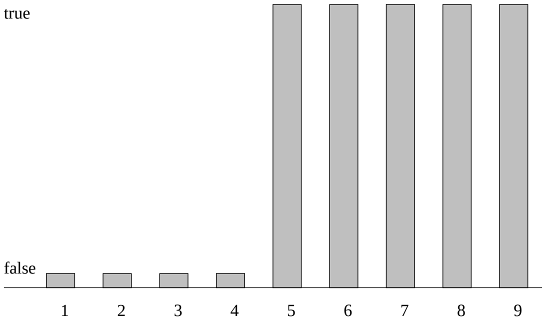

Figure 7: $RAF1$ decreases with probability 95%

The second property evaluates the probability of accumulating a protein (assume $RAF1$) within 120 seconds, but only once $RAF1$ has reached a given level of (lower) concentration. The property is defined by :

$$P_{=?}[(true)\mathbf{U}^{\leq 120}(RAF1 > C)\{(RAF1 = C)\}] \tag{12}$$

This property expresses "What is the probability of reaching a state with a higher level of $RAF1$ from the state where the concentration level is C?" The result is given in Figure 8, with C ranging over $0 \ldots 9$.

Note that the first property concerns the probability of protein decrease, from a given level; the second property concerns the probability of exceeding a given level, within a given time. From the combined results of the two experiments, we conclude there is a low probability of accumulating $RAF1$, when the concentration is between levels 5 and 9.

4.3 Protein stability in steady state while varying rates

This type of property is particularly useful during model fitting, i.e. fitting the model to experimental data. As an example, consider evaluating the probability that $RAF1$ is stable at level 2 or level 3 (in the steady state), whilst varying the performance of the reaction $r1$ (the reaction which binds $RAF1$ and RKIP). We vary the parameter k_1 (which determines the rate of $r1$) over the interval $[0 \ldots 1]$. The stability property is expressed by:

$$S_{=?}[(RAF1 \geq 2) \wedge (RAF1 \leq 3)] \tag{13}$$

Consider also the probability that $RAF1$ is stable at levels 0 and 1; the formula for this is:

$$S_{=?}[(RAF1 \geq 0) \wedge (RAF1 \leq 1)] \tag{14}$$

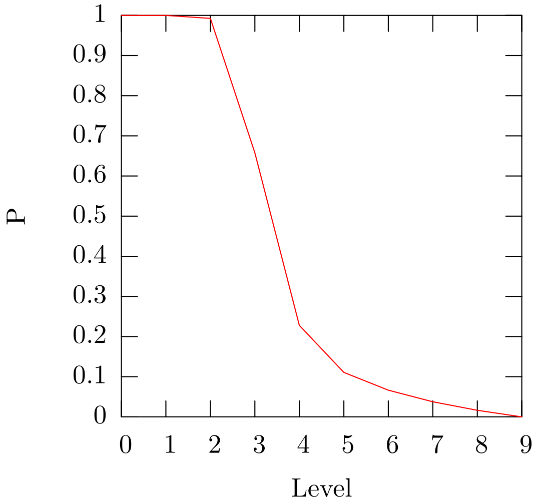

Figure 8: $RAF1$ increases from a given concentration level

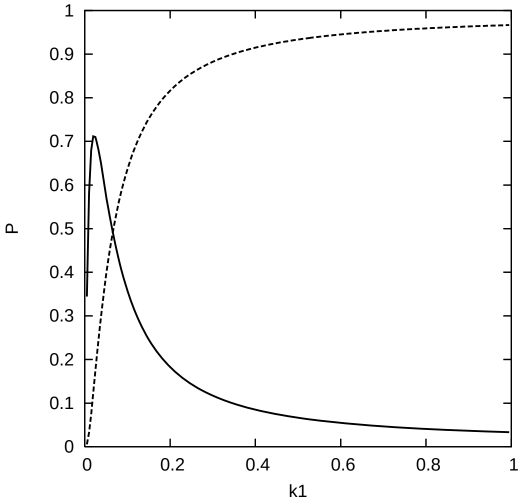

Figure 9: Stability of $RAF1$ at levels {2,3} and {0,1}

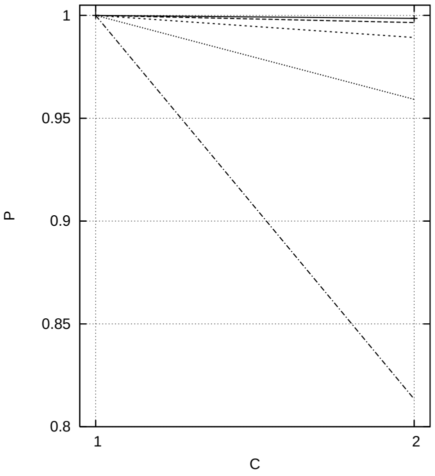

Figure 10: Activation sequence

Figure 9 gives results for both these properties, when $N = 5$. The likelihood of property (13) (solid line) peaks at $k1 = 0.03$ and then decreases; the likelihood of property (14) (dashed line) increases dramatically, becoming very likely when $k1 > 0.4$.

4.4 Activation sequence analysis

The last example illustrates queries over several proteins: sequences of protein activations. Consider two proteins: $RAF1/RKIP$ and $RAF1/RKIP/ERK-PP$. Is it possible that the (concentration of the) former “peaks” before the latter? Let M be the peak level.

The formula for this property is:

$$P_{=?}[(RAF1/RKIP/ERK - PP < M)\mathbf{U}(RAF1/RKIP = C)] \qquad (15)$$

This property expresses “What is the probability that the concentration of $RAF1/RKIP/ERK-PP$ is less than level M, until $RAF1/RKIP$ reaches concentration level C?” The results of this query, for C ranging over $0, 1, 2$ and M ranging over $1 \ldots 5$ are given in Figure 10: the line with steepest slope represents $M = 1$, the line which is nearly horizontal is $M = 5$. For example, the probability $RAF1/RKIP$ reaches concentration level 2 before $RAF1/RKIP/ERK-PP$ reaches concentration level 5 is more than 99%, the probability $RAF1/RKIP$ reaches concentration level 2 before $RAF1/RKIP/ERK-PP$ reaches concentration level 2 is almost 96%.

To confirm these results, we conducted the inverse experiment – check if it is possible for $RAF1/RKIP/ERK-PP$ to reach concentration level 5 *before*

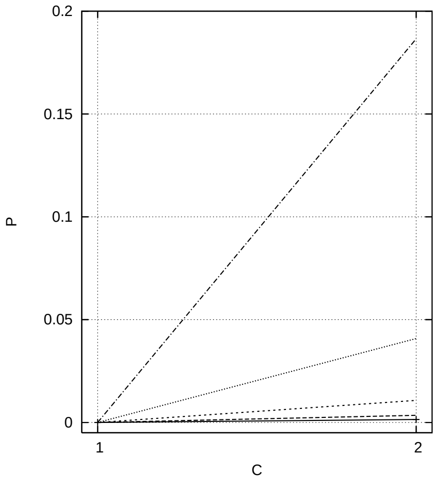

Figure 11: Inverse activation sequence

RAF1/RKIP reaches concentration level 2. The property is:

$$P_{=?}[(RAF1/RKIP < C)\mathbf{U}(RAF1/RKIP/ERK - PP = M)] \qquad (16)$$

This property expresses "What is the probability that the concentration of $RAF1/RKIP$ is less than level C until $RAF1/RKIP/ERK - PP$ reaches concentration level M?" The results are given in Figure 11 which is symmetric to Figure 10: for example, the probability $RAF1/RKIP/ERK - PP$ reaches concentration level 5 before $RAF1/RKIP$ reaches concentration level 2 is less than 0.14%.

This concludes our analysis of temporal queries, we now consider using our stochastic model for simulations, and relating those simulations to (deterministic) ODE simulations.

