Modular modelling of signalling pathways and their cross-talk

Robin Donaldson[a], Muffy Calder[a,*]

[a] *School of Computing Science, University of Glasgow, Sir Alwyn Williams Building, G12 8QQ, United Kingdom*

Abstract

Signalling pathways are well-known abstractions that explain the mechanisms whereby cells respond to signals. Collections of pathways form networks, and interactions between pathways in a network, known as cross-talk, enables further complex signalling behaviours. While there are several formal modelling approaches for signalling pathways, none make cross-talk explicit; the aim of this paper is to define and categorise cross-talk in a rigorous way. We define a modular approach to pathway and network modelling, based on the modules construct in the PRISM modelling language, and a set of generic signalling modules. Five different types of cross-talk are defined according to various biologically meaningful combinations of variable sharing, synchronisation labels and reaction renaming. The approach is illustrated with a case-study analysis of cross-talk between the TGF-β, WNT and MAPK pathways.

1. Introduction

Signalling pathways[1] are well-known abstractions that explain how cells respond to signals. They comprise biochemical reactions that transfer information from a receptor to a target such as the nucleus or mitochondria. Several agent-based, formal modelling techniques from computer science have

[*] Corresponding author

Email addresses: radonald@dcs.gla.ac.uk (Robin Donaldson), muffy.calder@glasgow.ac.uk (Muffy Calder)

[1] we refer simply to pathways henceforth

been extended and applied to signalling pathways in recent years, for example, rewrite rules [1], Petri nets [2] and process algebras [3, 4, 5, 6, 7]. However, there has been less focus on collections of pathways that form networks, and very little on the interactions between pathways, known in the life sciences as *cross-talk*. Cross-talk accounts for many useful behaviours, for example, producing a variety of responses to a single signal, and reuse of proteins between pathways. Cross-talk is an essential aspect of network behaviour, yet there are no known formal models of pathways that make cross-talk explicit. Here we aim to develop a formal framework for pathway and network modelling that allows one to explain, categorise, and detect cross-talk in a systematic way. Our long term motivation is to develop predictive models that inform both systems and synthetic biology.

The paper has four parts. First, we develop a modular modelling framework for pathways and networks, based on the modules construct in the PRISM modelling language and multiway synchronisation. We define a set of generic pathway modules, including basic signalling pathway behaviours such as receptor, 3-stage cascade and gene expression. A pathway is a composition of (instances of) the generic modules; a network is a composition of pathways. We assume continuous-time Markov chain (CTMC) semantics.

Second, we give our main contribution, which is to define and categorise cross-talk according to the way pathways are composed to form a network. Specifically, we consider the combinations of variable sharing, synchronisation labels and reaction renamings that can give rise to biologically meaningful behaviours. We define five types of cross-talk: substrate availability, signal flow, receptor function, gene expression and intracellular communication, and show that the categorisation is well-defined. We can also generate all possible cross-talk from a given network, which provides a rigorous way of generating hypotheses to explain data. We give illustrative examples of each type of cross-talk from biological literature. While we cannot prove our categorisation is complete, we have been unable to find a cross-talk that cannot be categorised.

Third, we apply the approach to a case-study involving three pathways concerning cell growth and differentiation: the TGF-β, WNT and MAPK pathways. One interesting result is that we are able to resolve an ambiguity in the literature concerning one complicated form cross-talk between the WNT and TGF-β pathways.

Finally, we discuss briefly how cross-talk can be detected and categorised for a given model, in the absence of a model description. Such an approach may be useful when we have a model that has been derived from data, rather than the modular description, which may be more appropriate in a synthetic setting (i.e. the design of networks and cross-talk).

The paper is organised as follows.

The following section outlines the background to pathway cross-talk, the temporal logics used in analysis, and related work on pathway and cross-talk modelling.

Sections 3 – 5 contribute to the first part of the paper. Section 3 describes the reagent-centric approach to modelling in the PRISM language. Each reagent in a reaction is mapped to a process, which is represented by a PRISM module with reaction-labelled transitions. Modules are composed using multiway synchronisation over sets of labels. We define two extensions to the language, to allow for variable sharing and synchronisation over labels within a module. In Section 4 we give an example set of generic modules and then define a pathway as a composition of (instances of) those modules, modulo label renaming and hiding. In Section 5 we define a network as a composition of two pathways, possibly with variable sharing, renaming and synchronisation of labels.

Sections 6 and 7 define and categorise cross-talk. In Section 6 we define how the generic modules can be extended with additional reactions, and then define cross-talk in terms of the synchronisations and variable sharings in a network. Section 7 gives our categorisation of the five types of cross-talk. Examples of each are drawn from biological literature and illustrated using two simple pathways; we give a theorem showing that categorisation is well-defined. We give an algorithm to enumerate all cross-talks between two pathways and apply it to the example pathways. We discuss (in Section 6) how to generalise to higher order networks (i.e. three or more pathways), but explain why for practical purposes we have focused on pairwise composition.

In the next sections we give preliminary results on the complementary problem – how to detect (Section 8) and characterise (Section 9) cross-talk in the absence of a model description.

The case-study is in Section 10, where we define a richer set of generic modules, and then apply our approach to the cross-talk between TGF-β, WNT and MAPK pathways, and compare our results with those in the literature.

There follows a discussion in Section 11, and in Section 12 we give our conclusions and directions for future work.

2. Background

In this section we outline the background to pathway cross-talk, temporal logics, and related work on pathway and cross-talk modelling.

2.1. Signal transduction: networks, pathways and cross-talk

A cell has many types of receptors that detect extra- or intra-cellular biochemical signals; signal transduction is the mechanism whereby a cell responds to a detected signal. The result is a signalling network – a collection of pathways that comprise biochemical reactions that transfer information from a receptor to a target such as the nucleus or mitochondria. A typical response initiated at the target is a change in gene expression/protein activation levels, resulting in phenotype changes.

Pathways were first thought to be a linear series of reactions, but more recent results indicate they are non-linear [8]. Many of the reactions involved in signalling pathways are enzyme catalysed protein activations, often arranged in a "signalling cascade". In such a cascade, the activated protein on one "level" is the enzyme for the activating reaction of the next "level," as shown in Figure 1 using standard biochemical graphical notation. The series of reactions forming the pathway can diverge or interact upstream/downstream in the chain of reactions forming a feedback/feedforward loop. We note that pathways are a human abstraction, based on laboratory experiments, to explain and structure the coordination of cellular activity. Although there is a well-known set of pathways in biological literature, there is a lack of rigorous definitions of what constitutes a single pathway and cross-talk between pathways. This paper focuses on defining cross-talk between signalling pathways.

The term cross-talk was first applied to electronic circuits to describe a signal in one circuit having an undesired effect on another circuit [9]. Cross-talk in this setting is a design flaw: the electronic circuit has been specified and built, and has resulted in an undesired interaction between signals, called "signal interference." Biochemical cross-talk [10] is an interaction between signals flowing through two or more signalling pathways in a cell, however, this is not necessarily indicative of signal interference.

We summarise the three fundamental concepts: networks, pathways and cross-talk, as follows.

Pathway: an abstraction that helps life scientists structure the coordination of cellular activity.

Cross-talk: the interaction of two or more pathways.

Network: a collection of pathways and cross-talk that govern how the cell responds to incoming signals.

Throughout the paper the following notation is used. A reaction, for example protein X turns into protein Y, is denoted by a solid line with an arrow. There are two types of modifiers that change the rate of a reaction, catalysis and inhibition. Catalysis (increase in the rate of a reaction) is denoted by a dashed line with an arrow. Inhibition (decrease in the rate of a reaction) is denoted by a solid line with a blunt end. Finally, we distinguish between inactive and active proteins rather than the various mechanisms by which a protein changes state. An active protein is decorated with $*$. This notation is illustrated in Figure 1a).

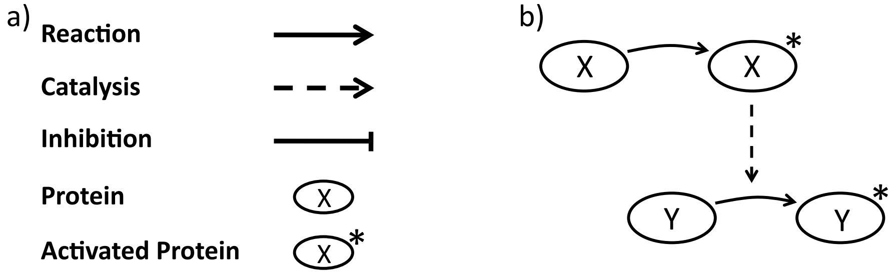


Figure 1: a) the notation used throughout this paper for arcs and nodes, b) an example of a 2-stage signalling cascade in which the activated protein X catalyses the activation of protein Y.

2.2. Temporal logics

In this section we give a brief overview of two temporal logics: the qualitative logic CTL (Computational Tree Logic) and the quantitative logic CSL (Continuous Stochastic Logic). The latter is a quantitative extension of the former with probabilities and timing. In both logics we refer to safety properties ("bad" properties to be avoided) and liveness properties ("good" properties that capture required functionality).

2.2.1. Computational Tree Logic

An Atomic Proposition (AP) is a formula in propositional logic that can be evaluated to a boolean value for a state in a Markov chain. An AP may compare combinations of variables in a Markov chain and constant values, using equalities and inequalities $=$, $<$, $\leq$, etc. The arithmetic operations $+$, $-$, $*$ and $/$ may be applied to any combination of variables and constant values. Looking ahead to the PRISM model of a simple reaction introduced in the next section in Figure 2, examples of APs are: $(A > 0)$, $(A > B + C)$, and $(A = 1)$.

