Available online at www.sciencedirect.com

ScienceDirect

Theoretical Computer Science 391 (2008) 239–257

Theoretical Computer Science

www.elsevier.com/locate/tcs

Probabilistic model checking of complex biological pathways

John Heath[a], Marta Kwiatkowska[b,*], Gethin Norman[b], David Parker[b], Oksana Tymchyshyn[c]

[a] School of Biosciences, University of Birmingham, Edgbaston, Birmingham B15 2TT, United Kingdom
[b] Oxford University Computing Laboratory, Wolfson Building, Parks Road, Oxford OX1 3QD, United Kingdom
[c] School of Computer Science, University of Birmingham, Edgbaston, Birmingham B15 2TT, United Kingdom

Abstract

Probabilistic model checking is a formal verification technique that has been successfully applied to the analysis of systems from a broad range of domains, including security and communication protocols, distributed algorithms and power management. In this paper we illustrate its applicability to a complex biological system: the FGF (Fibroblast Growth Factor) signalling pathway. We give a detailed description of how this case study can be modelled in the probabilistic model checker PRISM, discussing some of the issues that arise in doing so, and show how we can thus examine a rich selection of quantitative properties of this model. We present experimental results for the case study under several different scenarios and provide a detailed analysis, illustrating how this approach can be used to yield a better understanding of the dynamics of the pathway. Finally, we outline a number of exact and approximate techniques to enable the verification of larger and more complex pathways and apply several of them to the FGF case study.

Keywords: Probabilistic model checking; Probabilistic verification; Biological pathways; Biological modelling

1. Introduction

There has been considerable success recently in adapting approaches from computer science to the analysis of biological systems and, in particular, biochemical pathways. The majority of this work has relied on simulation-based techniques developed for discrete-stochastic models [1]. These allow modelling of the evolution of individual molecules, whose rates of interaction are controlled by exponential distributions. The principal alternative modelling paradigm, using ordinary differential equations, differs in that it reasons about how the average concentrations of the molecules evolve over time. In this paper, as in [2,3], we adopt the stochastic modelling approach, but employ methods which allow calculation of *exact* quantitative measures of the model under study.

We use probabilistic model checking [4] and the probabilistic model checker PRISM [5,6] as a framework for the modelling and analysis of biological pathways. This approach is motivated by the success of the previous work

* Corresponding author. Tel.: +44 1865 283509; fax: +44 1865 273839.
E-mail address: marta.kwiatkowska@comlab.ox.ac.uk (M. Kwiatkowska).

0304-3975/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2007.11.013

which has demonstrated the applicability of these techniques to the analysis of a wide variety of complex systems [7]. One benefit of this is the ability to employ the existing efficient implementations and tool support developed in this area. Additionally, we enjoy the advantages of model checking, for example, the use of both a formal model and specification of the system under study, as well as the fact that the approach is exhaustive, that is, all possible behaviours of the system are analysed. Our intention is that the methods in this paper should be used in conjunction with the classical simulation and differential-equation-based approaches to provide greater insight into the complex interactions of biological pathways. This paper provides a detailed illustration of the applicability of probabilistic model checking to this domain through the analysis of a complex biological pathway called FGF (Fibroblast Growth Factor).

One of the main problems facing model checking in practice is the well-known state-space-explosion problem whereas the complexity of the system under study increases, there is an exponential growth in the state space of the underlying model. A similar phenomenon is found when modelling and analysing biological networks and pathways: as the complexity of the pathway increases, the number of molecular species grows exponentially. In this paper we also outline some of the approaches that have been used to overcome this problem both in the fields of computer science and biology. In addition, to demonstrate how these methods can be used in practice, we apply a number of these techniques to the FGF pathway.

Related Work. The closest approach to that presented here is in [2], where the probabilistic model checker PRISM is used to model the RKIP inhibited ERK pathway. The main difference is that in [2] the authors consider a “population”-based approach to modelling using approximate techniques where concentrations are modelled by discrete abstract quantities. In addition, here we demonstrate how a larger class of temporal properties, including reward-based measures, are applicable to the study of biological systems. Also related to the RKIP inhibited ERK pathway is [3], where it is demonstrated how the stochastic process algebra PEPA [8] can be used to model biological pathways. The stochastic π-calculus [9] has been proposed as a model language for biological systems [10,11]; this approach has so far been used in conjunction with stochastic simulation, for example through the tools BioSpi [11] and SPiM [12].

In parallel with the development of the PRISM model of the FGF pathway presented in this paper, we have constructed a separate π-calculus model [13,14] and applied stochastic simulation through BioSpi. Although currently these models focus on different aspects of the pathway, in the future we aim to use this complex case study as a basis for investigating the advantages of stochastic simulation and probabilistic model checking.

Outline of the paper. In the next section we give an overview of probabilistic model checking and PRISM. Section 3 uses a simple example to demonstrate how PRISM can be used to model and analyse biological systems. In Section 4 we introduce the FGF case study, while Sections 5 and 6 describe the modelling and specification of the case study in PRISM. Section 7 presents and discusses the results obtained with PRISM. In Section 8 we summarise a number of approaches to reduce the size of the model under study and in Section 9 demonstrate these techniques by considering variants of the FGF pathway. Section 10 concludes the paper.

A preliminary version of this paper appeared as [15].

2. Probabilistic model checking and PRISM

Probabilistic model checking is a formal verification technique for the modelling and analysis of systems which exhibit stochastic behaviour. This technique is a variant of *model checking*, a well-established and widely used formal method for ascertaining the correctness of real-life systems. Model checking requires two inputs: a description of the system in some high-level modelling formalism (such as a Petri net or process algebra), and specification of one or more desired properties of that system in temporal logic (e.g. CTL or LTL). From these, one can construct a model of the system, typically a labelled state-transition system in which each state represents a possible configuration and the transitions represent the evolution of the system from one configuration to another over time. It is then possible to automatically verify whether or not each property is satisfied, based on a systematic and exhaustive exploration of the model.

In probabilistic model checking, the models are augmented with quantitative information regarding, the likelihood that transitions occur and, the times at which they do so. In practice, these models are typically Markov chains or Markov decision processes. In this paper, it suffices to consider *continuous-time Markov chains* (CTMCs), in which transitions between states are assigned (positive, real-valued) rates, which are interpreted as the rates of negative exponential distributions.

Formally, letting AP be a fixed, finite set of atomic propositions used to label states with properties of interest, a CTMC is a tuple $(S, \mathbf{R}, L)$ where:

- S is a finite set of *states*;
- $\mathbf{R} : S \times S \to \mathbb{R}_{\geq 0}$ is a *transition rate matrix*;
- $L : S \to 2^{AP}$ is a *labelling* function which assigns to each state the set of atomic propositions that are valid in the state.

The transition rate matrix $\mathbf{R}$ assigns rates to each pair of states, which are used as parameters of the exponential distribution. A transition can only occur between states s and s' if $\mathbf{R}(s, s') > 0$ and, in this case, the probability of this transition being triggered within t time units equals $1 - \mathrm{e}^{-\mathbf{R}(s,s')\cdot t}$. Typically, in a state s, there is more than one state s' for which $\mathbf{R}(s, s') > 0$; this is known as a *race condition* and the first transition to be triggered determines the next state. The time spent in state s before any such transition occurs is exponentially distributed with rate $E(s) = \sum_{s' \in S} \mathbf{R}(s, s')$, while the probability of moving to state s' is given by $\frac{\mathbf{R}(s,s')}{E(s)}$.

A CTMC can be augmented with two types of rewards: those associated with states are cumulated in proportion to the time spent in the state, and those associated with transitions are cumulated each time the transition is taken. Formally, a reward structure for a CTMC is a pair $(\underline{\rho}, \boldsymbol{\iota})$ where:

- the *state reward function* $\underline{\rho} : S \to \mathbb{R}_{\geq 0}$ assigns the rate (per time unit) at which the reward is acquired while in a state;
- the *transition reward function* $\boldsymbol{\iota} : S \times S \to \mathbb{R}_{\geq 0}$ assigns the reward acquired each time a transition occurs.

The properties used in probabilistic model checking, while still expressed in temporal logic, are now quantitative in nature. For example, rather than verifying that "the protein always eventually degrades", we may ask "what is the probability that the protein eventually degrades?" or "what is the probability that the protein degrades within T hours?". Reward-based properties include "what is the expected time the proteins are bound within the first T time units?" and "what is the expected number of complexation reactions before relocation occurs?". For further details on probabilistic model checking of CTMCs see [16].

PRISM [5,6] is a probabilistic checking tool developed at the University of Birmingham. Models are specified in a simple-state-based language based on Reactive Modules. An extension of the temporal logic CSL [17,18] is used to specify properties of CTMC models augmented with rewards. The tool employs state-of-the-art symbolic approaches using data structures based on binary decision diagrams [19]. Also of interest, the tool includes support for PEPA [8] and has recently been extended to allow for simulation-based analysis using Monte Carlo methods and discrete event simulation. For further details, see [6].

3. Modelling a simple biological system in PRISM

We now illustrate PRISM's modelling and specification languages through an example: the simple set of biological reactions given in Fig. 1(a). We consider two proteins A and B which can undergo complexation with rate r_1 and decomplexation with rate r_2. In addition, A can degrade with rate r_3.

We give two alternative approaches for modelling these reactions in PRISM, shown in Fig. 1(b) and 1(c), respectively. A model described in the PRISM language comprises a keyword, corresponding to the model type (to describe a CTMC the keyword `ctmc` is used), and a set of *modules*, the state of each being represented by a set of finite-ranging *variables*. In approach 1 (Fig. 1(b)) we use a single module with one variable, representing the (three) possible states of the whole system (which are listed in the italicised comments in the figure). The behaviour of this module, i.e. the changes in states which it can undergo, is specified by a number of *guarded commands* of the form $[]\ g \to r : u$, with the interpretation that if the predicate (guard) g is true, then the system is updated according to u (where $x' = \ldots$ denotes how the value of variable x is changed). The rate at which this occurs is r, i.e. this is the value that will be attached to the corresponding transition in the underlying CTMC.