5 Comparison with ODE simulations

While our primary motivation is analysis with respect to temporal logic properties, it is interesting to consider simulation as well. Our stochastic models permit simulation, in PRISM, using the concept of state rewards [Pri]. For comparison, we also implemented a standard deterministic model, given by a set of ODEs, in the MATLAB® toolset. The ODEs are given in Appendix A.2.

To compare simulation results between the two types of model (i.e. stochastic and deterministic), consider the concentration of phosphorylated MEK, $MEK - PP$, over the time interval $[0 \ldots 100]$. Concentration is the vertical axis. Figure 12 plots the results, using the ODE model and two instances of our stochastic model, with $N = 3$ and $N = 7$. The "upper" curve is the ODE simulation, the "lower" curve is the stochastic simulation, when $N = 3$; the curve

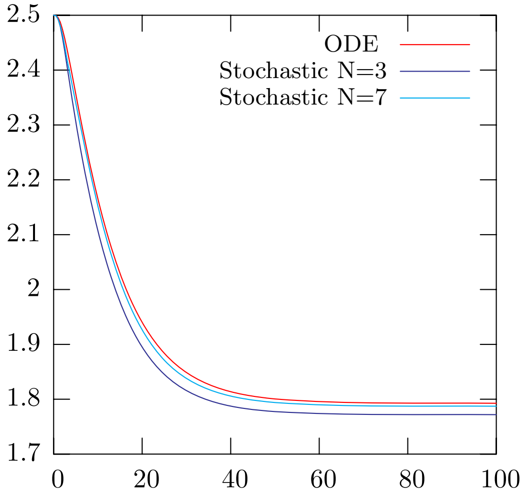

Figure 12: Comparison ODE and Stochastic models: MEK-PP simulation

in between the two is the stochastic behaviour when $N = 7$. As N increases, the closer the plots; with $N = 7$ the difference is barely discernable.

We make the comparison more precise by defining the distance between the stochastic and deterministic models:

$$\Delta = \sum_{i=1}^{x} \int_0^y (m_i(t) - \tilde{m}_{i,N}(t))^2 \cdot dt$$

where x is the number of proteins, $0 \ldots N$ are concentrations levels in the stochastic model, m_i is the concentration of i^{th} protein in the deterministic model, and $\tilde{m}_{i,N}$ is the concentration of the i^{th} protein in the stochastic model. $[0 \ldots y]$ is the time interval for the comparison. As N increases, the stochastic and deterministic models converge, namely

$$\lim_{N \to \infty} \Delta = 0.$$

Convergence is surpisingly quick. For example Table 2 gives the number of cumulative error metrics over 200 data points, in the time interval [0..100], of the protein $RAF1/RKIP$ (which exhibits the maximum error in this pathway).

The metrics are:

- Maximal absolute error of simulation ϵ_a,
- Maximal relative error of simulation ϵ_r,
- Cumulative absolute error of simulation $C\epsilon_a$,

N	ϵ_a	ϵ_r	$C\epsilon_a$	$C\epsilon_a^2$
3	0.126 mM	0.28	21.557 mM	2.58
4	0.103 mM	0.217	17.569 mM	1.727
5	0.086 mM	0.176	14.582 mM	1.191
7	0.061 mM	0.122	10.402 mM	0.605
11	0.036 mM	0.071	6.042 mM	0.204

Table 2: Error measurements

- Cumulative square error of simulation $C\epsilon_a^2$.

We conclude that in this network, $N = 7$ is quite sufficient to make the two models indistinguishable, for all practical purposes. This was a surprising and very useful result, since computation with small N is tractable on a single processor. This means that for the example network, our stochastic approach offers a new, practical analysis and simulation technique.

6 Discussion

A number of interesting (generic) temporal biological properties were proposed in [CF03], but we have not repeated that analysis here. Rather, we have concentrated on further properties which are specific to signalling networks and population based models. Mainly, we have found steady-state analysis most useful, but we have also illustrated the use of transient properties.