A CTL formula ϕ is defined as follows:

$$\phi ::= AP \mid \neg\ \phi \mid \phi \wedge \phi \mid \phi \vee \phi \mid \mathbf{A}\ X\ \phi \mid \mathbf{E}\ X\ \phi \mid \mathbf{A}\ \phi\ U\ \phi \mid \mathbf{E}\ \phi\ U\ \phi \mid$$
$$\mathbf{A}\ F\ \phi \mid \mathbf{E}\ F\ \phi \mid \mathbf{A}\ G\ \phi \mid \mathbf{E}\ G\ \phi$$

where $\neg$, $\vee$ and $\wedge$ denote "not," "or" and "and" respectively.

Path operators:		
Universal	**A**	all paths from the state
Existential	**E**	at least one path from the state
Temporal operators:		
Next	$X\ \phi$	ϕ holds in the next state
Until	$\phi_1\ U\ \phi_2$	ϕ_1 holds in every state before ϕ_2
Finally	$F\ \phi$	ϕ holds in some future state
Globally	$G\ \phi$	ϕ holds in every state

We also use the non-standard filter construct $\phi\ \{\ \psi\ \}$ as implemented by PRISM. A filter allows a property ϕ to be checked from a state other than the initial state of the Markov chain, in this case a state that satisfies ψ.

2.2.2. Continuous Stochastic Logic

In CSL the path operators **A** and **E** are replaced with the probability operator $\mathbf{P}_{\bowtie \mathbf{x}}$ where x is the probability of the formula and $\bowtie \in \{>, \geq, <, \leq\}$. Hence the operator **A** is equivalent to $\mathbf{P}_{\geq \mathbf{1}}$ and **E** to $\mathbf{P}_{> \mathbf{0}}$. The probability of the formula x can be returned in the PRISM model checker using $\mathbf{P}_{=?}$.

Furthermore, the temporal operators can have a time bound thus for the Finally operator, $F_{\leq 10}\ \phi$, expresses ϕ must become true within 10 time units.

2.3. Related work

There are several computational models of specific pathways that include an aspect of cross-talk. For example, in [11, 12, 13], Ordinary Differential Equations (ODEs) are used to model the cross-talk between the MAPK and AKT pathways, the MAPK and PKC pathways, and the hyperosmolar and the pheromone MAPK pathways respectively. The analysis depends on the pathways involved. In [13] the motivation for modelling is to answer how the pathways maintain signal specificity, given shared common proteins. Two models are proposed, one that contains mutual inhibition between pathways to limit signal bleed-through and one that contains scaffold proteins. In [12] the goal is to investigate whether cross-talk has an effect on bistability, namely whether the signal switches from transient to sustained activation as a consequence of varying the duration of the signal. Cross-talk is expressed implicitly in computational models, i.e. it is part of the system of equations, with no explicit reference to pathways or interactions between pathways. Therefore, there is no direct way to reason about cross-talk, especially to detect or classify the cross-talk.

Formal models are also employed. For example, Petri nets are used to model apoptosis decision-making in the Fas-induced and mitochondrial DNA damage pathways, and this includes the Bid controlled cross-talk between them [14]. [15] contains a discrete, state-based model of the multiple modes of intercellular cross-talk between the EGFR and LIN-12/Notch signalling pathways, developed in the language of Reactive Modules. Model checking is used to check the validity of the model and to generate new biological insights. But, *intercellular* cross-talk is considered within a multi-cellular model. This work bears little relation to *intracellular* cross-talk, indeed, it is considered a misnomer within parts of the life science community. Our focus is cross-talk in a single-cell model.

In general, it is difficult to draw any generic methods or techniques from these specific models.

We are aware of only one paper, [1], that addresses a more generic concept of pathway and cross-talk. Models in [1] are defined using the rewrite rules of the κ calculus; the notion of a "story" corresponds to a pathway and an "influence map" defines how rules can inhibit each other. Superposition of an influence map with a pathway suggests ways in which a story's ending can be delayed or prevented (i.e. delay or prevent pathway output), this can be interpreted as detecting cross-talk. The pathways of [1] are minimal execution paths to a goal, and thus cannot be compared directly with the established signalling pathways (as defined by biologists and used in biological literature) that we consider. Nonetheless, we note that superposition (via renaming and synchronisation) is also fundamental to our approach.

We note this paper extends a preliminary study in [16] in several ways, including enhancements to the PRISM language and a more rigorous treatment of pathways and types of cross-talk.

3. Reagent-centric modelling

We adopt a reagent-centric approach [17] to modelling in which each of the reagents in a reaction is mapped to a process, whose variation reflects increase or decrease in amount of the reagent, through production or consumption. We give a brief outline as follows.

As an example, the reaction $r1$ given in chemical notation by $A + B \xrightarrow{r1} C$ refers to three reagents and so it is modelled by three processes: A, B and C, which are then composed in parallel, synchronising on the event $r1$. After the event $r1$, C is increased and A and B are decreased. The processes can model individuals (molecules) or populations (concentrations of biochemical species); we assume the latter here, and an underlying semantics of continuous-time Markov chain (CTMC) with levels [18].

Definition 1 (CTMC)**.** *Given a finite set of atomic propositions AP, a* continuous-time Markov chain *(CTMC) is a triple $\mathcal{C} = (S, R, L)$ where S is a finite set of states, $R : S \times S \to \mathbb{R}_{\geq 0}$ a rate matrix, and $L : S \to 2^{AP}$ a labelling of states. For a given state s, there is a race between outgoing transitions from s if there is more than one state s' such that $R(s, s') > 0$. The probability that a transition from s to s' completes within t time units when $R(s, s') > 0$ is determined according to the distribution $1 - e^{-R(s,s')\cdot t}$.*

In a CTMC *with levels*, states are characterised by concentration *ranges*, discretised uniformly into N levels with step size h, for each species. Note, the choice of h applies to all species. Though, we can increase or decrease h depending on the degree of granularity of the model that we require.[2]

Definition 2 (CTMC with levels)**.** *The states of a CTMC with levels are vectors of levels $s = ([A_1], [A_2], ..., [A_n])$, where for $i = 1, 2, ..., n$, A_i is a species, and $[A_i]$ is the level of the species A_i. The transitions represent reactions and each transition causes a change in the level number of one or more species, the variation in the number of levels depending on the stoichiometry of the reaction. With each reaction we associate characteristic species vectors pre and post of size n for the set of reactants and the set of products respectively. The reaction can be fired from a state s if $s - pre \geq \mathbf{0}$ and $s - pre + post \leq (N, \ldots, N)$. If a transition from s is taken according to this*

[2] Different choices of h for different species are possible, but only in very restricted circumstances.

reaction, the new state is $s' = s - pre + post$. Assuming all species have the same step size h, stoichiometry 1, and mass action kinetics, then the rate associated with a transition from state u to state v, when the reactants of u are $[R_1], [R_2], ..., [R_m]$ is: $\frac{r*[R_1]*h*[R_2]*h...*[R_m]*h}{h}$.

3.1. Reagent-centric modelling in the PRISM modelling language

Reagent-centric modelling is implemented in a straight forward way in a state-based formalism [4] such as the language of reactive modules [19] for the PRISM model checker [20]. The PRISM language includes modules with local variables and labelled transitions, multiway synchronisation between modules and process algebraic operators. Each process is implemented by a module, and modules are composed with multiway synchronisation on reaction names, which are used to label transitions.

For example, the PRISM model for the reaction $r1$ above, i.e. $A+B \xrightarrow{r1} C$, is in Figure 2. There are three modules: A, B and C, and a system description stating that the three modules run concurrently. Each module has the form: a state variable denoting the species concentration, followed by transitions labelled by the reactions in which the species is a reagent. In this case, there is a single transition labelled $r1$. The transition has the form *condition* → *rate:assignment*, meaning when the condition is true, then perform the assignment at the given rate. The assignments in the first two modules decrease the level by 1 and in the third module increase the level by 1. Initially, there are N levels of A and B and 0 levels of C. Since all three transitions have the same label, they *synchronise*, and when they do, the resulting transition occurs with a rate that is the product of the individual rates, i.e. $e1 * e2 * e3$. The exact definition of the rate expressions depends on the level of detail required in the model. For example, if we require mass action kinetics, then we would define $e1 = A * h$, $e2 = B * h$, and $e3 = r/h$. We note that for the purposes of this study we do not require exact mass action kinetics in our models and we simply use the (multiplicative) identity 1 for all rate expressions.

Synchronisation, renaming and hiding. In general, synchronisation between modules is parameterised by labels as follows. Given modules $M1$ and $M2$, and set of labels L, $M1\ |[L]|\ M2$ denotes the concurrent composition of $M1$ and $M2$, synchronising on *all* labels in L. If the label set is omitted, i.e. $M1\ ||\ M2$, then $M1$ synchronises with $M2$ on the *intersection*

```
module A
  A : [0..N] init N;

  [r1] A > 0 -> e1:(A' = A - 1);
endmodule

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

  [r1] B > 0 -> e2:(B' = B - 1);
endmodule

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

  [r1] C < N -> e3:(C' = C + 1);
endmodule

system
      A || B || C
endsystem
```

Figure 2: A reagent-centric model of $A + B \xrightarrow{r1} C$ in PRISM.

of labels occurring in $M1$ and $M2$. PRISM also allows renaming of labels, denoted thus $M1$ $\{old\_label \leftarrow new\_label\}$, and hiding, denoted thus $M \setminus \{label_1, \ldots, label_n\}$. Hidden labels are not available for synchronisation.

3.2. PRISM language extensions

The PRISM modelling language does not include all the abstractions required for generic modules. We therefore introduce two extensions to make the language more convenient for modelling – they do not add any expressive power to the language.