In approach 2 (Fig. 1(c)) we represent the different possible forms that the proteins can take (A, B and $A{:}B$) as separate modules, each with a single variable taking value 0 or 1, representing its absence or presence, respectively. To model interactions where the state of several modules changes simultaneously, we use *synchronisation*, denoted by attaching action labels to guarded commands (placed inside the square brackets). For example, when the *bind* action occurs, variables a and b in modules A and B change from 1 to 0 and variable ab in module AB changes from 0

Reactions:

1. $A{+}B \longleftrightarrow A{:}B$ (complexation)
2. $A \longrightarrow$ (degradation)

Reaction rates:

- complexation : r_1
- decomplexation : r_2
- degradation : r_3

(a) System of reactions

```
module M

  ab : [0..2] init 1;
  // 0: A degraded and B free 1
  // 1: A and B free (initial state)
  // 2: A and B bound

  [] ab=1 → r1 : (ab'=2); // bind
  [] ab=2 → r2 : (ab'=1); // unbind
  [] ab=1 → r3 : (ab'=0); // degrade

endmodule
```

(b) PRISM encoding 1

```
module A

  a : [0..1] init 1;

  [bind]  a=1 → r1 : (a'=0);
  [rel]   a=0 → r2 : (a'=1);
  []      a=1 → r3 : (a'=0);

endmodule
```

```
module B

  b : [0..1] init 1;

  [bind]  b=1 → (b'=0);
  [rel]   b=0 → (b'=1);

endmodule
```

```
module AB

  ab : [0..1] init 0;

  [bind]  ab=0 → (ab'=1);
  [rel]   ab=1 → (ab'=0);

endmodule
```

(c) PRISM encoding 2

```
rewards "rew1"

  a=1 : 1;

endrewards
```

(d) Reward structure 1

```
rewards "rew2"

  [bind]  true  :  1;

endrewards
```

(e) Reward structure 2

Fig. 1. A simple example and possible PRISM representations.

to 1. In this example, the rate of each combined transition is fully specified in module A and we have omitted the rates from the other modules. More precisely, PRISM assigns a rate of 1 to any command for which none is specified and computes the rate of a combined transition as the product of the rates for each command. Note that independent transitions, involving only a single module, can also be included, as shown by the modelling of degradation (which only involves A), by omitting the action label.

In general, a combination of the above two modelling approaches is used. In simple cases it is possible to use a single variable, but as the system becomes more complex the use of separate variables and synchronisation becomes more desirable. We will see this later in the paper.

Properties of CTMCs are specified in PRISM using an extension of the temporal logic CSL [17,18]. We now give a number of examples for the model in Fig. 1(c).

- $\mathcal{P}_{=?}[\ \mathtt{true}\ \mathcal{U}^{[T,T]}\ ab{=}1\ ]$ — the probability that the protein A is bound to the protein B at time instant T.
- $\mathcal{P}_{=?}[\ ab{=}0\ \mathcal{U}\ (a{=}0 \wedge ab{=}0)\ ]$ — the probability that the protein A degrades before binding to the protein B.
- $\mathcal{R}_{=?}[\ \mathcal{C}^{\leq T}\ ]$ — during the first T time units, the expected time that the protein A is free. The reward structure for this property associates reward 1 with states where the variable a equals 1 (see the reward structure "*rew1*" in Fig. 1(d)).
- $\mathcal{R}_{=?}[\ \mathcal{F}\ (a{=}0 \wedge ab{=}0)\ ]$ — the expected number of times that the proteins A and B bind before A degrades. In this case, a reward of 1 is associated with any transition labelled by *bind* and all other rewards are set to 0 (see the reward structure "*rew2*" in Fig. 1(e)).

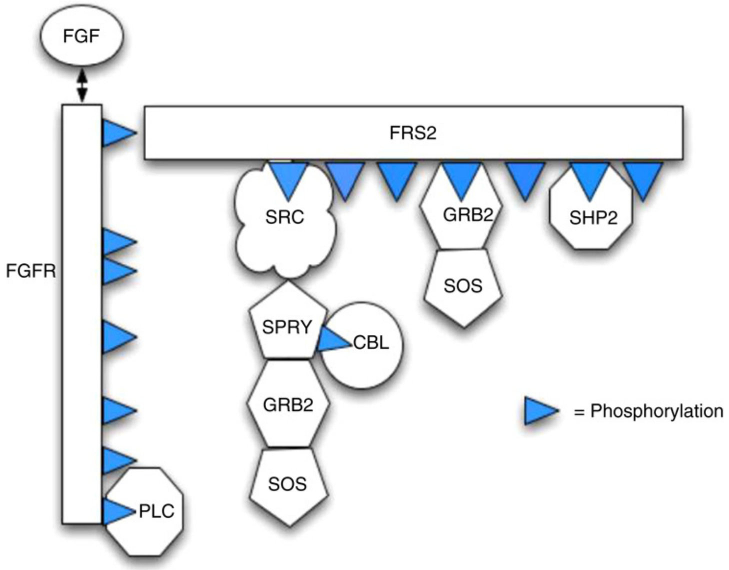


Fig. 2. Diagram showing the different possible bindings in the pathway.

4. Case study: FGF

Cell signalling is a set of communication mechanisms that allows cells to sense and respond to their environment. Signalling systems are composed of networks of independent molecular components that undergo state transitions (i.e. covalent modification) and bind to other molecules in order to transfer a regulatory signal. Covalent modification, such as phosphorylation, induces conformational changes within the molecule to alter its activity, binding properties or subcellular localization. Protein molecules often contain multiple distinct residues that can undergo independent state transitions. For further details on cell signalling see, for example, [20,21].

Fibroblast Growth Factors (FGF) are a family of proteins which play a key role in the process of cell signalling in a variety of contexts, for example wound healing and skeletal development. The mechanisms of the FGF signalling pathway are complex and not yet fully understood, see [22,23]. In this section, we present a model of the pathway which is based on literature-derived information regarding the early stages of FGF signal propagation and which incorporates several features that have been reported to negatively regulate this propagation [24,25].

Our model describes state transitions of pathway molecules due to association/disassociation with partner molecules, phosphorylation and dephosphorylation, relocation and degradation. Fig. 2 illustrates the different components in the pathway and their possible bindings. Below is a list of the reactions included in the model. Further details are provided in Fig. 3.

1. An FGF ligand binds to an FGF receptor (FGFR).

2. The existence of this FGF:FGFR complex leads to phosphorylation of FGFR on two residues Y653 and Y654 in the activation loop of the receptor.

3. The phosphorylated form of the receptors Y653 and Y654 leads to phosphorylation of other FGFR receptor residues: Y463, Y583, Y585, Y766 (in this model we only consider Y766 further).

4. and 5. The phosphorylated form of the receptors Y653 and Y654 also leads to phosphorylation of the FGFR substrate FRS2, which binds to both the phosphorylated and dephosphorylated forms of the FGFR.

6. FRS2 can be dephosphorylated by Shp2 bound to FRS2.

7. A number of effector proteins interact with the phosphorylated form of FRS2. In this model we include Src, Grb2:Sos and Shp2.

8. and 9. These are two methods of attenuating signal propagation by removal (i.e. relocation) of components. In reaction **8**, if Src is associated with the phosphorylated FRS2 Y219, this leads to relocation (i.e. endocytosis and/or degradation of FGFR:FRS2). In reaction **9**, Plc binds to the phosphorylated receptor Y766 of FGFR, which leads to relocation/degradation of FGFR.

1. FGF binds to FGFR
FGF+FGFR ↔ FGFR:FGF (k_{on} = 5e+8M^{-1}s^{-1}, k_{off}=1e−1s^{-1})
2. Whilst FGFR:FGF exists
FGFR Y653 → FGFR Y653P (k_{cat}=0.1s^{-1})
Y766PFGFR Y654 → FGFR Y654P (k_{cat}=0.1s^{-1})
3. When FGFR Y653P and FGFR Y654P
FGFR Y463 → FGFR Y463P (k_{cat}=70s^{-1})
FGFR Y583 → FGFR Y583P (k_{cat}=70s^{-1})
FGFR Y585 → FGFR Y585P (k_{cat}=70s^{-1})
FGFR Y766 → FGFR Y766P (k_{cat}=70s^{-1})
4. FGFR binds FRS2
FGFR+ FRS2 ↔ FGFR:FRS2 (k_{on} = 1e+6M^{-1}s^{-1}, k_{off}=2e−2s^{-1})
5. When FGFR Y653P, FGFR Y654P and FGFR:FRS2
FRS2 Y196 → FRS2 Y196P (k_{cat}=0.2s^{-1})
FRS2 Y290 → FRS2 Y290P (k_{cat}=0.2s^{-1})
FRS2 Y306 → FRS2 Y306P (k_{cat}=0.2s^{-1})
FRS2 Y382 → FRS2 Y382P (k_{cat}=0.2s^{-1})
FRS2 Y392 → FRS2 Y392P (k_{cat}=0.2s^{-1})
FRS2 Y436 → FRS2 Y436P (k_{cat}=0.2s^{-1})
FRS2 Y471 → FRS2 Y471P (k_{cat}=0.2s^{-1})
6. Reverse when Shp2 bound to FRS2:
FRS2 Y196P → FRS2 Y196 (k_{cat}=12s^{-1})
FRS2 Y290P → FRS2 Y290 (k_{cat}=12s^{-1})
FRS2 Y306P → FRS2 Y306 (k_{cat}=12s^{-1})
FRS2 Y382P → FRS2 Y382 (k_{cat}=12s^{-1})
FRS2 Y392P → FRS2 Y392 (k_{cat}=12s^{-1})
FRS2 Y436P → FRS2 Y436 (k_{cat}=12s^{-1})
FRS2 Y471P → FRS2 Y471 (k_{cat}=12s^{-1})
7. FRS2 effectors bind phosphoFRS2:
Src+FRS2 Y196P ↔ Src:FRS2 Y2196P (k_{on} = 1e+6M^{-1}s^{-1}, k_{off}=2e−2s^{-1})
Grb2+FRS2 Y306P ↔ Grb2:FRS2 Y306P (k_{on} = 1e+6M^{-1}s^{-1}, k_{off}=2e−2s^{-1})
Shp2+FRS2 Y471P ↔ Shp2:FRS2 Y471P (k_{on} = 1e+6M^{-1}s^{-1}, k_{off}=2e−2s^{-1})
8. When Src:FRS2 we relocate/remove
Src:FRS2 → relocate out ($t_{1/2}$=15min)
9. When Plc:FGFR it degrades FGFR
Plc+FGFR Y766P ↔ Plc:FGFR Y766P (k_{on} = 1e+6M^{-1}s^{-1}, k_{off}=2e−2s^{-1})
Plc:FGFR Y766P → degFGFR ($t_{1/2}$=60min)
10. Spry appears in time-dependent manner:
→ Spry ($t_{1/2}$=15min)
11. Spry binds Src and is phosphorylated:
Spry+Src ↔ Spry Y55:Src (k_{on} = 1e+5M^{-1}s^{-1}, k_{off}=1e−4s^{-1})
Spry Y55:Src → Spry Y55P:Src (k_{cat}=10s^{-1})
Spry Y55P+Src ↔ Spry Y55P:Src (k_{on} = 1e+5M^{-1}s^{-1}, k_{off}=1e−4s^{-1})
Spry Y55P+Cbl ↔ Spry Y55P:Cbl (k_{on} = 1e+5M^{-1}s^{-1}, k_{off}=1e−4s^{-1})
Spry Y55P+Grb2 ↔ Spry Y55P:Grb2 (k_{on} = 1e+5M^{-1}s^{-1}, k_{off}=1e−4s^{-1})
12. phosphoSpry binds Cbl which degrades/removes FRS2
Spry Y55P:Cbl+FRS2 ↔ FRS2-Ubi (k_{cat}=8.5e−4s^{-1})
FRS2-Ubi → degFrs2 ($t_{1/2}$=5min)
13. Spry is dephosphorylated when Shp2 bound to FRS2:
Spry Y55P → Spry Y55 (k_{cat}=12s^{-1})
14. Grb2 binds Sos
Grb2+Sos ↔ Grb2:Sos (k_{on} = 1e+5M^{-1}s^{-1}, k_{off}=1e−4s^{-1})