PRISM has been a useful tool for model checking, experimentation, and even simulation. All computations have been tractable on a single standard processor (the times are trivial and have been omitted).

We have assumed that the duration of a reaction is exponentially distributed. We choose the negative exponential because this is the only "memoryless" distribution. Our underlying assumption is that reactions are independent of history, that is they depend only on the current concentration of each reagent. Mass action kinetics are based on a similar assumption. It would be interesting to consider whether other distributions have a physical interpretation, and if so, to investigate how they relate to experimental and statistical results.

In Section 3.4 we showed that our PRISM model relates to mass action kinetics. While simulation is not the primary goal of our approach, in Section 5 we demonstrated that with small N, our model provides (more than) sufficient simulation accuracy, for the example system. This is because the example pathway has reactions which are all on a similar scale. If we were to apply our approach to a pathway where the changes of concentrations are on different scales, i.e. the corresponding ODE model is a set of stiff equations, then we could still reason about the stochastic model using temporal logic queries. However, simulations would not be as accurate, for small N. If more accuracy of simulation was required, then we would have to either increase N, or encode a more sophisticated solver within the PRISM representation (at the expense of transparency).

7 Related Work

The standard models of functional dynamics are ordinary differential equations (ODEs) for population dynamics [Voi00, CSK$^+$03], or stochastic simulations for individual dynamics [Gil77].

The recent work of Regev et al on π-calculus models [RSS01, PRSS01] has been deeply influential. In this work, a correspondence is made between molecules and processes. Here we have proposed a more abstract correspondence between species (i.e. concentrations) and processes. Whereas the emphasis of Regev et al is on simulation, we have concentrated on temporal properties expressed in CSL. More closely related work is presented in [CGH04] where the stochastic process algebra PEPA is used to model the same example pathway. The main advantage is that using the algebra, different formulations of the model can be compared (by bisimulation). One formulation relates clearly to the approach here (proteins as processes) whereas another permits abstraction over sub-pathways. Throughput analysis is main form of qualitative reasoning, though it is possible to "translate" the algebraic models into PRISM and then model check. The algebraic models cannot be used directly for simulation.

Petri nets provide an alternative to Markov chains [PWM03, KH04], with time, hybrid and stochastic extensions [PZHK04, MDNM00, GP98]. However, there are no appropriate model checkers for quantitative analysis (e.g. for stochastic Petri nets), or there are difficulties encoding our nonlinear dynamics (e.g. in time Petri nets), thus we cannot directly compare approaches.

The BIOCHAM workbench [CF03, CRCD$^+$04] provides an interface to the symbolic model checker NuSMV; the interface is based on a simple language for representing biochemical networks. The workbench provides mechanisms to reason about reachability, existence of partially described stable states, and some types of temporal behaviour. However, quantitative model checking is not supported, only qualitative queries can be verified.

8 Conclusions

We have described a new quantitative modelling and analysis approach for signal transduction networks. We model the dynamics of networks by continuous time Markov chains, making discrete approximations to protein molar concentrations. We describe the models in a high level language, using the PRISM modelling language: proteins are synchronous processes and concentrations are discrete, abstract quantities. Throughout, we have illustrated our approach with an example, the RKIP inhibited ERK pathway [CSK$^+$03].

The main advantage of our approach is that using a (continuous time) stochastic logic and the PRISM model checker, we can perform quantitative analysis such as *what is the probability that a protein concentration reaches a certain level and remains at that level thereafter?* and *how does varying a reaction rate affect that probability?* The approach offers considerably more expressive power than simulation or qualitative analysis. We can also perform standard simulations and we have compared our results with traditional ordinary differential equation-based (simulation) methods, as implemented in MATLAB®. An interesting and useful result is that in the example pathway, only a small number of discrete data values is required to render the simulations practically indistinguishable. Future work will include the addition of spatial dimensions (e.g. scaffolds) to our models.