Variable Sharing. We require two or more modules to reference and update the same variable. We cannot use a PRISM global variable for this purpose because they cannot be updated within a labelled transition. So, we use PRISM local variables and introduce new syntax as follows: $M1 \,|[L,\ V]|\, M2$, where $V = \{(v_1, w_1), \ldots, (v_n, w_n)\}$. (v_i, w_i) is called a *variable sharing* where v_i (local to $M1$) and w_i (local to $M2$) are shared. We implement a variable sharing (v_i, w_i) in PRISM by a pre-processing step in which we substitute w_i for v_i. For each command r in $M1$, we remove all references to v_i from r and define a new command r in $M2$, replacing v_i with w_i; we then synchronise $M1$ and $M2$ over r. We assume that the PRISM modules have the same number of levels N for all variables and so the ranges of the shared variables are the same. The initial value of the shared variable is $max(init(v_i), init(w_i))$ where $init(var)$ is the initial value of the variable var. Because we can now share variables between two modules, we extend the hide operator to hide local variables so that they are unavailable for sharing. Hence we can hide labels and variables in a module thus $M \ \backslash \ \{label_1,\ \ldots,\ label_n,\ var_1,\ \ldots,\ var_n\}$.

Synchronisation Within Modules. We will also require synchronisations involving labelled transitions within the same module. For example, suppose we have two labels $r1$ and $r2$ in module $M1$. Renaming one by the other will not force a synchronisation. $M1\ \{r2\ \leftarrow\ r1\}$ will create two $r1$ labels in $M1$, and a non-deterministic choice between the labels. So, we simply assume an alternative semantics for renaming when the labels are in the same module. With the new semantics, our example $M1\ \{r2\ \leftarrow\ r1\}$ means synchronise $r1$ and $r2$, which we implement by pre-processing (to form a single transition $r1$ that is the conjunction of the transitions for $r1$ and $r2$).

4. Modelling a pathway

We define a *generic pathway module* to be a behavioural pattern within a pathway. For example, commonly occurring pathway modules are: Receptor, 3-stage Cascade and Gene Expression (Figure 3).

The Receptor module has three species (L for ligand, R for receptor and R^* for active receptor) and two reactions (r1 and r2). The 3-stage Cascade module has 3 species (proteins X, Y and Z) and 4 reactions (r3, r4, r5 and r6). The Gene Expression module has 2 species (Gene and Protein) and one reaction (r7). While Gene is not strictly a biochemical species, for modelling purposes we treat it as a species. We use shading to indicate species with initial concentrations (the species present in the initial state).

We represent the three generic pathway modules as PRISM modules with $N = 1$ in Figure 4.[3]

[3] PRISM reserves certain names such as X and does not allow names with the $*$ symbol – strictly we use names such as $XInactive$ instead of X and $XActive$ instead of X^*.

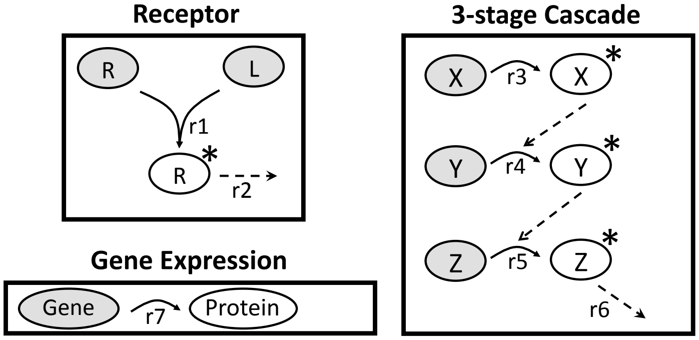


Figure 3: Three generic pathway modules: Receptor, 3-stage Cascade and Gene Expression.

We treat these modules as *generic*, that is, we instantiate them (strictly, duplicate and rename in PRISM) for multiple occurrences. We adopt the following convention. For generic module M, M_i denotes an *instance* of M with every variable and reaction renamed by an indexed form. For example, variable v becomes v_1 in module M_1.

We can compose modules synchronising over sets of labels as follows. Synchronising reaction a in module A with b in module B is achieved by renaming a to b and synchronising the modules over b, i.e. $A\ \{a \leftarrow b\}\ |[b]|\ B$. In this paper we use the term label and reaction synonymously.

A pathway is a parallel composition of instances of generic modules, renaming reactions to coordinate synchronisation within the pathway.

Definition 3 (Pathway)**.** *Let G be a set of generic modules. A pathway P has the form $(X_1 f_1\ |[L_1]|\ \ldots\ |[L_{n-1}]|\ X_n f_n)\ \setminus\ H$ where $X_1\ \ldots\ X_n$ are instances of modules in G, $f_1\ \ldots\ f_n$ are sets of renamings, $L_1\ \ldots\ L_{n-1}$ are labels (reactions) and H is a set of hidings.*

Definition 4 (Renaming pathway reactions)**.** *The reactions in a pathway P can be renamed creating a new pathway $P' = P\ \{renamings\}$ where renamings is a set of renamings.*

As an example, consider pathway $Pathway_1$ comprising instances of the Receptor, 3-stage Cascade and Gene Expression modules:

$$\begin{aligned} Pathway_1 \;=\; & (Receptor_1\ \{r2_1 \leftarrow r3_1\} \\ & |[r3_1]| \end{aligned}$$

```
module Receptor
   R : [0..1] init 1;      L : [0..1] init 1;      R* : [0..1] init 0;

   [r1] R = 1 & L = 1 & R* = 0 -> 1:(R’ = 0) & (L’ = 0) & (R*’ = 1);
   [r2] R* = 1 -> 1:true;
endmodule


module 3StageCascade
   X : [0..1] init 1;   X* : [0..1] init 0;
   Y : [0..1] init 1;   Y* : [0..1] init 0;
   Z : [0..1] init 1;   Z* : [0..1] init 0;

   [r3] X = 1 & X* = 0 -> 1:(X’ = 0) & (X*’ = 1);
   [r4] Y = 1 & Y* = 0 & X* = 1 -> 1:(Y’ = 0) & (Y*’ = 1);
   [r5] Z = 1 & Z* = 0 & Y* = 1 -> 1:(Z’ = 0) & (Z*’ = 1);
   [r6] Z* = 1 -> 1:true;
endmodule


module GeneExpression
   Gene : [0..1] init 1;      Protein : [0..1] init 0;

   [r7] Gene = 1 & Protein = 0 -> 1:(Gene’ = 0) & (Protein’ = 1);
endmodule
```

Figure 4: The three generic modules in PRISM with $N = 1$.

$$\begin{array}{l} 3StageCascade_1\ \{r6_1 \leftarrow r7_1\} \\ |[r7_1]| \\ GeneExpression_1) \\ \setminus \{r1_1, r3_1,\ r4_1, r5_1, R_1, L_1, R_1^*, Gene_1, Protein_1\} \end{array}$$

$Receptor_1$ and $3StageCascade_1$ modules synchronise on $r2_1$ and $r3_1$, and $3StageCascade_1$ and $GeneExpression_1$ synchronise on $r6_1$ and $r7_1$ (strictly, we rename $r2_1$ to $r3_1$ and synchronise the modules on $r3_1$, and similarly for $r6_1$ and $r7_1$). Because of these synchronisations, the active receptor catalyses the activation of protein X and active protein Z catalyses the expression of Gene. Reactions $r1_1$, $r3_1$, $r4_1$ and $r5_1$ and variables R_1, L_1, R_1^*, $Gene_1$ and $Protein_1$ are hidden using the \ operator.

Reactions and (local) variables are considered to be external or internal.

Definition 5 (External reactions and variables)**.** *For a pathway P, the set of external reactions, $ext_r(P)$, is the set of reactions, modulo renamings, that have not been hidden and the set of external variables, $ext_v(P)$, is the set of (local) variables that have not been hidden. External reactions are available for synchronisation and external variables are available for sharing.*

Hence, $ext_r(Pathway_1) = \{r7_1\}$ and $ext_v(Pathway_1) = \{X_1, Y_1, Z_1, X_1^*, Y_1^*, Z_1^*\}$.

As a further example we define pathway $Pathway_2$:

$$\begin{array}{lcl} Pathway_2 & = & (Receptor_2\ \{r2_2 \leftarrow r3_2\} \\ & & |[r3_2]| \\ & & 3StageCascade_2\ \{r6_2 \leftarrow r7_2\} \\ & & |[r7_2]| \\ & & GeneExpression_2) \\ & & \setminus \{r1_2, r3_2,\ r4_2, r5_2, R_2, L_2, R_2^*, Gene_2, Protein_2\} \end{array}$$

With $ext_r(Pathway_2) = \{r7_2\}$ and $ext_v(Pathway_2) = \{X_2, Y_2, Z_2, X_2^*, Y_2^*, Z_2^*\}$.

Pathways $Pathway_1$ and $Pathway_2$ are shown graphically in Figure 5.

We now consider *networks* of pathways.

5. Modelling a network of independent pathways

Here we give the general definition of a network, and then consider the special case of networks of independent pathways. Later we consider networks with cross-talk.