Fig. 3. Reaction rules for the pathway.

10. The signal attenuator Spry is a known inhibitor of FGFR signalling and is synthesised in response to FGFR signalling. Here we include a variable to regulate the concentration of Spry protein in a time-dependent manner.

11. We incorporate the association of Spry with Src and concomitant phosphorylation of Spry residue Y55.

12. The Y55 phosphorylated form of Spry binds with Cbl, which leads to ubiquitin modification of FRS2 and a degradation of FRS2 through ubiquitin-mediated proteolysis.

13. The Y55P form of Spry is dephosphorylated by Shp2 bound to FRS2 Y247P.

14. Grb2 binds to the Y55P form of Spry. In our model Spry competes with FRS2 for Grb2 as has been suggested by some studies in the literature.

Note that this model is not intended to, and cannot be, a fully accurate representation of a real-world FGF signalling pathway. Its primary purpose at this stage of development is as a tool to evaluate biological hypotheses that are not easily obtained by intuition or manual methods. To this end, the model is an abstraction as argued in [26], created to facilitate predictive "in silico" experiments for a range of scenarios. Results of such "in silico genetics" experiments based on simulations of a stochastic π-calculus model of the above set of reactions are described in [13,14].

We have selected the reactions according to their current biological interest rather than complete understanding of the components of FGF signalling. Indeed, at this stage we have ignored many reactions that could prove to be significant in the regulation of FGFR signalling in real cells. However, the design permits the incorporation of further modifications to the core model as biological understanding advances. The model is idealised in that it does not take into account variations in composition, affinities or rate constants that might occur in different cell types or physiological conditions. However, a useful computational modelling approach should accommodate future quantitative or qualitative modifications to the core model.

5. Modelling in PRISM

We now describe the specification in PRISM of the FGF model from the previous section. We employ a combination of the two approaches discussed in Section 3. Each of the basic elements of the pathway, including all possible compounds and receptor residues (FGF, FGFR, FRS2, Plc, Src, Spry, Sos, Grb2, Cbl and Shp2), is represented by a separate PRISM module. Synchronisation between modules is used to model reactions involving interactions of multiple elements. However, the different forms that each can take (for example, which other compounds it is bound to) are represented by one or more variables within the module.

Note that, in the PRISM model of the pathway, we have only included the phosphorylation sites of FGFR and FRS2 that influence downstream reactions. More precisely, we have omitted phosphorylation sites Y463, Y583 and Y585 of FGFR and phosphorylation sites Y290, Y382, Y436 and Y392 of FRS2.

Our model represents a single instance of the pathway, i.e. there can be at most one element of each compound. This has the advantage that the resulting state space is relatively small (80,616 states); however, the model is highly complex due to the large number of different interactions that can occur in the pathway (there are over 560,000 transitions between states). Furthermore, as will be demonstrated later in the paper, the model is sufficiently rich to explain the roles of the components in the pathway and how they interact. The study of a single instance of the pathway is also motivated by the fact that the same signal dynamics (Fig. 7(a)) were obtained in [13,14] for a model where the number of molecules of each type was initially set to 100. Fragments of the PRISM code for the modules representing FRS2, Src and Sos are given in Figs. 4–6, respectively. The full version is available from the PRISM web page [6].

Fig. 4 shows the module for FRS2. It contains variables representing whether FRS2 is currently:

- undergoing ubiquitin modification (*FrsUbi*);
- relocated (*relocFrs2*);
- degraded (*degFrs2*);
- bound to other compounds (*FrsFgfr*, *FrsGrb*, *FrsShp* and *FrsSrc*).

In addition, the module has variables representing the phosphorylation status of each of FRS's receptors, namely *Y196P*, *Y306P* and *Y471P* (recall that we have included only the receptors of FRS2 that influence downstream reactions).

The first set of commands given in Fig. 4 corresponds to the phosphorylation of receptors in FRS2 (reaction **5** in Fig. 3). Since the only variables that are updated are local to this module, the commands have no action label, i.e. we do not require any other module to synchronise on these commands. The guards of these commands incorporate dependencies on the current state both of FRS2 itself and of other compounds. More precisely, FGFR must be bound to FRS2 and certain receptors of FGFR must have already been phosphorylated.

Elsewhere, in Fig. 4, we see commands that use synchronisation to model interactions with other compounds, e.g. the release of Src (the commands labelled *src_rel*) and the binding and release of Sos (the commands labelled

```
formula Frs = relocFrs2=0 ∧ degFrs2=0; // FRS2 not relocated or degraded

module FRS2

  FrsUbi    : [0..1] init 0; // ubiquitin modification of FRS2
  relocFrs2 : [0..1] init 0; // FRS2 relocated
  degFrs2   : [0..1] init 0; // FRS2 degraded
  Y196P     : [0..1] init 0; // receptor Y196 phosphorylated
  Y306P     : [0..1] init 0; // receptor Y306 phosphorylated
  Y471P     : [0..1] init 0; // receptor Y471 phosphorylated
  // compounds bound to FRS2
  FrsFgfr   : [0..1] init 0; // 0: FGFR not bound, 1: FGFR bound
  FrsGrb    : [0..2] init 0; // 0: Grb2 not bound, 1: Grb2 bound, 2: Grb2:Sos bound
  FrsShp    : [0..1] init 0; // 0: Shp2 not bound, 1: Shp2 bound
  FrsSrc    : [0..8] init 0; // 0: Src not bound, 1: Src bound, 2: Src:Spry, 3: Src:SpryP,
                             // 4: Src:SpryP:Cbl, 5: Src:SpryP:Grb, 6: Src:SpryP:Grb:Cbl
                             // 7: Src:SpryP:Grb:Sos, 8: Src:SpryP:Grb:Sos:Cbl

  // FGFR+FRS2 ↔ FGFR:FRS2 (4)
  [fgfr_bind] Frs∧FrsFgfr=0 → 10 : (FrsFgfr'=1);
  [fgfr_rel]  Frs∧FrsFgfr=1 → 0.001 : (FrsFgfr'=0);

  // phosphorylation of receptors (5)
  [] Frs∧Y653P=1∧Y654P=1∧FrsFgfr=1∧Y196P=0 → 0.2 : (Y196P′=1); // Y196
  [] Frs∧Y653P=1∧Y654P=1∧FrsFgfr=1∧Y306P=0 → 0.2 : (Y306P′=1); // Y306
  [] Frs∧Y653P=1∧Y654P=1∧FrsFgfr=1∧Y471P=0 → 0.2 : (Y471P′=1); // Y471

  // dephosphorylation of Y196 (6) - remove Src if bound
  []          Frs∧FrsShp=1∧Y196P=1∧FrsSrc=0 → 12 : (Y196P′=0);
  [src_rel]   Frs∧FrsShp=1∧Y196P=1∧FrsSrc>0 → 12 : (Y196P′=0)∧(FrsSrc′=0);

  // dephosphorylation of Y306 (6) - remove Grb2 if bound
  []          Frs∧FrsShp=1∧Y306P=1∧FrsGrb=0 → 12 : (Y306P′=0);
  [grb_rel]   Frs∧FrsShp=1∧Y306P=1∧FrsGrb>0 → 12 : (Y306P′=0)∧(FrsGrb′=0);

  // dephosphorylation of Y471 (6) - remove Shp2 since bound
  [shp_rel] Frs∧FrsShp=1∧Y471P=1 → 12 : (Y471P′=0)∧(FrsShp′=0);
                                  ...
  // Src:FRS2→degFRS2 (8)
  [] Frs∧FrsSrc>0 → 1/(15*60) : (relocFrs2′=1);
                                  ...
  // Spry55p:Cbl+FRS2→Frs-Ubi (12)
  [] Frs∧FrsSrc=4,6,8 ∧ FrsUbi=0 → 0.00085 : (FrsUbi′=1);
  // FRS2-Ubi→degFRS2 (12)
  [] Frs∧FrsUbi=1 → 1/(5*60) : (degFrs2′=1);
                                  ...
  // Grb2+Sos↔Grb2:Sos (14)
  [sos_bind_frs] Frs∧FrsGrb=1   → 1      : (FrsGrb′=2);            // Grb:FRS2
  [sos_bind_frs] Frs∧FrsSrc=5,6 → 1      : (FrsSrc′=FrsSrc+2);     // Grb:SpryP:Src:FRS2
  [sos_rel_frs]  Frs∧FrsGrb=2   → 0.0001 : (FrsGrb′=1);            // Grb:FRS2
  [sos_rel_frs]  Frs∧FrsSrc=7,8 → 0.0001 : (FrsSrc′=FrsSrc−2);     // Grb:SpryP:Src:FRS2
                                  ...

endmodule
```