Acknowledgements

This research is supported by the project *A Software Tool for Simulation and Analysis of Biochemical Networks*, funded by the DTI Beacon Bioscience Programme.

A Models

A.1 PRISM

The PRISM model is defined by the following; the system description is omitted - it simply runs all modules concurrently.

```
stochastic

const int N = 7;
const double R = 2.5/N;

module RAF1
RAF1: [0..N] init N;

[r1] (RAF1 > 0) -> RAF1*R: (RAF1’ = RAF1 - 1);
[r2] (RAF1 < N) -> 1: (RAF1’ = RAF1 + 1);
[r5] (RAF1 < N) -> 1: (RAF1’ = RAF1 + 1);
endmodule

module RKIP
RKIP: [0..N] init N;

[r1] (RKIP > 0) -> RKIP*R: (RKIP’ = RKIP - 1);
[r2] (RKIP < N) -> 1: (RKIP’ = RKIP + 1);
[r11] (RKIP < N) -> 1: (RKIP’ = RKIP + 1);
endmodule
```

```
module RAF1/RKIP
RAF1/RKIP: [0..N] init 0;

[r1] (RAF1/RKIP < N) -> 1: (RAF1/RKIP' = RAF1/RKIP + 1);
[r2] (RAF1/RKIP > 0) -> RAF1/RKIP*R:
              (RAF1/RKIP' = RAF1/RKIP - 1);
[r3] (RAF1/RKIP > 0) -> RAF1/RKIP*R:
              (RAF1/RKIP' = RAF1/RKIP - 1);
[r4] (RAF1/RKIP < N) -> 1: (RAF1/RKIP' = RAF1/RKIP + 1);
endmodule

module ERK-PP
ERK-PP: [0..N] init N;

[r3] (ERK-PP > 0) -> ERK-PP*R: (ERK-PP' = ERK-PP - 1);
[r4] (ERK-PP < N) -> 1: (ERK-PP' = ERK-PP + 1);
[r8] (ERK-PP < N) -> 1: (ERK-PP' = ERK-PP + 1);
endmodule

module RAF1/RKIP/ERK-PP
RAF1/RKIP/ERK-PP: [0..N] init 0;

[r3] (RAF1/RKIP/ERK-PP < N) -> 1:
(RAF1/RKIP/ERK-PP' = RAF1/RKIP/ERK-PP + 1);
[r4] (RAF1/RKIP/ERK-PP > 0) ->
RAF1/RKIP/ERK-PP*R:
(RAF1/RKIP/ERK-PP' = RAF1/RKIP/ERK-PP - 1);
[r5] (RAF1/RKIP/ERK-PP > 0) ->
RAF1/RKIP/ERK-PP*R:
(RAF1/RKIP/ERK-PP' = RAF1/RKIP/ERK-PP - 1);
endmodule

module ERK
ERK: [0..N] init 0;

[r5] (ERK < N) -> 1: (ERK' = ERK + 1);
[r6] (ERK > 0) -> ERK*R: (ERK' = ERK - 1);
[r7] (ERK < N) -> 1: (ERK' = ERK + 1);
endmodule

module RKIP-P
RKIP-P: [0..N] init 0;

[r5] (RKIP-P < N) -> 1: (RKIP-P' =RKIP-P + 1);
[r9] (RKIP-P > 0) -> RKIP-P*R: (RKIP-P' =RKIP-P - 1);
[r10] (RKIP-P < N) -> 1: (RKIP-P' =RKIP-P + 1);
```

```
endmodule

module RP
RP: [0..N] init N;

[r9] (RP > 0) -> RP*R: (RP’ = RP - 1);
[r10] (RP < N) -> 1: (RP’ = RP + 1);
[r11] (RP < N) -> 1: (RP’ = RP + 1);
endmodule

module MEK-PP
MEK-PP: [0..N] init N;

[r6] (MEK-PP > 0) ->  MEK-PP*R: (MEK-PP’ = MEK-PP - 1);
[r7] (MEK-PP < N) -> 1: (MEK-PP’ = MEK-PP + 1);
[r8] (MEK-PP < N) -> 1: (MEK-PP’ = MEK-PP + 1);
endmodule

module  MEK-PP/ERK
MEKPPERK: [0..N] init 0;

[r6] (MEK-PP/ERK < N) -> 1: (MEK-PP/ERK’ =  MEK-PP/ERK + 1);
[r7] (MEK-PP/ERK > 0) ->  MEK-PP/ERK*R:
           (MEK-PP/ERK’ =  MEK-PP/ERK - 1);
[r8] (MEK-PP/ERK > 0) ->  MEK-PP/ERK*R:
           (MEK-PP/ERK’ =  MEK-PP/ERK - 1);
endmodule

module RKIP-P/RP
 RKIP-P/RP: [0..N] init 0;

[r9] (RKIP-P/RP < N) -> 1: (RKIP-P/RP’ = RKIP-P/RP + 1);
[r10] (RKIP-P/RP > 0) -> RKIP-P/RP*R:
           (RKIP-P/RP’ = RKIP-P/RP - 1);
[r11] (RKIP-P/RP > 0) -> RKIP-P/RP*R:
           (RKIP-P/RP’ = RKIP-P/RP - 1);
endmodule

module Constants
x: bool init true;

[r1] (x) -> 0.53/R: (x’ = true);
[r2] (x) -> 0.0072/R: (x’ = true);
[r3] (x) -> 0.625/R: (x’ = true);
[r4] (x) -> 0.00245/R: (x’ = true);
[r5] (x) -> 0.0315/R: (x’ = true);
```

```
[r6] (x) -> 0.8/R: (x' = true);
[r7] (x) -> 0.0075/R: (x' = true);
[r8] (x) -> 0.071/R: (x' = true);
[r9] (x) -> 0.92/R: (x' = true);
[r10] (x) -> 0.00122/R: (x' = true);
[r11] (x) -> 0.87/R: (x' = true);

endmodule
```