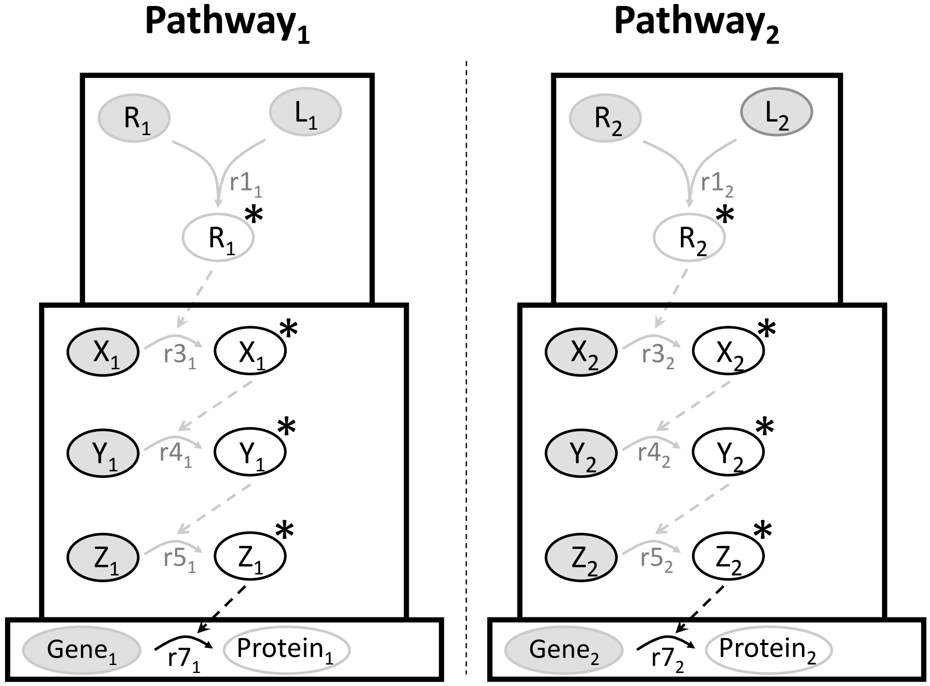


Figure 5: The two pathways $Pathway_1$ and $Pathway_2$ each comprise three instances of the generic pathway modules Receptor, 3-stage Cascade and Gene Expression. External reactions and variables are denoted by black lines, internal reactions and variables by grey lines and species that are present in the initial state by shaded ellipses.

A network is a parallel composition of pathways, with optional synchronisation of external reactions and sharing of variables between pathways.

Definition 6 (Network). *A network is a composition of two pathways of the form* P_1 $\{renamings_1\}$ $|[E \cup U,\ V]|$ P_2 $\{renamings_2\}$ *where* $|[E \cup U,\ V]|$ *defines the interaction between* P_1 *and* P_2. $renamings_1$ *and* $renamings_2$ *are optional sets of renamings of reactions.* $E = ext_r(P_1\ \{renamings_1\}) \cap ext_r(P_2\ \{renamings_2\})$*, in other words,* E *is the intersection of the sets of external reactions, modulo renamings, in* P_1 *and* P_2. V *is a set of variable sharings between* P_1 *and* P_2. $U \subseteq ext_r(P_1\ \{renamings_1\}) \cup ext_r(P_2\ \{renamings_2$ $\})$.

Note, P_1, P_2, $renamings_1$, $renamings_2$, U and V determine the network. Now consider the special case of a network of independent pathways.

Definition 7 (Independent pathways)**.** *A network of two pathways* P_1 $\{renamings_1\}$ $|[E \cup U,\ V]|$ P_2 $\{renamings_2\}$ *is independent if there is no synchronisation of reactions and sharing of variables between the pathways, hence* $E = \emptyset$ *and* $V = \emptyset$*.*

We can compose our two example pathways independently thus: $Pathway_1$ $|[E\ \cup\ U,\ V]|$ $Pathway_2$ where $E = V = U = \emptyset$.

We now turn our attention to the case where there is synchronisation of reactions or sharing of variables between the pathways, i.e. there is cross-talk. However, before doing so we introduce the concept of auxiliary reactions, and ultimately how they result in unused reactions, i.e. the set U.

6. Auxiliary reactions

Auxiliary reactions are additional basic reactions and modifiers that can be used to express interactions between pathways.

Definition 8 (Auxiliary reactions)**.** *There are four types of auxiliary reactions for a species* X *as given in Figure 6.*

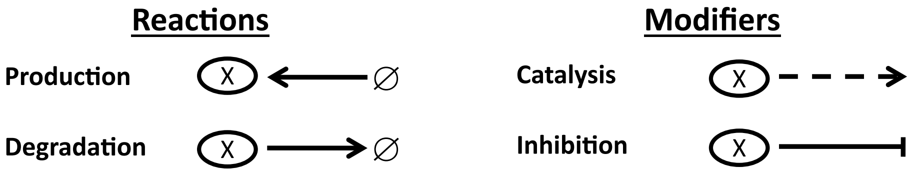


Figure 6: The four types of auxiliary reactions, two are reactions and two are modifiers.

Production and degradation reactions are the two basic reactions for any species. All other reactions can be defined by synchronising production and degradation reactions. For example, we can express the formation of Z from X and Y by synchronising the degradation of X and Y with the production of Z.

Catalysis and inhibition are modifiers: they change the (PRISM) condition of reactions. Catalysis and inhibition auxiliary reactions must synchronise with a reaction to make (biological) sense.

For any species X we can add any number of any type of auxiliary reactions.

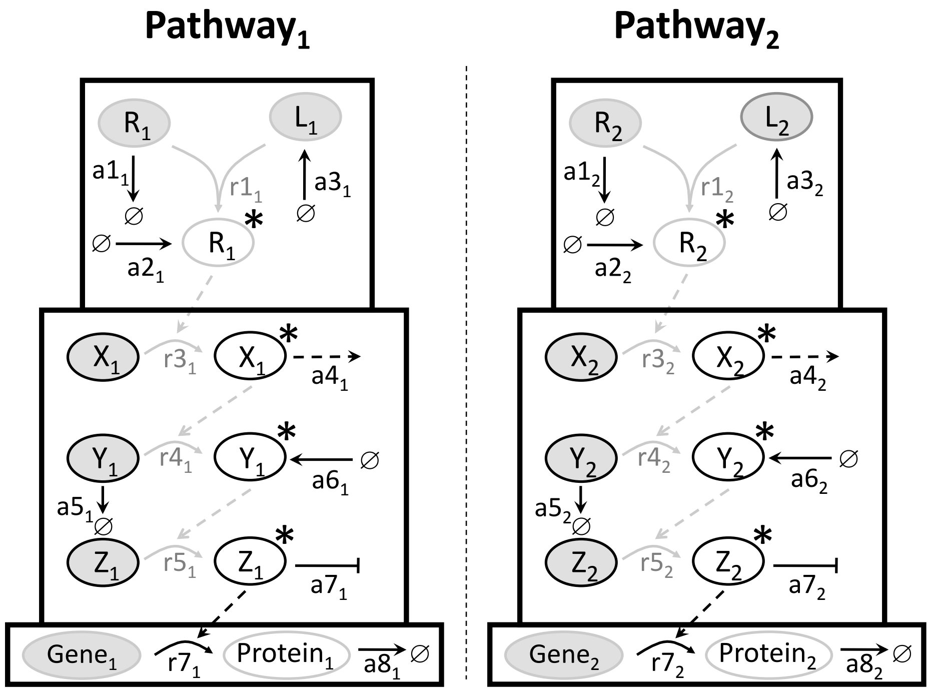


Figure 7: The two pathways $Pathway_1$ and $Pathway_2$ with added auxiliary reactions.

Definition 9 (PRISM implementation)**.** *For any species X in a module, for any i, any of the 4 types of auxiliary reactions can be added as follows:*

```
[prod_i]  X = 0 -> 1:(X' = 1);
[deg_i]   X = 1 -> 1:(X' = 0);
[cat_i]   X = 1 -> 1:true;
[inhib_i] X = 0 -> 1:true;
```

Note, although we have not defined an explicit syntax for adding auxiliary reactions, we assume for any given pathway P we can augment it with a given set of auxiliary reactions $aux(P)$. For a pathway P there is an infinite number of augmentations of auxiliary reactions.

Definition 10 (Pathway auxiliary reactions)**.** *For a given pathway P, the set of auxiliary reactions is $aux(P)$ and we extend $ext_r(P)$ to include $aux(P)$, i.e. all auxiliary reactions are external.*

We add to our two example pathways some auxiliary reactions, the motivation for these will be given in the next section.

We adopt the following convention. In $Pathway_j$ we label auxiliary reaction i as ai_j (or ai_j in PRISM).

In the Receptor module in $Pathway_j$ we add:

```
[a1_j] R = 1 -> 1:(R'=0);
[a2_j] R* = 0 -> 1:(R*' = 1);
[a3_j] L = 0 -> 1:(L' = 1);
```

In the 3-stage Cascade module in $Pathway_j$ we add:

```
[a4_j] X* = 1 -> 1:true;
[a5_j] Y = 1 -> 1:(Y'=0);
[a6_j] Y* = 0 -> 1:(Y*'=1);
[a7_j] Z* = 0 -> 1:true;
```

In the Gene Expression module in $Pathway_j$ we add:

```
[a8_j] Protein = 1 -> 1:(Protein' = 0);
```

The pathways with added auxiliary reactions are shown graphically in Figure 7.

Auxiliary reactions are an integral part of modelling cross-talk. We model cross-talk by different combinations of synchronisation of external reactions (which includes the auxiliary reactions) and sharing of variables.

Definition 11 (Cross-talk)**.** *Given a network of two pathways P_1 $\{renamings_1\}$ $|[E \cup U, V]|$ P_2 $\{renamings_2\}$, there is cross-talk if there is at least one reaction $e \in E$ or one variable sharing $v \in V$. The number of cross-talks is $|E| + |V|$.*

We now introduce the concept of unused reactions U with the aid of the following functions.