Fig. 4. Fragment of the PRISM module for FRS2 and related compounds.

sos_bind_frs and *sos_rel_frs*). Note the corresponding commands in modules *SRC* (Fig. 5) and *SOS* (Fig. 6). In each of these cases, as discussed in Section 3, the rate of the combined interaction is specified in the *FRS2* module and is hence omitted from the corresponding commands in *SRC* and *SOS*. Also, in the module for Sos (Fig. 6), there are different action labels for the binding and release of Sos with Grb2; this is because Grb2 can be either free or bound to a number of different compounds when it interacts with Sos. For example, Grb2 can be bound to Frs2

```
module SRC

  Src : [0..8] init 1;
  // 0: Src bound to FRS2,
  // 1: Src
  // 2: Src:Spry
  // 3: Src:SpryP
  // 4: Src:SpryP:Cbl
  // 5: Src:SpryP:Grb
  // 6: Src:SpryP:Grb:Cbl
  // 7: Src:SpryP:Grb:Sos
  // 8: Src:SpryP:Grb:Sos:Cbl

  // Src+FRS2196P↔Src:FRS2 (7)
  [src_bind]  Src>0 → (Src′=0);
  [src_rel]   Src=0 → (Src′=FrsSrc);

  // Spry+Src→Spry55:Src or Spry55P+Src→Spry55P:Src (11)
  [spry_bind] Src=1 → 1 : (Src′=Spry+1);

  // Spry+Src←Spry55:Src (11)
  [spry_rel] Src=2 → 0.01 : (Src′=1);

  // Spry55P+Src←Spry55P:Src (11)
  [spry_rel] Src>2 → 0.0001 : (Src′=1);

  // Spry55:Src→Spry55P:Src (11)
  [] Src=2 → 10 : (Src′=3);

  // SpryP+Cbl↔SpryP:Cbl (11)
  [cbl_bind_src]  Src=3,5,7 → 1       : (Src′=Src+1);
  [cbl_rel_src]   Src=4,6,8 → 0.0001: (Src′=Src−1);

  // SpryP+Grb↔SpryP:Grb (11)
  [grb_bind_src]  Src=3,4 → 1       : (Src′=Src+2*Grb);
  [grb_rel_src]   Src=5,6 → 0.0001 : (Src′=Src−2); // SOS not bound
  [grb_rel_src]   Src=7,8 → 0.0001 : (Src′=Src−4); // SOS bound

  // Grb2+Sos↔Grb2:Sos (14)
  [sos_bind_src]  Src=5,6 → 1       : (Src′=Src+2);
  [sos_rel_src]   Src=7,8 → 0.0001 : (Src′=Src-2);

endmodule
```

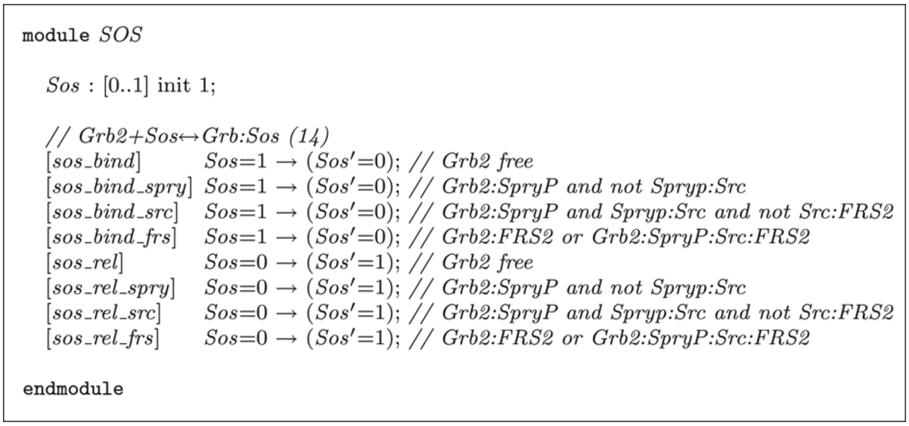
Fig. 5. PRISM module for Src and related compounds.

(through reaction **7**) or Spry (through reaction **11**), and Spry can in turn be bound to Src, which can also be bound to FRS2.

Notice how, in the commands for binding and unbinding of Src with FRS2 in Fig. 4 (labelled *sos_bind_frs* and *sos_rel_frs*), we can use the value of *FrsSrc* to update the value of *Src*, rather than having an individual command for each separate case. Also worthy of note are the updates to *Src* in Fig. 5 when either Grb2 or Grb2:Sos bind to Src. To simplify the code, we have used a single command for each of these possible reactions, and therefore updates which either increment or decrement the variable *Src* by 2 or 4 (the variable *Grb* takes value 1 if Grb2 is not bound to Sos and value 2 if Sos is bound).

6. Property specification

Our primary goal in this case study is to analyse the various mechanisms previously reported to negatively regulate signalling. Since the binding of Grb2 to FRS2 serves as the primary link between FGFR activation and ERK signalling, we examine the amount of Grb2 bound to FRS2 as the system evolves. In addition, we investigate the different



```
module SOS

   Sos : [0..1] init 1;

   // Grb2+Sos↔Grb:Sos (14)
   [sos_bind]        Sos=1 → (Sos'=0); // Grb2 free
   [sos_bind_spry]   Sos=1 → (Sos'=0); // Grb2:SpryP and not Spryp:Src
   [sos_bind_src]    Sos=1 → (Sos'=0); // Grb2:SpryP and Spryp:Src and not Src:FRS2
   [sos_bind_frs]    Sos=1 → (Sos'=0); // Grb2:FRS2 or Grb2:SpryP:Src:FRS2
   [sos_rel]         Sos=0 → (Sos'=1); // Grb2 free
   [sos_rel_spry]    Sos=0 → (Sos'=1); // Grb2:SpryP and not Spryp:Src
   [sos_rel_src]     Sos=0 → (Sos'=1); // Grb2:SpryP and Spryp:Src and not Src:FRS2
   [sos_rel_frs]     Sos=0 → (Sos'=1); // Grb2:FRS2 or Grb2:SpryP:Src:FRS2

endmodule
```

Fig. 6. PRISM module for Sos.

causes of degradation which, based on the system description, can result from the occurrence of one of the following reactions:

- when Src:FRS2 is present, FRS2 is relocated (reaction **8**);
- when Plc:FGFR is present, it degrades FGFR (reaction **9**);
- when phosphoSpry binds to Cbl, it degrades FRS2 (reaction **12**).

Below, we present a list of various properties of the model that we have analysed, and the form in which they are supplied to the PRISM tool. For the latter, we define a number of *atomic propositions*, essentially predicates over the variables in the PRISM model, which can be used to identify states of the model that have certain properties of interest. These include $a_{grb2:frs2}$, which indicates that Grb2 is bound to FRS2 (i.e. those states where the variable *FrsGrb* of Fig. 4 is greater than zero), and a_{src_reloc}, a_{plc_deg} and a_{spry_deg}, which identify those states reached after the corresponding causes of degradation/relocation given above have occurred. For properties using expected rewards (with the $\mathcal{R}_{=?}[\cdot]$ operator), we also explain the reward structure used.

A. $\mathcal{P}_{=?}[\ \mathtt{true}\ \mathcal{U}^{[t,t]}\ a_{grb2:frs2}\ ]$ — the probability that Grb2 is bound to FRS2 at the time instant t.

B. $\mathcal{R}_{=?}[\ \mathcal{C}^{\leq t}\ ]$ — the expected number of times that Grb2 binds to FRS2 by time t. In this case, the only non-zero rewards are associated with transitions involving Grb2 binding to FRS2 which have a reward 1.

C. $\mathcal{R}_{=?}[\ \mathcal{C}^{\leq t}\ ]$ — the expected time that Grb2 spends bound to FRS2 within the first T time units. The reward structure for this property assigns a reward of 1 to all states where Grb2 is bound to FRS2 (i.e. all states satisfying $a_{grb2:frs2}$) and 0 to all other states and transitions.

D. $\mathcal{S}_{=?}[\ a_{grb2:frs2}\ ]$ — the long-run probability that Grb2 is bound to FRS2.

E. $\mathcal{R}_{=?}[\ \mathcal{F}\ (a_{src\_reloc} \vee a_{plc\_deg} \vee a_{spry\_deg})\ ]$ — the expected number of times Grb2 binds to FRS2 before degradation or relocation occurs. As for property **B**, transitions involving Grb2 binding to FRS2 are assigned reward 1.

F. $\mathcal{R}_{=?}[\ \mathcal{F}\ (a_{src\_reloc} \vee a_{plc\_deg} \vee a_{spry\_deg})\ ]$ — the expected time Grb2 spends bound to FRS2 before degradation or relocation occurs. As for property **C**, all states where Grb2 is bound to FRS2 have a reward of 1.