A.2 ODE model

The ODE based model, given by the following reactions, is implemented in MATLAB® toolset. The kinetics are taken from [CSK+03].

1. RAF1 + RKIP → RAF1/RKIP
   $\frac{dv}{dt} = 0.53 \cdot RAF1 \cdot RKIP$
2. RAF1/RKIP → RAF1 + RKIP
   $\frac{dv}{dt} = 0.0072 \cdot RAF1/RKIP$
3. RAF1/RKIP + ERK-PP → RAF1/RKIP/ERK-PP
   $\frac{dv}{dt} = 0.625 \cdot RAF1/RKIP \cdot ERK - PP$
4. RAF1/RKIP/ERK-PP → RAF1/RKIP + ERK-PP
   $\frac{dv}{dt} = 0.00245 \cdot RAF1/RKIP/ERK - PP$
5. RAF1/RKIP/ERK-PP → ERK +RKIP-P + RAF1
   $\frac{dv}{dt} = 0.0315 \cdot RAF1/RKIP/ERK - PP$
6. MEK-PP + ERK → MEK-PP/ERK
   $\frac{dv}{dt} = 0.8 \cdot MEK - PP \cdot ERK$
7. MEK-PP/ERK → MEK-PP + ERK
   $\frac{dv}{dt} = 0.0075 \cdot MEK - PP/ERK$
8. MEK-PP/ERK → MEK-PP + ERK-PP
   $\frac{dv}{dt} = 0.071 \cdot MEK - PP/ERK$
9. RKIP-P + RP →RKIP-P/RP
   $\frac{dv}{dt} = 0.92 \cdot RKIP - P \cdot RP$
10. RKIP-P/RP →RKIP-P + RP
    $\frac{dv}{dt} = 0.00122 \cdot RKIP - P/RP$
11. RKIP-P/RP → RKIP + RP
    $\frac{dv}{dt} = 0.87 \cdot RKIP - P/RP$

The initial concentrations are: $RAF1 = RKIP = ERK - PP = MEK - PP = RP = 2.5$ Molar; all other proteins are 0.