Definition 12 (Mapped)**.** *Given a network of two pathways P_1 $\{renamings_1\}$ $|[E \cup U, V]|$ P_2 $\{renamings_2\}$, for any $e \in E$, we define the function $mapped(e) = \{x_i | x_i \leftarrow e\} \cup \{u_i | u_i \leftarrow e\}$ where $renamings_1 = \{x_1 \leftarrow y_1, \ldots, x_n \leftarrow y_n\}$ and $renamings_2 = \{u_1 \leftarrow v_1, \ldots, u_n \leftarrow v_n\}$. We also define $mapped(E) = \bigcup_{e \in E} mapped(e)$. In other words, $mapped(e)$ is the set of reactions involved in one synchronisation e between the pathways and $mapped(E)$ is the set of reactions involved in any synchronisation $e \in E$ between the pathways.*

The unused reactions U are the auxiliary reactions that are not used for synchronisation between two pathways, and so $U = (aux(P_1) \cup aux(P_2)) \setminus mapped(E)$. Note that since each reaction in U occurs in only one pathway, they cannot synchronise and their corresponding transitions never execute. This will be explored in detail in the next section, where we turn our attention to a network of two pathways in which there is cross-talk.

7. Categorisation of cross-talk

Although there is some discussion of types of cross-talk [21], there appears to be no universal categorisation in the literature. In this section we propose that there are five types of cross-talk: substrate availability, signal flow, receptor function, gene expression and intracellular communication. We note that four of the five types are alluded to in [22] but are not made specific.

We give the motivation for the five types using indicative examples from the literature. We define several functions that are used in the formalisation. We then formalise the types and prove that the types are distinct. Finally, we give examples of each type of cross-talk using our example pathways $Pathway_1$ and $Pathway_2$.

7.1. Motivation for types

In this section we give evidence of each of the five types. While there is no proof that there are no other types of cross-talk, we have found no examples after performing an exhaustive literature search and discussing this with domain experts.

Substrate availability cross-talk In [13] there are two pathways that compete for activation of the MAPK cascade. The pathways share the MAPKKK protein STE11 and have homologous MAPKK and MAPK proteins.

Signal flow cross-talk In [10] there is signal flow cross-talk between the MAPK and Integrin signalling pathways. Activation of the Integrin pathway enhances signalling through the MAPK pathway by increased rate of activation of key proteins in the pathway.

Receptor function cross-talk In [23] other signalling pathways can activate the Estrogen receptor in the absence of the Estrogen ligand.

Gene expression cross-talk In [24] two pathways contain cross-talk within the nucleus. One pathway contains a transcription factor GR that resides outside the nucleus. Upon signalling, GR relocates to the nucleus and represses the transcription factor NF-κB that is activated by another pathway.

Intracellular communication cross-talk In [21] the TGF-β and WNT pathways reciprocally regulate the production of their ligands. There is some contention in the literature as to whether this is genuine cross-talk: the interaction is less direct than other types of cross-talk and involves lengthy processes such as gene expression and ligand excretion.

We now turn to formalising the 5 types, but before doing so, we define several useful functions on the generic modules.

7.2. Functions on modules

We define several functions that operate on modules, of type $func : Module \rightarrow \{Labels\}$.

all – all reactions
$all(Receptor) = \{a1, a2, a3, r1\}$
$all(3StageCascade) = \{a4, a5, a6, a7, r3, r4, r5\}$
$all(GeneExpression) = \{a8, r7\}$

$trans$ – all transformation reactions
$trans(Receptor) = \{a1, a2, a3, r1\}$
$trans(3StageCascade) = \{a5, a6, r3, r4, r5\}$
$trans(GeneExpression) = \{a8, r7\}$

mod – all modifiers
$mod(Receptor) = \emptyset$
$mod(3StageCascade) = \{a4, a7\}$
$mod(GeneExpression) = \emptyset$

Note that $\forall x.all(x) = trans(x) \cup mod(x)$.

$catalysis$ – all catalysis reactions
$catalysis(Receptor) = \emptyset$

$catalysis(3StageCascade) = \{a4\}$
$catalysis(GeneExpression) = \emptyset$

$inhib$ – all inhibition reactions
$inhib(Receptor) = \emptyset$
$inhib(3StageCascade) = \{a7\}$
$inhib(GeneExpression) = \emptyset$

$prod$ – all production reactions
$prod(Receptor) = \{a2, a3\}$
$prod(3StageCascade) = \{a6\}$
$prod(GeneExpression) = \emptyset$

deg – all degradation reactions
$deg(Receptor) = \{a1\}$
$deg(3StageCascade) = \{a5\}$
$deg(GeneExpression) = \{a8\}$

$receptor\_deg$ – all degradation of (inactive) receptor reactions
$receptor\_deg(Receptor) = \{a1\}$
$receptor\_deg(3StageCascade) = \emptyset$
$receptor\_deg(GeneExpression) = \emptyset$

$receptor\_act$ – all ligand-receptor binding reactions
$receptor\_act(Receptor) = \{r1\}$
$receptor\_act(3StageCascade) = \emptyset$
$receptor\_act(GeneExpression) = \emptyset$

$active\_receptor\_prod$ – all production of active receptor reactions
$active\_receptor\_prod(Receptor) = \{a2\}$
$active\_receptor\_prod(3StageCascade) = \emptyset$
$active\_receptor\_prod(GeneExpression) = \emptyset$

$ligand\_prod$ – all production of ligand reactions
$ligand\_prod(Receptor) = \{a3\}$
$ligand\_prod(3StageCascade) = \emptyset$
$ligand\_prod(GeneExpression) = \emptyset$

$gene\_expression$ – all gene expression reactions
$gene\_expression(Receptor) = \emptyset$
$gene\_expression(3StageCascade) = \emptyset$
$gene\_expression(GeneExpression) = \{r7\}$

7.3. Cross-talk types

We now formalise the 5 types of cross-talk: substrate availability, signal flow, receptor function, gene expression and intracellular communication, in terms of the three modules introduced so far. Extending the formalisation to include extra modules that behave in a similar way is trivial.

Given a network of the form $P_1\ \{renamings_1\}\ |[E \ \cup \ U,\ V]|\ P_2\ \{renamings_2\}$, a cross-talk is either a $e \in E$ or a $v \in V$.

A cross-talk $v \in V$ is always **substrate availability cross-talk**.

A cross-talk $e \in E$ is categorised according to the rules below. Rules are either necessary rules or biological constraints. Biological constraints prevent infeasible cross-talk due to, for example, different cellular locations or different species types. These constraints concern which reactions can be synchronised between two pathways, thus a constraint always concerns members of $mapped(e)$.

We use the notation $Module\_Type \in P_i$ to mean a *module* in P_i of type $Module\_Type$. The operator $\exists! x.f(x)$ is defined by $\exists x.(f(x) \wedge \forall y.(f(y) \rightarrow y = x))$

Finally, we note by the definition of E in a network the following properties hold.

- $\forall e \in E. \forall x \in mapped(e).(x \in P_1 \vee x \in P_2)$
- $\forall e \in E. \exists x \in mapped(e).(x \in P_1)$
- $\forall e \in E. \exists x \in mapped(e).(x \in P_2)$

Signal flow cross-talk
e is signal flow cross-talk if and only if the rules in Case 1 or Case 2 hold.

Case 1: P_1 affects a transformation reaction in P_2 or vice-versa.

Rules

$\exists x \in \mathit{mapped}(e).\exists 3StageCascade \in (P_1 \cup P_2).x \in \mathit{trans}(3StageCascade)$

Biological constraints

$\forall x \in \mathit{mapped}(e).\forall \mathit{Receptor} \in (P_1 \cup P_2).x \notin \mathit{trans}(\mathit{Receptor})$

$\forall x \in \mathit{mapped}(e).\forall \mathit{GeneExpression} \in (P_1 \cup P_2).x \notin \mathit{all}(\mathit{GeneExpression})$

Case 2: P_1 produces a protein in P_2 or vice-versa, or else P_2 catalyses P_1's production of a protein, or vice-versa, or P_2 inhibits P_1 from producing a protein, or vice-versa.

Rules

$\exists! x \in \mathit{mapped}(e).\exists \mathit{GeneExpression} \in (P_1 \cup P_2).x \in \mathit{deg}(\mathit{GeneExpression})$

$\exists! x \in \mathit{mapped}(e).\exists 3StageCascade \in (P_1 \cup P_2).x \in \mathit{prod}(3StageCascade)$

Biological constraints

$\forall x \in \mathit{mapped}(e).\forall \mathit{Receptor} \in (P_1 \cup P_2).x \notin \mathit{all}(\mathit{Receptor})$

$\forall x \in \mathit{mapped}(e).\forall 3StageCascade \in (P_1 \cup P_2).x \in \mathit{trans}(3StageCascade)$
$\quad \rightarrow x \in \mathit{prod}(3StageCascade)$

$\forall x \in \mathit{mapped}(e).\forall \mathit{GeneExpression} \in (P_1 \cup P_2).x \in \mathit{all}(\mathit{GeneExpression})$
$\quad \rightarrow x \in \mathit{deg}(\mathit{GeneExpression})$

Receptor function cross-talk

e is receptor function cross-talk if and only if the rules in Case 1, Case 2 or Case 3 hold.

Case 1: P_1 catalyses P_2's receptor degradation, or vice-versa, with possible modifiers from 3-stage cascades.

Rules

$\exists! x \in \mathit{mapped}(e).\exists \mathit{Receptor} \in P_i.x \in \mathit{receptor\_deg}(\mathit{Receptor})$

$\exists x \in \mathit{mapped}(e).\exists 3StageCascade \in P_j.x \in \mathit{catalysis}(3StageCascade)$

where $i \neq j$

Biological constraints

$\forall x \in \mathit{mapped}(e).\forall \mathit{GeneExpression} \in (P_1 \cup P_2).x \notin \mathit{all}(\mathit{GeneExpression})$

$\forall x \in \mathit{mapped}(e).\forall 3StageCascade \in (P_1 \cup P_2).x \notin \mathit{trans}(3StageCascade)$

$\forall x \in \mathit{mapped}(e).\forall \mathit{Receptor} \in (P_1 \cup P_2).x \in \mathit{all}(\mathit{Receptor})$
$\quad \rightarrow x \in \mathit{receptor\_deg}(\mathit{Receptor})$

Case 2: The activation of P_1's receptor is inhibited by P_2, or vice-versa, with possible extra modifiers from 3-stage cascades.