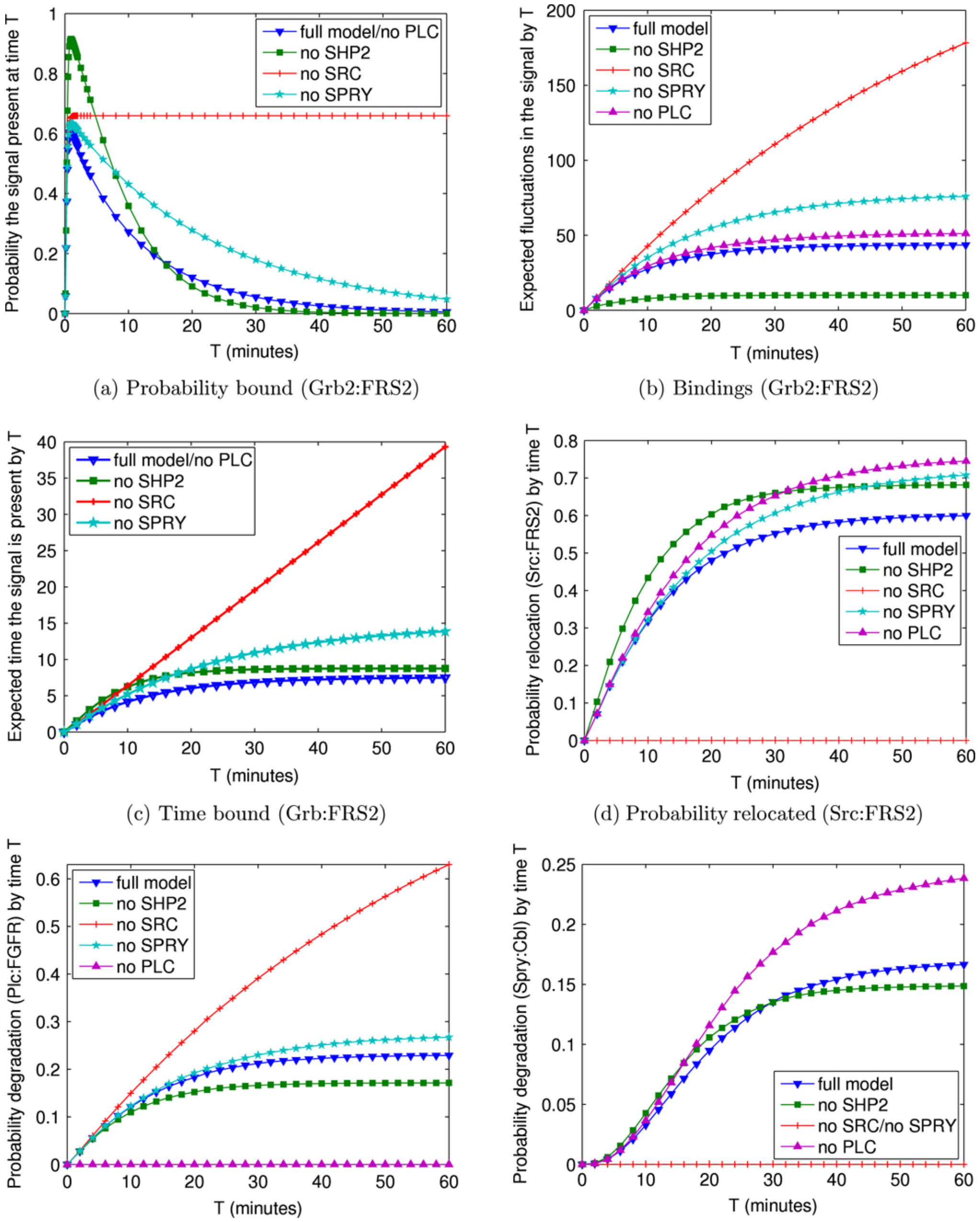
G. $\mathcal{P}_{=?}[\ \neg(a_{src\_reloc} \vee a_{plc\_deg} \vee a_{spry\_deg})\ \mathcal{U}^{[0,t]}\ a\ ]$ for a equal to a_{src_reloc}, a_{plc_deg} and a_{spry_deg} — the probability that degradation or relocation occurs by time t and Src, Plc or Spry is the cause.

H. $\mathcal{P}_{=?}[\ \neg(a_{src\_reloc} \vee a_{plc\_deg} \vee a_{spry\_deg})\ \mathcal{U}\ a\ ]$ for a equal to a_{src_reloc}, a_{plc_deg} and a_{spry_deg} — the probability that Src, Plc or Spry is the first cause of degradation or relocation.

I. $\mathcal{R}_{=?}[\ \mathcal{F}\ (a_{src\_reloc} \vee a_{plc\_deg} \vee a_{spry\_deg})\ ]$ — the expected time until degradation or relocation occurs in the pathway. For this property all states are assigned reward 1 (and all transitions are assigned reward 0).

7. Results and analysis

We used PRISM to construct the FGF model described in Section 5 and analyse the set of properties listed in Section 6. This was done for a range of different scenarios. First, we developed a base model, representing the



(a) Probability bound (Grb2:FRS2)

(b) Bindings (Grb2:FRS2)

(c) Time bound (Grb:FRS2)

(d) Probability relocated (Src:FRS2)

(e) Probability degraded (Plc:Fgfr)

(f) Probability degraded (Spry:Cbl)

Fig. 7. Transient numerical results

complete system, in which we suppose that initially FGF, unbound and unphosphorylated FGFR, unphosphorylated FRS2, unbound Src, Grb2, Cbl, Plc and Sos are all present in the system (Spry arrives within the system with the half-time of 10 min).

Subsequently, we performed a series of "in silico genetics" experiments on the model designed to investigate the roles of the various components of the activated receptor complex in controlling signalling dynamics. This involves deriving a series of modified models of the pathway, where certain components are omitted (Shp2, Src, Spry or Plc), which is easily achieved in a PRISM model by just changing the initial value of the component under study. For example, to remove Src from the system we just need to change the initial value of the variable *Src* from 1 to 0 (see Fig. 5).

Table 1
Long run and expected reachability properties for the signal

	Probability bound	Expected no. of bindings	Expected time bound (min)
Full model	7.54e−7	43.1027	6.27042
No Shp2	3.29e−9	10.0510	7.78927
No Src	0.659460	283.233	39.6102
No Spry	4.6e−6	78.3314	10.8791
No Plc	0.0	51.5475	7.56241

Table 2
Probability and expected time until degradation/relocation in the long run

	Probability of degradation/relocation			Expected time (min)
	Src:FRS2	Plc:FGFR	Spry:Cbl	
Full model	0.602356	0.229107	0.168536	14.0258
No Shp2	0.679102	0.176693	0.149742	10.5418
No Src	–	1.0	0.0	60.3719
No Spry	0.724590	0.275410	–	16.8096
No Plc	0.756113	–	0.243887	17.5277

For each property we include the statistics for 5 cases: for the complete pathway and for the pathway when either Shp2, Src, Spry or Plc is removed. Fig. 7(a)–(c) show the transient behaviour (i.e. at each time instant T) of the signal (binding of Grb2 to FRS2) for the first 60 minutes, namely properties **A**, **B** and **C** from the previous section. Table 1 gives the long-run behaviour of the signal, i.e. properties **D**, **E** and **F**. The latter three results can be regarded as the values of the first three in "the limit", i.e. as either T tends to infinity or degradation occurs. Fig. 7(d)–(f) show the transient probability of each of the possible causes of relocation or degradation occurring (property **G**). Table 2 shows the results relating to degradation in the long run (properties **H** and **I**).

We begin with an analysis of the signal (binding of Grb2 to FRS2) in the complete model, i.e. see the first plot ("full model") in Fig. 7 and the first lines of Tables 1 and 2. The results presented demonstrate that the probability of the signal being present (Fig. 7(a)) shows a rapid increase, reaching its maximum level in about 1 to 2 minutes. The peak is followed by a gradual decrease in the signal, which then levels off at a small non-zero value. In this time interval Grb2 repeatedly binds to FRS2 (Fig. 7(b)) and, as time passes, Grb2 spends a smaller proportion of time bound to FRS2 (Fig. 7(c)).

The rapid increase in the signal is due the relevant reactions (the binding of Grb2 to FRS2 triggered by phosphorylation of FRS2, which requires activated FGFR to first bind to FRS2) all occurring at very fast rates. On the other hand, the decline in the signal is caused either by dephosphorylation of FRS2 (due to Shp2 being bound to FRS2) or by relocation/degradation of FRS2. Dephosphorylation of FRS2 is both fast and allows Grb2 to rebind (as FRS2 can become phosphorylated again). The overall decline in the signal is due to relocation of FRS2 caused by bound Src which takes a relatively long time to occur (Table 2 and Fig. 7(d)). Degradation caused by Spry has little impact since it is not present from the start and, by the time it appears, it is more likely that Grb2 is no longer bound or Src has caused relocation (Table 2, Fig. 7(d) and (f)).

The fact that the signal levels out at a non-zero value (Table 1) is caused by Plc degrading the FGF receptor bound to FRS2 and Grb2. More precisely, after FGFR is degraded by Plc, no phosphorylation of partner FRS2 residues is possible. The signal stays non-zero since neither Src-mediated relocation and degradation, nor Shp-mediated dephosphorylation, are possible when respective FRS2 residues are not active. The non-zero value is very small because it is more likely that Src has caused relocation (Table 2). The repeated binding of Grb2 to FRS2 (Fig. 7(b)) is caused by the dephosphorylation of FRS2, which is soon phosphorylated again and allows Grb2 to rebind. The decrease in the proportion of time that Grb2 is bound to FRS2 is due to the probability of FRS2 becoming relocated/degraded increasing as time passes (Fig. 7(d)–(f)).

Next, we further illustrate the role of the components by analysing models in which different elements of the pathway are not present.

Shp2. Fig. 7(a) shows that the peak in the signal is significantly larger than that seen under normal conditions. By removing Shp2 we have removed, as explained above, the fast reaction for the release of Grb2 from FRS2, and this justifies the larger peak. The faster decline in the signal is due a greater chance of Src being bound (as Shp2 causes the dephosphorylation of FRS2, it also causes the release of Src from FRS2), and hence the increased chance of relocation (Fig. 7(d) and Table 2). These observations are also the cause for the decrease in the time until degradation/relocation when Shp2 is removed (Table 2) and the fact that the other causes of degradation/relocation are less likely (Fig. 7(e)–(f) and Table 2). Dephosphorylation due to bound Shp2 was responsible for the large number of times that Grb2 and FRS2 bind (and unbind) in the original model; we do not see such a large number of bindings once Shp2 is removed (Fig. 7(b) and Table 1).