References

[ASSB00] A. Aziz, K. Sanwal, V. Singhal, and R. Brayton. Model checking continuous time markov chains. *ACM Transactions on Computational Logic*, 1:162–170, 2000.

[BHHK00] C. Baier, B. Haverkort, H. Hermanns, and J.-P. Katoen. Model checking continuous-time Markov chains by transient analysis. *In CAV 2000*, 2000.

[CF03] Nathalie Chabrier and François Fages. Symbolic model checking of biochemical networks. *Lecture Notes in Computer Science*, 2602:149–162, 2003.

[CGH04] M. Calder, S. Gilmore, and J. Hillston. Modelling the influence of RKIP on the ERK signalling pathway using the stochastic process algebra PEPA. *In Proceedings of Bio-Concur 2004*, 2004.

[CRCD$^+$04] Nathalie Chabrier-Rivier, Marc Chiaverini, Vincent Danos, François Fages, and Vincent Schächter. Modeling and querying biomolecular interaction networks. *Theoretical Computer Science*, 325(1):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. *Lecture Notes in Computer Science*, 2602:127–141, 2003.

[Gil77] D. Gillespie. Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, 81(25):2340 –2361, 1977.

[GP98] Peter J.E. Goss and Jean Peccoud. Quantitative modeling of stochastic systems in molecular biology by using stochastic Petri nets. *Proc. Natl. Acad. Sci. USA (Biochemistry)*, 95:6750 – 6755, 1998.

[KDMH99] B. N. Kholodenko, O. V. Demin, G. Moehren, and J. B. Hoek. Quantification of short term signaling by the epidermal growth factor receptor. *The Journal of Biological Chemistry*, 274(42):30169–30181, October 1999.

[KH04] I. Koch and M. Heiner. Qualitative modelling and analysis of biochemical pathways with petri nets. *Tutorial Notes, 5th Int. Conference on Systems Biology - ICSB 2004*, Heidelberg/Germany, October 2004.

[KNP02] M. Kwiatkowska, G. Norman, and D. Parker. PRISM: Probabilistic Symbolic Model Checker. *Lecture Notes in Computer Science*, 2324:200–204, 2002.

[MDNM00] H. Matsuno, A. Doi, M. Nagasaki, and S. Miyano. Hybrid petri net representation of gene regulatory network. *Pacific Symposium on Biocomputing*, 5:341–352, 2000.

[PNK04] D. Parker, G. Norman, and M. Kwiatkowska. *PRISM 2.1 Users' Guide*. The University of Birmingham, September 2004.

[Pri] PRISM Web page. http://www.cs.bham.ac.uk/∼dxp/prism/.

[PRSS01] C. Priami, A. Regev, W. Silverman, and E. Shapiro. Application of a stochastic name passing calculus to representation and simulation of molecular processes. *Information Processing Letters*, 80:25–31, 2001.

[PWM03] J.W. Pinney, D.R. Westhead, and G.A. McConkey. Petri Net representations in systems biology. *Biochem. Soc. Trans.*, 31:1513 – 1515, 2003.

[PZHK04] L. Popova-Zeugmann, M. Heiner, and I. Koch. Modelling and analysis of biochemical networks with time petri nets. *Informatik-Berichte der HUB Nr. 170*, 1(170):136 – 143, 2004.

[RSS01] A. Regev, W. Silverman, and E. Shapiro. Representation and simulation of biochemical processes using π-calculus process algebra. *Pacific Symposium on Biocomputing 2001 (PSB 2001)*, pages 459–470, 2001.

[SEJGM02] B. Schoeberl, C. Eichler-Jonsson, E. D. Gilles, and G. Muller. Computational modelling of the dynamics of the map kinase cascade activated by surface and internalised egf receptors. *Nature Biotechnology*, 20:370–375, April 2002.

[Voi00] E. O. Voit. *Computational Analysis of Biochemical Systems*. Cambridge University Press, 2000.