Rules

$\exists! x \in \mathit{mapped}(e).\exists \mathit{Receptor} \in P_i.x \in \mathit{receptor\_act}(\mathit{Receptor})$

$\exists x \in \mathit{mapped}(e).\exists 3\mathit{StageCascade} \in P_j.x \in \mathit{inhib}(3\mathit{StageCascade})$

where $i \neq j$

Biological constraints

$\forall x \in \mathit{mapped}(e).\forall \mathit{GeneExpression} \in (P_1 \cup P_2).x \notin \mathit{all}(\mathit{GeneExpression})$

$\forall x \in \mathit{mapped}(e).\forall 3\mathit{StageCascade} \in (P_1 \cup P_2).x \notin \mathit{trans}(3\mathit{StageCascade})$

$\forall x \in \mathit{mapped}(e).\forall \mathit{Receptor} \in (P_1 \cup P_2).x \in \mathit{all}(\mathit{Receptor})$
$\quad \rightarrow x \in \mathit{receptor\_act}(\mathit{Receptor})$

Case 3: P_1's receptor is activated without the need for a ligand and this is catalysed by P_2, or vice-versa, with possible extra modifiers from 3-stage cascades.

Rules

$\exists! x \in \mathit{mapped}(e).\exists \mathit{Receptor} \in P_i.x \in \mathit{receptor\_deg}(\mathit{Receptor})$

$\exists! x \in \mathit{mapped}(e).\exists \mathit{Receptor} \in P_i.x \in \mathit{active\_receptor\_prod}(\mathit{Receptor})$

$\exists x \in \mathit{mapped}(e).\exists 3\mathit{StageCascade} \in P_j.x \in \mathit{catalysis}(3\mathit{StageCascade})$

where $i \neq j$

Biological constraints

$\forall x \in \mathit{mapped}(e).\forall 3\mathit{StageCascade} \in (P_1 \cup P_2).x \notin \mathit{trans}(3\mathit{StageCascade})$

$\forall x \in \mathit{mapped}(e).\forall \mathit{GeneExpression} \in (P_1 \cup P_2).x \notin \mathit{all}(\mathit{GeneExpression})$

$\forall x \in \mathit{mapped}(e).\forall \mathit{Receptor} \in (P_1 \cup P_2).x \in \mathit{all}(\mathit{Receptor})$
$\quad \rightarrow (x \in \mathit{receptor\_deg}(\mathit{Receptor}) \vee x \in \mathit{active\_receptor\_prod}(\mathit{Receptor}))$

Gene expression cross-talk

The rate of P_1's gene expression reaction is modified by a species in a 3-stage cascade module in P_2 or vice-versa.

Rules

$\exists! x \in \mathit{mapped}(e).\exists \mathit{GeneExpression} \in (P_1 \cup P_2).$
$\quad x \in \mathit{gene\_expression}(\mathit{GeneExpression})$
$\exists x \in \mathit{mapped}(e).\exists 3\mathit{StageCascade} \in (P_1 \cup P_2).x \in \mathit{mod}(3\mathit{StageCascade})$

Biological constraints

$\forall x \in \mathit{mapped}(e).\forall \mathit{Receptor} \in (P_1 \cup P_2).x \notin \mathit{all}(\mathit{Receptor})$
$\forall x \in \mathit{mapped}(e).\forall 3\mathit{StageCascade} \in (P_1 \cup P_2).x \notin \mathit{trans}(3\mathit{StageCascade})$
$\forall x \in \mathit{mapped}(e).\forall \mathit{GeneExpression} \in (P_1 \cup P_2).x \in \mathit{all}(\mathit{GeneExpression})$
$\quad \rightarrow x \in \mathit{gene\_expression}(\mathit{GeneExpression})$

Intracellular Communication cross-talk

A protein is released from P_1 that is the ligand for P_2, or vice-versa, with possible extra modifiers from 3-stage cascades.

Rules

$(\exists! x \in \mathit{mapped}(e).\exists \mathit{GeneExpression} \in P_i.x \in \mathit{deg}(\mathit{GeneExpression}))\vee$
$\quad (\exists! x \in \mathit{mapped}(e).\exists 3\mathit{StageCascade} \in P_i.x \in \mathit{deg}(3\mathit{StageCascade}))$
$\exists x \in \mathit{mapped}(e).\exists \mathit{Receptor} \in P_j.x \in \mathit{ligand\_prod}(\mathit{Receptor})$
where $i \neq j$

Biological constraints

$\forall x \in \mathit{mapped}(e).\forall \mathit{GeneExpression} \in (P_1 \cup P_2).x \in \mathit{all}(\mathit{GeneExpression})$
$\quad \rightarrow x \in \mathit{deg}(\mathit{GeneExpression})$
$\forall x \in \mathit{mapped}(e).\forall 3\mathit{StageCascade} \in (P_1 \cup P_2).x \in \mathit{trans}(3\mathit{StageCascade})$
$\quad \rightarrow x \in \mathit{deg}(3\mathit{StageCascade})$
$\forall x \in \mathit{mapped}(e).\forall \mathit{Receptor} \in (P_1 \cup P_2).x \in \mathit{all}(\mathit{Receptor})$
$\quad \rightarrow x \in \mathit{ligand\_prod}(\mathit{Receptor})$

7.4. Categorisation is well-defined

We prove below that the categorisation is well-defined, i.e. that any cross-talk that has been categorised has only one type.

Theorem 1. *Categorisation is well-defined.*

Proof. Trivially, any cross-talk v is of type substrate availability cross-talk.

We now in turn assume a cross-talk e of each type and give a *witness*, a label that must/must not be part of the cross-talk, that prevents it from being another type of cross-talk.

Suppose e has been categorised as signal flow cross-talk. In both cases of signal flow cross-talk, no transformation reactions from a Receptor module are allowed in $\mathit{mapped}(e)$, hence $x \in \{a1, a2, a3, r1\} \rightarrow x \notin \mathit{mapped}(e)$.

A receptor function cross-talk must have a label from $\mathit{receptor\_deg}(\mathit{Receptor})$ (Case 1 and Case 3) or a label from $\mathit{receptor\_act}(\mathit{Receptor})$ (Case 2). Therefore, $a1 \in \mathit{mapped}(e) \vee r1 \in \mathit{mapped}(e)$.

The witness to e not being a receptor function cross-talk is $a1 \notin \mathit{mapped}(e)$ and $r1 \notin \mathit{mapped}(e)$.

Assuming e is each type of cross-talk in turn, we list below the witnesses that prove e can have no other type.

e is signal flow cross-talk	Witness
cannot be receptor function	$\forall x \in \{a1, r1\}.x \notin \mathit{mapped}(e)$
cannot be gene expression	$r7 \notin \mathit{mapped}(e)$
cannot be intracellular communication	$a3 \notin \mathit{mapped}(e)$

e is receptor function cross-talk	Witness
cannot be signal flow	$\forall x \in \{a5, a6, r3, r4, r5\}.$ $x \notin \mathit{mapped}(e)$
cannot be gene expression	$r7 \notin \mathit{mapped}(e)$
cannot be intracellular communication	$a3 \notin \mathit{mapped}(e)$

e is gene expression cross-talk	Witness
cannot be signal flow	$\forall x \in \{a5, a6, r3, r4, r5\}.$ $x \notin \mathit{mapped}(e)$
cannot be receptor function	$\forall x \in \{a1, r1\}.x \notin \mathit{mapped}(e)$
cannot be intracellular communication	$a3 \notin \mathit{mapped}(e)$

e is intracellular communication cross-talk	Witness
cannot be signal flow	$a3 \in \mathit{mapped}(e)$
cannot be receptor function	$\forall x \in \{a1, r1\}.x \notin \mathit{mapped}(e)$
cannot be gene expression	$r7 \notin \mathit{mapped}(e)$

□

7.5. Examples of cross-talk

Given two pathways, we can generate all possible instances of cross-talk; the algorithm is given in Appendix A, it depends upon k, the maximum number of synchronisations on a reaction. Applying the algorithm to our two example pathways, $Pathway_1$ and $Pathway_2$, with $k = 3$, yields 757 *candidate* examples of cross-talk (that do not necessarily satisfy the biological constraints) and 175 *actual* examples of cross-talk. The latter are categorised as follows: 36 substrate availability, 65 signal flow, 18 receptor function, 28 gene expression, and 28 intracellular communication. Below, we give an example from each type.

Substrate availability cross-talk The pathways compete for activation of protein X, hence the pathways share variables X_1 and X_2.
$Pathway_1 \; |[E \;\cup\; U,\; V]| \; Pathway_2$

where $U = aux(Pathway_1) \cup aux(Pathway_2)$, $E = \emptyset$ and $V = \{(X_1, X_2)\}$. This example is shown in Figure 8(a).

Signal flow cross-talk An alternative reaction to activate Y_1 through the X_2^* enzyme. Synchronise $a5_1$, the degradation of Y_1, with $a6_1$, the production of Y_1^* to create the reaction[4] $Y_1 \rightarrow Y_1^*$, and also synchronise with $a4_2$, the enzymatic activity of X_2^*.
$Pathway_1 \, \{a5_1 \leftarrow r_{new}, a6_1 \leftarrow r_{new}\} \, |[E \cup U,\, V]| \, Pathway_2 \, \{a4_2 \leftarrow r_{new}\}$

where $U = aux(Pathway_1) \;\cup\; aux(Pathway_2) \;\setminus\; \{a5_1, a6_1, a4_2\}$, $E = \{r_{new}\}$ and $V = \emptyset$. This example is shown in Figure 8(b).