Src. As Fig. 7(a) demonstrates, the suppression of Src is predicted to have a major impact on signalling dynamics: after a fast increase, the signal fails to decrease substantially. This is supported by the results presented in both Fig. 7(d)–(f) and Table 2 which show that Src is the main cause of signal degradation, and by removing Src the time until degradation or relocation greatly increases. The failure of Spry to degrade the signal (Fig. 7(f) and Table 2) is attributed to its activation being downstream of Src. Note that this also means that Plc is the only remaining cause of degradation.

Spry. The model fails to reproduce the role of Spry in inhibiting the activation of the ERK pathway by competition for Grb2:Sos. More precisely, our results show that the suppression of Spry does not result in signal reduction. This can be explained by the differences in system designs: under laboratory conditions the action of Spry is measured after Spry is over expressed, whereas, under normal physiological conditions, Spry is known to arrive slowly within the system. Removing Spry removes one of the causes of degradation, and therefore increases the likelihood of the other causes of degradation/relocation (Fig. 7(d)–(e) and Table 2). Moreover, the increase in the probability of Plc causing degradation/relocation leads to an increase in the chance of Grb2 and FRS2 remaining bound (Table 2).

Plc. While having a modest effect on transient signal expression, the main action of Plc removal is to cause the signal to stabilise at zero (Table 1). This is due to Plc being the only cause of degradation/relocation not relating to FRS2. The increase in time until degradation (Table 2) is also attributed to the fact that, by removing Plc, we have eliminated one of the possible causes of degradation. This also has the effect that the other causes of relocation/degradation are more likely (Fig. 7(d), (f) and Table 2).

8. State-space-reduction techniques

One of the principal limitations of model checking techniques, regardless of the application domain, is the so-called *state-space-explosion problem*. This refers to the fact that the sizes of the models that need to be constructed and analysed (in this case CTMCs) have a tendency to become huge when considering real-life systems. In this section we outline a number of reduction techniques that can be used to reduce model sizes, thus speeding up the process of model checking and extending the range of models that can be analysed. In most cases, these approaches can be seen as *aggregation* techniques, where model states with either similar or identical behaviour are grouped together, resulting in a reduction in the state space.

We describe two classes of reduction techniques: *exact* approaches, for which the reduced model will yield identical model checking results to the original, and *approximate* approaches, which construct a simplified version of the model, producing only approximate results. The former can usually be automated but may be expensive to apply. The latter normally require input from the modeller and some understanding of the model being analysed, but offer potentially greater scope for state-space reduction. In Section 9, we will present some empirical results to illustrate the applicability of the various approaches to the FGF pathway studied in Sections 4–7.

8.1. Exact reduction techniques

The reduction techniques described in this section are based on the idea of grouping together states of the underlying CTMC whose behaviours are identical. This is formalised by the notion of *bisimulation* or *lumpability* [27]: the set of states of a CTMC is partitioned such that equivalent states, i.e. states within the same element of the partition, satisfy the same atomic propositions and the sum of the rates of transitions to states in any element of the partition is the same. Analysis can now be performed on the reduced (or *quotient*) CTMC in which each state is an element of the partition. The logic CSL, which we have used in this paper to express properties of CTMCs for model checking, can be shown to be preserved by bisimulation [18], i.e. the result of checking a property on the original model is guaranteed to be the same as checking it on the reduced model.

An efficient algorithm for the coarsest (smallest) possible computing bisimulation of a CTMC and constructing the corresponding reduced model was presented in [28]. The complexity of the algorithm is linear in the number of transitions and logarithmic in the number of states and it can hence be expensive for large models. Recent studies, however, have shown this approach to be of benefit for CSL model checking on a range of different models [29]. A new algorithm, based on the same efficient, *symbolic* data structures that are used in the PRISM model checker, was presented in [30]. An additional benefit of bisimulation reduction techniques is that they can be applied compositionally [31], i.e. first applied to individual components of a model and then the results combined.

An alternative way to generate smaller but identical CTMCs is to apply *symmetry reduction* techniques, which are a way of exploiting the presence of replication in a model. More specifically, we mention here *component symmetry*, where a model contains multiple copies of symmetric components which can be exchanged with no effect on the overall behaviour of the model. In similar fashion to the techniques described above, we can partition the states of a CTMC, grouping states which differ only by a permutation of symmetric components, and then construct a reduced CTMC with one state for each element of the partition. In fact, the resulting reduction is simply a particular instance of bisimulation (although typically not the coarsest bisimulation possible). In [32], an extension of PRISM is described which incorporates techniques to construct the symmetry-reduced CTMC.

Alternatively, we can view symmetry reduction as a *population-based* approach to modelling, where a model includes, rather than the actual local state of each individual symmetric component, counters of how many components there are in each possible state. Progress has been made towards automating such translations at the modelling language level, see e.g. [33], although this can be also be performed manually.

8.2. Approximate reduction techniques

We now summarise a variety of approximate techniques for state-space reduction, including both methods aimed specifically at biological systems and those developed in the more general context of CTMC analysis.

In [34,35], a number of general principles to aid the modelling and quantitative analysis of protein signalling networks are introduced and conditions are given for instances where their applications yield either exact or approximate dynamics. The approach is shown to be equally applicable to the discrete-stochastic (CTMC) and continuous-deterministic (ODE) modelling frameworks, and can lead to drastic reductions in the computational model (states or differential equations respectively) of the network under study. The types of biological networks to which these principles have been developed are those where receptors/proteins act as *scaffolds*, i.e. have multiple docking sites and engage multiple downstream signalling proteins. The reduction approach is based on identifying specific ‘micro-state’ components of the network, for example corresponding to individual protein domains/docking sites or receptors, analysing these components in isolation and then using the results of this analysis to construct an approximate ‘macro-description’ of the complete network. The advantage of employing this approach is that the ‘macro-description’ grows linearly as the number of components increases, whereas a ‘micro-state’ model of the whole network would grow exponentially. The approach is reliant on the fact that the different ‘micro-state’ components either act independently (yielding exact results) or can be assumed to act independently without a significant variation in the network’s evolution (yielding approximate results).

An alternative method is based on the observation that many biological pathways/networks are *stiff*: the rates of the reactions appearing in the model differ by many orders of magnitude. For such models, to reduce the complexity of the model under study one can first analyse, in isolation, the ‘fast’ reactions (i.e. reactions with very high rates) and ‘fast’ states (i.e. states in which ‘fast’ reactions are available), and use the results to construct a model, for the remaining ‘slow’ reactions and states, which serves as an approximate model for the complete pathway/network. More generally, one can divide the reactions and states into different classes based on their relative speed and perform the analysis hierarchically. Examples of approaches based on this observation include approximate methods for solving systems of ODEs [36,37], approximate stochastic simulation algorithms [38,39] and an approximate numerical algorithm to compute transient probabilities of CTMCs [40].

Stiff models also appear naturally in performance analysis, for example in reliability models where failures rates are normally orders of magnitude smaller (i.e. slower) than the corresponding repair rates. Techniques for the computation of performance measures of stiff CTMCs have therefore been an important topic of study in this domain. These are

Table 3
Symmetry reduction of the simplified pathway (states and transitions)

N	Unreduced		Symmetry reduction		
	States	Transitions	States	Transitions	Time (s)
1	22	82	22	82	0.001
2	477	3,441	252	1,725	0.048
3	10,222	111,634	2,002	19,672	0.938
4	216,961	3,189,289	12,397	156,181	4.877
5	4,568,094	84,614,826	63,756	967,002	13.32
6	95,525,293	2,137,868,041	283,360	4,973,017	44.74
7	1,985,831,398	521,56,810,042	1,118,260	22,102,144	99.08
8	41,071,054,113	1,239,158,792,433	3,996,135	87,224,361	184.0
9	845,602,665,094	28,831,879,987,426	13,123,110	311,698,630	411.8
10	17,340,217,864,861	659,570,078,773,441	40,060,020	1,023,602,701	949.3

again often based on first analysing the ‘fast’ states in isolation and then using the results to construct an approximate model of the complete system. For example, based on this methodology, [41] and [42] presented algorithms for the computation of transient and long-run measures of CTMCs respectively.

Another method for constructing approximate models of complex biological pathways/networks is given in [2], where the concentrations of molecules or molecular species are modelled as *discrete, abstract quantities*, rather than precise values. This approach is applied to the RKIP inhibited ERK pathway [43] and has been shown to give accurate results with only a small range of discrete quantities. The PRISM model considered in Sections 4–7, where there is only one molecule of each type, can be seen as a special case where there are only two abstract quantities for the concentrations (i.e. high and low).

The final reduction approach we mention is that of *abstraction* [44], where an abstract model is constructed from a CTMC through a partition of its state space. Like the exact reduction techniques mentioned earlier, each state of the abstract model corresponds to an element of the partition of the full state space. In contrast, though, the partition here must be specified explicitly — it is not derived by grouping states with ‘equivalent’ behaviour. Furthermore, the reduced model is an approximation of the original. The authors show how CSL model checking of the original concrete model can be performed on the approximate abstract model using a three valued logic with truth values: ‘true’, ‘false’ and ‘do not know’. This approach can be applied given any partition; however, to gain knowledge about the original system, i.e. obtain answers other than ‘do not know’, the partition needs to be based on an understanding of the components of the system and how they interact.

9. Applying reduction methods to the FGF pathway

We now illustrate the applicability of some of the reduction techniques described in Section 8 by applying them to the FGF pathway case study discussed in this paper.

9.1. *Exact reduction techniques*

We first apply the *exact* approaches (see Section 8.1) to a simplified version of the pathway as presented in [45]. In the simplified model, an FGF protein (molecule) can bind to an FGF receptor (FGFR) and, when FGF and FGFR are bound, two different residues on FGFR can become phosphorylated which, subsequently, allows the signal transducing proteins Src and Grb2, respectively, to bind to FGFR. Each of these binding and phosphorylation reactions is also reversible. Finally, when Src is bound to the FGFR complex, FGFR can be relocated, along with any components bound to it.

We first applied symmetry reduction to the model. Symmetry is present in the FGF pathway example since all of the molecules of a particular type are identical. Table 3 shows, as the number (N) of each type of molecule varies, the sizes of the CTMC (number of states and transitions) both before and after symmetry reduction has been applied. In fact, we constructed the reduced model using both techniques discussed in Section 8.1: firstly, using the automatic reduction of [32] and, secondly, with a ‘population-based’ version of the model, created manually in the PRISM modelling language. The time required for the former is also shown in the table. The two methods result in identical CTMCs. Although, in this case, both techniques can be used, their applicability can vary depending on the model.

More specifically, the method from [32] performs better for models with a relatively small number of (potentially complex) symmetric components, whereas the population-based method is better suited to models where there are a large number of simple components.

We also applied the symbolic algorithm of [30] to compute the coarsest possible bisimulation reduction for these models. Our first observation is that, in this case, these reductions offer only a very slight improvement over those obtained using symmetry given in Table 3. For example, in the case $N = 4$, we obtain 12,373 states instead of 12,397. Furthermore, the process is significantly slower: for $N = 4$ this requires over 2.5 hours compared to less than 5 seconds. This is to be expected since, unlike with symmetry reduction, the algorithm works at a low level, manipulating states and transitions of the CTMC, and cannot exploit any high-level information about the model. Bisimulation reduction algorithms are, however, more generally applicable. We also used the implementation of [30] on the full FGF model from Section 4 (which exhibits no symmetry since there is only one of each molecule) and were able to reduce the state space from 80,616 to 38,661.

9.2. Approximate reduction techniques

In this section, based on the analysis of the FGF pathway given in [46], we present three approaches for reducing the complexity of the complete model of the FGF pathway as given in Section 5.

Note that, as discussed in Section 5, we have already performed one simplification of the pathway, namely the removal of phosphorylation sites Y463, Y583 and Y585 of FGFR and phosphorylation sites Y290, Y382, Y436 and Y392 of FRS2. This was motivated by the fact that these sites have no influence over the signal, and therefore removing these sites does not affect measures relating to it. This step was vital in making the analysis of Section 7 tractable, decreasing the state space from 10,285,320 to 80,616 and the number of transitions from 92,767,336 to 560,520.

This model reduction can be seen as an application of the principles given in [34,35] since the sites we have removed are independent of the other sites in the pathway. Furthermore, by examining the reaction rules (see Fig. 3), it can be seen that the remaining phosphorylation sites are not independent. For example, the sites Y653 and Y654 of FGFR must become phosphorylated before any other sites can become phosphorylated; Grb2 can bind both at the sites Y196 and Y306 of FRS2; and Shp2 binding at the site Y471 of FRS2 can cause the remaining sites of FRS2 to dephosphorylate.

We have also applied three further approximation reductions to the model:

(1) *Removal of Sos from the pathway*. Since Sos is downstream of the signal (Grb2 binding to FRS2), and hence cannot influence it, we have chosen to remove Sos from the pathway. Note that, for measures relating to the signal, it follows that this reduction will lead to exact, not approximate, results. The reduction would, however, affect the results relating to components that Sos can influence.

(2) *Separation of fast and slow reactions*. Based on their rates, we classified the reactions corresponding to the binding of FGF, FGFR and FRS2 and the phosphorylation of FGFR Y653 and FGFR Y654 (see the reactions **1**, **2** and **4** of Fig. 3) as ‘fast’ and the remaining reactions as ‘slow’. As discussed in Section 8.2, for models in which reaction rates differ by orders of magnitude, approximate reduction approaches based on a decomposition of the system into ‘fast’ and ‘slow’ systems have been developed. For this study we take a simpler approach, reducing the complexity of the model by disabling all ‘slow’ reactions in states where a ‘fast’ reaction is available. Although this simple method might not lead to as significant reductions in model size as the methods discussed in Section 8.2, it only requires the high-level system description of the pathway to be modified. Moreover, standard model checking algorithms can be employed.

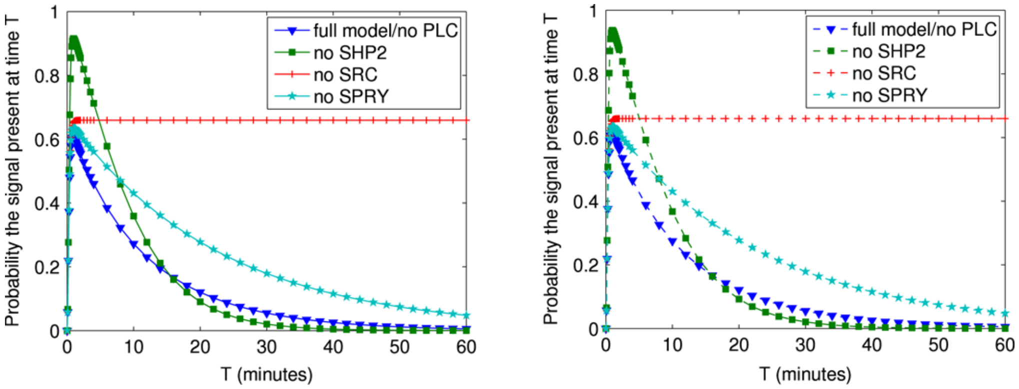
(3) *Limiting when the complexation and decomplexation of FRS2 and FGFR can occur*. In this reduction, we suppose that FRS2 can only bind to FGFR once both FGFR Y653 and FGFR Y654 have become phosphorylated and also that the decomplexation of FGFR:FRS2 cannot occur. The motivation behind the first restriction is that the complexation of FGFR and FRS2, and the phosphorylation of the sites Y653 and Y654 of FGFR, occur at much faster time scales than that of the subsequent downstream reactions, and therefore changes in this aspect of the pathway would only be noticeable over a very small time scale. Similarly, the removal of the decomplexation of FGFR:FRS2 would only be noticeable over a very small time scale: as the rate of FRS2 and FGFR complexation is extremely fast, following the decomplexation of FRS2 and FGFR one would see the (re)complexation of FGFR and FRS2 almost immediately.

Table 4
Model-based reductions of the complete pathway (model statistics)

	States	Transitions	Construction time (s)	Model checking time (s)
Complete model	80,616	560,520	0.352	66.55
Reduction (1)	23,940	149,916	0.186	24.37
Reduction (2)	43,424	168,930	0.309	28.41
Reduction (3)	40,528	275,574	0.287	41.30
Reductions (1) and (2)	12,881	46,090	0.169	8.17
Reductions (1) and (3)	12,080	73,717	0.155	11.16
Reductions (2) and (3)	35,160	151,896	0.306	26.05
Reductions (1), (2) and (3)	10,422	40,936	0.162	7.45

Table 5
Probability of degradation/relocation (complete and reduced pathways)

	Src:FRS2		Plc:FGFR		Spry:Cbl	
	Complete	Reduced	Complete	Reduced	Complete	Reduced
Full model	0.602356	0.604139	0.229107	0.229782	0.168536	0.166079
No Shp2	0.679102	0.681475	0.176693	0.171749	0.149742	0.146775
No Src	–	–	1.0	1.0	0.0	0.0
No Spry	0.724590	0.724590	0.275410	0.275410	–	–
No Plc	0.756113	0.758703	–	–	0.243887	0.241297



(a) Complete pathway

(b) Reduced model

Fig. 8. Probability Grb2 bound to FRS2 at time T (complete and reduced pathways).

Table 4 gives the model statistics for both the complete model and the model obtained after applying the reductions (1), (2) and (3), both in isolation and, collectively. The results show that reduction (1) – removal of Sos – yields the greatest decrease in state space among the three. This is because it is the only one that completely removes an element of the pathway. The decreases in state space for reductions (2) and (3) are very similar, since they involve placing a similar number of restrictions when certain reactions can occur. We also observe that an even more substantial gain is obtained by employing multiple reductions. In general, it is therefore advantageous to look into a number of different reduction approaches, although, as already stated, this does require some understanding of the model under study. Table 4 also presents the times required for model construction and model checking of a single property (property **H** of Section 6) using each of the different combinations of model reductions. It can be seen that the decreases in model sizes are also reflected in these timings.

Finally, in Table 5 and Fig. 8, we present a selection of results for the properties analysed in Section 7 on both the complete pathway and the model obtained after applying all three of the reductions outlined above. As can be seen in both the graph and the table, the results obtained for the reduced model are a very good approximation for those of the complete pathway.

10. Conclusions

In this paper we have shown that probabilistic model checking can be a useful tool in the analysis of biological pathways. The technique's key strength is that it allows the calculation of exact quantitative properties for system events occurring over time, and can therefore support a detailed, quantitative analysis of the interactions between the pathway components. By developing a model of a complex, realistic signalling pathway that is not yet well understood, we were able to demonstrate, firstly, that the model is robust and that its predictions agree with biological data [13,14] and, secondly, that probabilistic model checking can be used to obtain a wide range of quantitative measures of system dynamics, thus resulting in deeper understanding of the pathway.

To allow probabilistic model checking to be applied to more complex pathways, we have also discussed a number of exact and approximate techniques for combatting the well-known state-space-explosion problem. In addition, to demonstrate the possible advantages that can be gained through these methods, we have presented the results obtained by applying a number of these approaches to the FGF pathway.

We intend to perform further analysis of the FGF pathway, including an investigation into the effect that changes to reaction rates and initial concentrations will have on the pathway's dynamics. Future work will involve both comparing this probabilistic model checking approach with simulation and ODEs, and also investigation of how to scale the methodology yet further.

Acknowledgements

This work was supported in part by EPSRC grants GR/S11107, GR/S46727, GR/S72023/01, EP/D07956X and EP/D076625, and Microsoft Research Cambridge contract MRL 2005-44. The authors Kwiatkowska, Norman and Parker were in the School of Computer Science at the University of Birmingham when this work was first carried out.

References

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

[2] M. Calder, V. Vyshemirsky, D. Gilbert, R. Orton, Analysis of signalling pathways using continuous time Markov chains, Transactions on Computational Systems Biology 4 (2006) 44–67.