Receptor function cross-talk An alternative reaction to activate receptor R_2 by the enzyme X_1^*. Synchronise $a1_2$, the degradation of receptor R_2, with $a2_2$, the production of the active receptor R_2^* to create the reaction $R_2 \rightarrow R_2^*$, and also synchronise $a4_1$, the enzymatic activity of X_1^*.
$Pathway_1 \, \{a4_1 \leftarrow r_{new}\} \, |[E \cup U,\, V]| \, Pathway_2 \, \{a1_2 \leftarrow r_{new}, a2_2 \leftarrow r_{new}\}$

where $U = aux(Pathway_1) \;\cup\; aux(Pathway_2) \;\setminus\; \{a4_1, a1_2, a2_2\}$, $E = \{r_{new}\}$ and $V = \emptyset$. This example is shown in Figure 8(c).

Gene expression cross-talk Inhibit the expression of $Gene_1$ by the Z_2^* protein. Synchronise $a7_2$, the inhibiting activity of Z_2^*, with $r7_1$, the expression of $Gene_1$.

[4] notice that this is synchronisation within a module

$Pathway_1\ \{r7_1\ \leftarrow\ r_{new}\}\ |[E\ \cup\ U,\ V]|\ Pathway_2\ \{a7_2\ \leftarrow\ r_{new}\}$

where $U = aux(Pathway_1)\ \cup\ aux(Pathway_2)\ \setminus\ \{a7_2, r7_1\}$, $E = \{r_{new}\}$ and $V = \emptyset$. This example is shown in Figure 8(d).

Intracellular communication cross-talk The output of expressing $Gene_1$ is the ligand for $Pathway_2$. Synchronise $a8_1$, the degradation of $Protein_1$, with $a3_2$, the production of the ligand L_2 to create the reaction $Protein_1 \rightarrow L_2$.

$Pathway_1\ \{a8_1\ \leftarrow\ r_{new}\}\ |[E\ \cup\ U,\ V]|\ Pathway_2\ \{a3_2\ \leftarrow\ r_{new}\}$

where $U = aux(Pathway_1)\ \cup\ aux(Pathway_2)\ \setminus\ \{a8_1, a3_2\}$, $E = \{r_{new}\}$ and $V = \emptyset$. This example is shown in Figure 8(e).

7.6. Higher order networks

We have defined how to model networks of two pathways. Higher order networks can be modelled by composing a network with a single pathway, hence:

$Network_2 = Pathway_1\ |[E_1\ \cup\ U_1,\ V_1]|\ Pathway_2$

$Network_3 = Network_2\ |[E_2\ \cup\ U_2,\ V_2]|\ Pathway_3$

$\ldots$

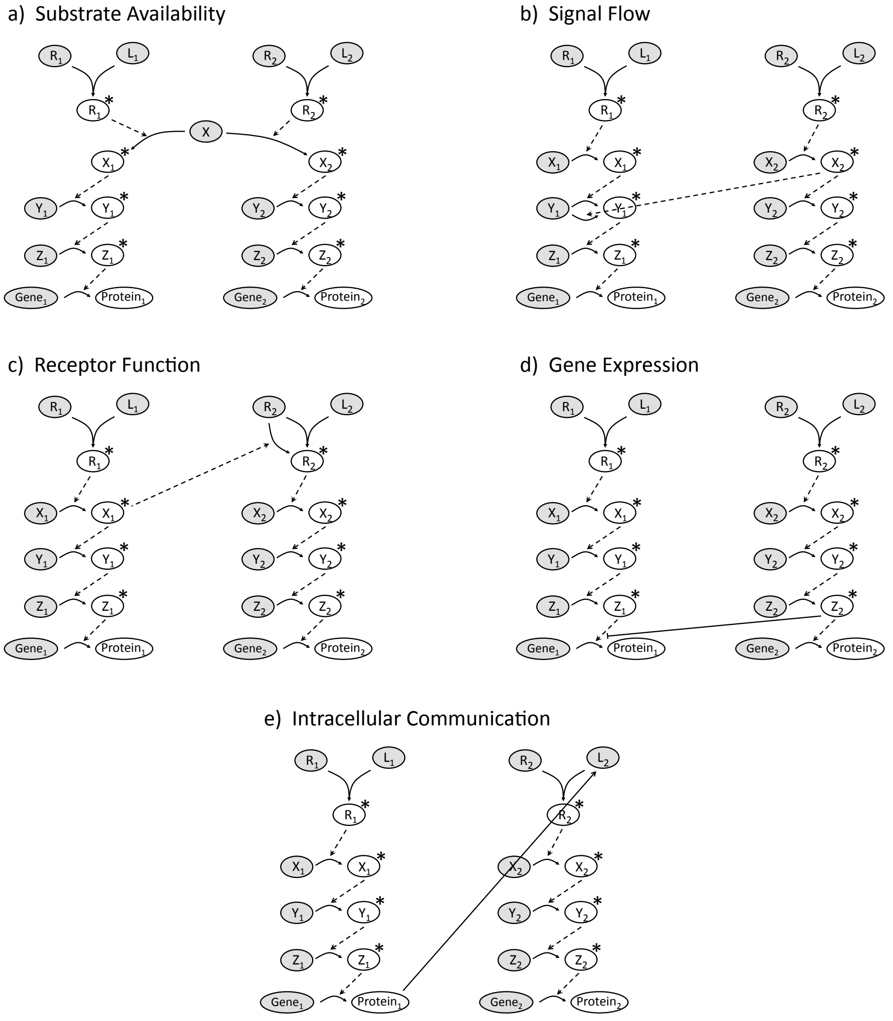
$Network_i = Network_{i-1}\ |[E_{i-1}\ \cup\ U_{i-1},\ V_{i-1}]|\ Pathway_i$

However, to the best of our knowledge all cross-talk are between pairs of pathways. Our definition of a network allows a single cross-talk in which three or more pathways participate, however no biological examples of such an interaction have been reported.

This concludes modelling cross-talk. We now switch our focus from defining and categorising cross-talk in a rigorous way to analysing models of cross-talk using logical properties.

8. Detecting cross-talk

So far we have discussed the main contribution of this paper, how to model cross-talk in a rigorous way by looking at the form of the model description, e.g. the synchronisations between PRISM modules. We now give preliminary results on the complementary problem, how to analyse at the model level (i.e.



a) Substrate Availability

b) Signal Flow

c) Receptor Function

d) Gene Expression

e) Intracellular Communication

Figure 8: An example of each of the five types of cross-talk: a) two pathways compete for a protein, b) a pathway up-regulates signal flow through another pathway, c) a pathway activates the receptor of another pathway in the absence of a ligand, d) two pathways have conflicting transcriptional responses, e) a pathway releases a ligand for another pathway.

at the level of the CTMC rather than the form of the model description[5]). We aim to both detect and characterise cross-talk, we first tackle detecting cross-talk in these models.

This section makes use of the two pathways $Pathway_1$ and $Pathway_2$ introduced in Section 4 and the five example cross-talk models of Section 7.5.

The presence of cross-talk can be detected by checking a set of temporal logic properties as follows.

We choose CSL because we need a quantitative logic – it is a change in the probability of a formula being true that allows us to detect the presence of a cross-talk. The probabilities are used to measure the number of paths that satisfy a property. For example, in the signal flow cross-talk example (compared to the independent model) there is a greater number of paths to the expression of $Protein_1$. There are other ways to detect cross-talk, however we use model checking of CSL properties as it is relatively straightforward and familiar to a large part of the community.

Given pathways $Pathway_1$ and $Pathway_2$ that conclude with gene expression ($Protein_1$ and $Protein_2$ being produced respectively), we detect cross-talk by comparing the probabilities of the following three CSL formulae with the probability of independent composition. Namely, we compare probabilities for the five example cross-talk models of Section 7.5 with $Pathway_1 \; \{renamings_1\} \; |[E \cup U, \; V]| \; Pathway_2 \; \{renamings_2\}$ where $E = \emptyset$ and $V = \emptyset$. In each case, cross-talk is indicated by a change of probability of at least one formula.

Competitive Signal Flow ($Pathway_1$ before $Pathway_2$): probability of signal flow through $Pathway_1$ before $Pathway_2$

$$\psi_1 \;=\; \mathbf{P}_{=?} \; [\; F \; (Protein_1 \;=\; 1 \;\wedge\; Protein_2 \;=\; 0) \;]$$

Independent Signal Flow ($Pathway_1$): probability of signal flow through $Pathway_1$ within a time bound (3 time units)

$$\psi_2 \;=\; \mathbf{P}_{=?} \; [\; F_{\leq 3} \; (Protein_1 \;=\; 1) \;]$$

Independent Signal Flow ($Pathway_2$): probability of signal flow through $Pathway_2$ within a time bound (3 time units)

[5] for example, we define CTMCs in PRISM using the PRISM *language* whereas in a tool like Matlab, we would define them with *equations*

	ψ_1	ψ_2	ψ_3
Substrate Availability Example	$=$	$\downarrow$	$\downarrow$
Signal Flow Example	$\uparrow$	$\uparrow$	$=$
Receptor Function Example	$\downarrow$	$=$	$\uparrow$
Gene Expression Example	$\downarrow$	$\downarrow$	$=$
Intracellular Communication Example	$=$	$=$	$=$

Table 1: The change in probability for each of the 5 cross-talk models compared with the independent pathways model for the three CSL properties.

$$\psi_3 \ = \ \mathbf{P}_{=?} \ [\ F_{\leq 3} \ (Protein_2 \ = \ 1) \ ]$$

The change in probability for each of the 5 cross-talk models, as compared to the independent pathways, is given in Table 1. ↑ denotes an increase, ↓ denotes a decrease and = denotes no change in probability. Results were obtained using the PRISM model checker (run times are negligible).