[3] M. Calder, S. Gilmore, J. Hillston, Modelling the influence of RKIP on the ERK signalling pathway using the stochastic process algebra PEPA, Transactions on Computational Systems Biology 7 (2006) 1–23.

[4] J. Rutten, M. Kwiatkowska, G. Norman, D. Parker, Mathematical Techniques for Analyzing Concurrent and Probabilistic Systems, in: CRM Monograph Series, vol. 23, AMS, 2004.

[5] A. Hinton, M. Kwiatkowska, G. Norman, D. Parker, PRISM: A tool for automatic verification of probabilistic systems, in: H. Hermanns, J. Palsberg (Eds.), Proc. 12th Int. Conf. Tools and Algorithms for the Construction and Analysis of Systems, TACAS'06, in: Lecture Notes in Computer Science, vol. 3920, Springer, 2006, pp. 441–444.

[6] PRISM web site. http://www.prismmodelchecker.org/.

[7] M. Kwiatkowska, G. Norman, D. Parker, Probabilistic model checking in practice: Case studies with PRISM, ACM SIGMETRICS Performance Evaluation Review 32 (4) (2005) 16–21.

[8] J. Hillston, A Compositional Approach to Performance Modelling, Cambridge University Press, 1996.

[9] C. Priami, Stochastic π-calculus, The Computer Journal 38 (7) (1995) 578–589.

[10] A. Regev, W. Silverman, E. Shapiro, Representation and simulation of biochemical processes using the π-calculus process algebra, in: R. Altman, A. Dunker, L. Hunter, T. Klein (Eds.), Pacific Symposium on Biocomputing, vol. 6, World Scientific Press, 2001, pp. 459–470.

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

[12] A. Phillips, L. Cardelli, Efficient, correct simulation of biological processes in the stochastic π-calculus, in: M. Calder, S. Gilmore (Eds.), Proc. Int. Conf. Computational Methods in Systems Biology, CMSB'07, in: Lecture Notes in Bioinformatics, vol. 4695, Springer Verlag, 2007, pp. 184–199.

[13] O. Tymchyshyn, S. Akbazardeh, D. McEwan, G. Norman, J. Heath, M. Kwiatkowska, Computational reasoning applied to FGF pathway: Simulation, analysis and experimental validation (submitted for publication).

[14] http://qav.comlab.ox.ac.uk/projects/sysbio/fgf.php.

[15] J. Heath, M. Kwiatkowska, G. Norman, D. Parker, O. Tymchyshyn, Probabilistic model checking of complex biological pathways, in: C. Priami (Ed.), Proc. Int. Conf. Computational Methods in Systems Biology, CMSB'06, in: Lecture Notes in Bioinformatics, vol. 4210, Springer Verlag, 2006, pp. 32–47.

[16] M. Kwiatkowska, G. Norman, D. Parker, Stochastic model checking, in: M. Bernardo, J. Hillston (Eds.), Formal Methods for the Design of Computer, Communication and Software Systems: Performance Evaluation, SFM'07, in: Lecture Notes in Computer Science (Tutorial Volume), vol. 4486, Springer, 2007, pp. 220–270.

[17] A. Aziz, K. Sanwal, V. Singhal, R. Brayton, Verifying continuous time Markov chains, in: R. Alur, T. Henzinger (Eds.), Proc. 8th Int. Conf. Computer Aided Verification, CAV'96, in: Lecture Notes in Computer Science, vol. 1102, Springer, 1996, pp. 269–276.

[18] C. Baier, B. Haverkort, H. Hermanns, J.-P. Katoen, Model checking continuous-time Markov chains by transient analysis, in: A. Emerson, A. Sistla (Eds.), Proc. 12th Int. Conf. Computer Aided Verification, CAV'00, in: Lecture Notes in Computer Science, vol. 1855, Springer, 2000, pp. 358–372.

[19] M. Kwiatkowska, G. Norman, D. Parker, Probabilistic symbolic model checking with PRISM: A hybrid approach, International Journal on Software Tools for Technology Transfer (STTT) 6 (2) (2004) 128–142.

[20] U. Bhalla, Understanding complex signaling networks through models and metaphors, Progress in Biophysics & Molecular Biology 81 (1) (2003) 450–465.

[21] J. Papin, T. Hunter, B. Palsson, S. Subramaniam, Reconstruction of cellular signalling networks and analysis of their properties, Nature Reviews Molecular Cell Biology 6 (2) (2005) 99–111.

[22] V. Eswarakumar, I. Lax, J. Schlessinger, Cellular signaling by fibroblast growth factor receptors, Cytokine Growth Factor Review 16 (2) (2005) 139–149.

[23] J. Schlessinger, Epidermal growth factor receptor pathway, science STKE (Connections Map). stke.sciencemag.org/cgi/cm/stkecm;CMP_14987.

[24] M. Tsang, I. Dawid, Promotion and attenuation of FGF signaling through the Ras-MAPK pathway, Science STKE 228 (2004) pe17.

[25] I. Dikic, S. Giordano, Negative receptor signalling, Current Opinion in Cell Biology 15 (2003) 128–135.

[26] A. Regev, E. Shapiro, Cellular abstractions: Cells as computation, Nature 419 (6905) (2002) 343.

[27] P. Buchholz, Exact and ordinary lumpability in finite Markov chains, Journal of Applied Probability 31 (1994) 59–75.

[28] S. Derisavi, H. Hermanns, W. Sanders, Optimal state-space lumping in Markov chains, Information Processing Letters 87 (6) (2003) 309–315.

[29] J.-P. Katoen, T. Kemna, I. Zapreev, D. Jansen, Bisimulation minimisation mostly speeds up probabilistic model checking, in: O. Grumberg, M. Huth (Eds.), Proc. 13th Int. Conf. Tools and Algorithms for the Construction and Analysis of Systems, TACAS'07, in: Lecture Notes In Computer Science, vol. 4424, Springer, 2007, pp. 87–101.

[30] S. Derisavi, A symbolic algorithm for optimal Markov chain lumping, in: O. Grumberg, M. Huth (Eds.), Proc. 13th Int. Conf. Tools and Algorithms for the Construction and Analysis of Systems, TACAS'07, in: Lecture Notes In Computer Science, vol. 4424, Springer, 2007, pp. 139–154.

[31] H. Hermanns, J.-P. Katoen, Automated compositional Markov chain generation for a plain-old telephone system, Science of Computer Programming 36 (61) (2000) 97–127.

[32] M. Kwiatkowska, G. Norman, D. Parker, Symmetry reduction for probabilistic model checking, in: T. Ball, R. Jones (Eds.), Proc. 18th Int. Conf. Computer Aided Verification, CAV'06, in: Lecture Notes in Computer Science, vol. 4114, Springer-Verlag, 2006, pp. 234–248.

[33] A. Donaldson, A. Miller, Symmetry reduction for probabilistic model checking using generic representatives, in: S. Graf, W. Zhang (Eds.), Proc. 4th Int. Symp. Automated Technology for Verification and Analysis, ATVA'06, in: Lecture Notes in Computer Science, vol. 4218, Springer, 2006, pp. 9–23.

[34] N. Borisov, N. Markevich, J. Hoek, B. Kholodenko, Signaling through receptors and scaffolds: Independent interactions reduce combinatorial complexity, Biophysical Journal 89 (2005) 951–966.

[35] N. Borisov, N. Markevich, J. Hoek, B. Kholodenko, Trading the micro-world of combinatorial complexity for the macro-world of protein interaction domains, Biosystems 83 (2–3) (2006) 152–166.

[36] A. Tikhonov, Systems of differential equations, containing small parameters at the derivatives, Matematicheskii Sbornik 31 (3) (1952) 575–586.

[37] R. Suckley, V. Biktashev, Comparison of asymptotics of heart and nerve exictability, International Journal on Bifurcation & Chaos 13 (3805–3826).

[38] Y. Cao, D. Gillespie, L. Petzold, Accelerated stochastic simulation of the stiff enzyme-substrate reaction, Journal of Chemical Physics 123 (14) (2005) 144917.

[39] C. Rao, A. Arkin, Stochastic chemical kinetics and the quasi-steady-state assumption: Application to the Gillespie algorithm, Journal of Chemical Physics 123 (2003) 4999–5010.

[40] H. Busch, W. Sandmann, V. Wolf, A numerical aggregation algorithm for the enzyme-catalyzed substrate conversion, in: C. Priami (Ed.), Proc. Int. Conf. Computational Methods in Systems Biology, CMSB'06, in: Lecture Notes in Bioinformatics, vol. 4210, Springer Verlag, 2006, pp. 298–311.

[41] A. Bobbio, K. Trivedi, An aggregation technique for the transient analysis of stiff Markov chains, IEEE Transactionon Computers 35 (9) (1986) 803–814.

[42] P. Courtois, Decomposibility: Queueing and Computer System Applications, Academic, New York, 1977.

[43] K.-H. Cho, S.-Y. Shin, H.-W. Kim, O. Wolkenhauser, B. McFerran, W. Kolch, Mathematical modelling of the influence of RKIP on the ERK signalling pathway, in: C. Priami (Ed.), Proc. 1st Int. Workshop Computational Methods in System Biology, in: Lecture Notes in Computer Science, vol. 2602, Springer, 2003, pp. 127–141.

[44] J.-P. Katoen, M. Leucker, D. Willems, V. Wolf, Three-valued abstraction for continuous time Markov chains, in: W. Damm, H. Hermanns (Eds.), Proc. 19th Int. Conf. Computer Aided Verification, CAV'07, in: Lecture Notes in Computer Science, vol. 4590, Springer, 2007, pp. 316–329.

[45] M. Kwiatkowska, G. Norman, D. Parker, O. Tymchyshyn, J. Heath, E. Gaffney, Simulation and verification for computational modelling of signalling pathways, in: L. Perrone, F. Wieland, J. Liu, B. Lawson, D. Nicol, R. Fujimoto (Eds.), Proc. Winter Simulation Conference, Omnipress, 2006, pp. 1666–1675.

[46] E. Gaffney, J. Heath, M. Kwiatkowska, A mass action model of FGF signalling and its simplification (submitted for publication).