Notice that there is no change in probability for the intracellular communication cross-talk model. In our qualitative model of this cross-talk, one pathway produces a ligand for another pathway only after the original ligand molecule has been consumed in a reaction. This means that the cross-talk has no effect on the rate of the activation reactions in either pathway. In a model with a greater level of quantitative detail, as discussed in Section 11, this cross-talk would change the rate of the activation reactions. This result is not unexpected as we have already identified that intracellular communication cross-talk is a source of contention in the literature.

We now move on to characterising cross-talk in models in which there is no model description.

9. Characterising cross-talk

The type of cross-talk can be characterised at the model level using different temporal logic properties.

We choose CTL because we need a qualitative logic – it is a difference in the structure of the Markov chain rather than the transition rates that allows us to distinguish between types of cross-talks. We define 5 CTL properties, each of which characterises a type of cross-talk. The properties are simple liveness or safety properties and do not exploit the rate information in the model.

As before, the activation of a pathway is reflected by the expression of $Protein$. This is checked by evaluating the proposition ($Protein = 1$).

Substrate Availability Example ($Pathway_1$ *and* $Pathway_2$ *compete for a protein).* It is not possible to activate X in both pathways (i.e. the pathways compete for a limited protein).

$\mathbf{A}\ G \neg (X_1^* = 1 \wedge X_2^* = 1)$

Signal Flow Example (flow from $Pathway_2$ *to* $Pathway_1$*).* It is possible to activate $Pathway_1$ without activating receptor R_1.

$\mathbf{E}\ F\ (R_1^* = 0 \wedge Protein_1 = 1)$

Receptor Function Example ($Pathway_1$ *activates* $Pathway_2$*'s receptor).* It is possible to activate the receptor R_2 without using the ligand L_2.[6]

$\mathbf{E}\ F\ (R_1^* = 0 \wedge R_2^* = 1 \wedge L_2 = 1)$

Gene Expression Example ($Pathway_2$ *inhibits* $Pathway_1$*'s gene expression).* It is not possible to activate $Pathway_1$ if the signal has already passed through $Pathway_2$.

$\mathbf{A}\ G \neg (Protein_1 = 1)\ \{Y_1^* = 1 \wedge Z_1^* = 0 \wedge Protein_2^* = 1\}$

Intracellular Communication Example ($Pathway_1$ *expresses* $Pathway_2$*'s ligand).* It is possible to use and replenish ligand L_2.

$\mathbf{E}\ (L_2 = 1)\ U\ (\ (L_2 = 0) \wedge \mathbf{E}\ (L_2 = 0)\ U\ (L_2 = 1)\ )$

We now demonstrate our approach on a prominent case-study of the cross-talk between the TGF-β, WNT and MAPK pathways.

10. Case-study

We apply our approach to a prominent biological case-study of the cross-talk between the TGF-β, WNT and MAPK pathways. Details are taken from [21] and from discussions with a domain expert [25]. We use the approach to classify the cross-talk in the model and to understand the effects of the cross-talk on the TGF-β pathway. We note that the effects of cross-talk are not discussed in [21].

[6] We include $R_1^* = 0$ because signalling in $Pathway_1$ in the intracellular communication model can produce L_2.

Our model of the pathways and their cross-talk is shown in Figure 9. To apply our modelling approach we need to expand our set of modules to: Receptor, Protein Activation, 2-stage Cascade, 3-stage Cascade, Translocation, Protein Binding and Gene Expression. This is a natural extension of our approach. The extra modules act in a similar manner to the modules that have been discussed and so the extension of the formalisation from Section 7.3 is trivial.

We define the following three pathways (for brevity, we omit the synchronisation sets and renamings).

$$\begin{aligned} MAPK &= (\mathit{Receptor}\ \{\ldots\}\ |[\ldots]|\ \mathit{ProteinActivation}\ \{\ldots\}\ |[\ldots]| \\ &\quad\ \ \mathit{Cascade}3\ \{\ldots\}\ |[\ldots]|\ \mathit{Cascade}2\ \{\ldots\}\ |[\ldots]| \\ &\quad\ \ \mathit{Translocation}\ \{\ldots\}\ |[\ldots]|\ \mathit{Translocation}) \\ TGFB &= (\mathit{Receptor}\ \{\ldots\}\ |[\ldots]|\ \mathit{ProteinActivation}\ \{\ldots\}\ |[\ldots]| \\ &\quad\ \ \mathit{ProteinActivation}\ \{\ldots\}\ |[\ldots]|\ \mathit{ProteinBinding}\ \{\ldots\} \\ &\quad\ \ |[\ldots]|\ \mathit{Translocation}\ \{\ldots\}\ |[\ldots]|\ \mathit{GeneExpression}\ \{\ldots\} \\ &\quad\ \ |[\ldots]|\ \mathit{GeneExpression}) \\ WNT &= (\mathit{Receptor}\ \{\ldots\}\ |[\ldots]|\ \mathit{ProteinActivation}\ \{\ldots\}\ |[\ldots]| \\ &\quad\ \ \mathit{Translocation}\ \{\ldots\}\ |[\ldots]|\ \mathit{GeneExpression}) \end{aligned}$$

As indicated in Figure 9, the following auxiliary reactions are added. In the MAPK pathway we add catalysis auxiliary reactions to the MAPK$*$, AKT$*$ and TF$*$ species. In the TGF-β pathway we add degradation auxiliary reactions to the Smad4 and Smad7 species. In the WNT pathway we add catalysis auxiliary reactions to the Axin and β-Catenin$*$ species.

We then consider four networks, $TGFB$, $TGFB\ |[\ldots]|\ MAPK$, $TGFB\ |[\ldots]|\ WNT$ and $(TGFB\ |[\ldots]|\ MAPK)\ |[\ldots]|\ WNT$, referred to as the full network.

We detect the presence of 9 cross-talks in the full network using the approach outlined in Section 9 – no new cross-talks are identified compared with the literature. We then characterise each cross-talk using the approach outlined in Section 8 and find that there are three types of cross-talk in the model, as follows.

We measure the output of the TGF-β pathway by the activity of the expression of *Proteins* (a set of proteins in the TGF-β pathway). We use the following CSL properties to compare the effects of cross-talk: ψ_1, the eventual expression of Proteins, and ψ_2, the time-dependent expression of Proteins (within 5 time units).

$$\begin{aligned} \psi_1 &= \mathbf{P}_{=?}\ [\ F\ (\mathit{Proteins}\ =\ 1)\ ] \\ \psi_2 &= \mathbf{P}_{=?}\ [\ F_{\leq 5}\ (\mathit{Proteins}\ =\ 1)\ ] \end{aligned}$$

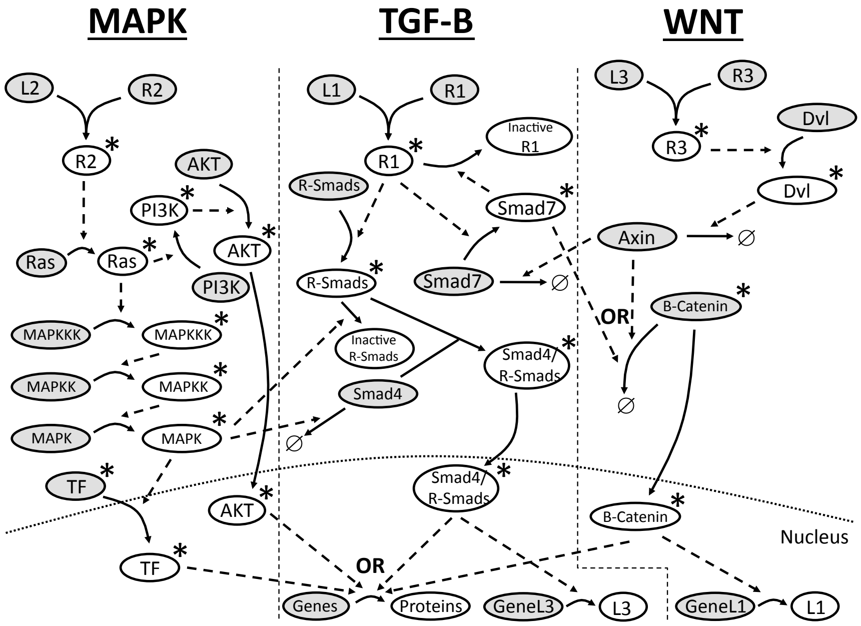


Figure 9: Cross-talk between the TGF-β, WNT and MAPK pathways. Species that are present in the initial state are denoted by shaded ellipses.

We follow with a detailed analysis of each of the four networks.

Independent Network. In the independent network, $TGFB$, the activation of the TGF-β pathway leads to the expression of Proteins within 5 time units, ψ_2, with probability 0.47 and to the eventual expression of Proteins, ψ_1, with probability < 1 due to the inactivation of the receptor.

TGF-β and MAPK Cross-talk. In the TGF-β and MAPK network, $TGFB$ $|[\ldots]|$ $MAPK$, there are two types of cross-talk. *Signal flow:* MAPK* proteins slow signal flow through the TGF-β pathway by deactivating the R-Smads and degrading Smad4. *Gene expression:* the TF* and AKT* proteins upregulate gene expression in the TGF-β pathway. Note that the appearance of the AKT and PI3K proteins in the MAPK pathway indicates an implicit cross-talk with the AKT and PI3K pathways respectively. The inclusion of cross-talk with the MAPK pathway can both provide alternative gene expression reactions and block signal flow through the TGF-β pathway, overall causing the probability of the expression of Proteins within 5 time units, ψ_2, to increase to 0.73. The probability of the eventual expression of Proteins, ψ_1, is 1 because the reactions in the MAPK pathway are not inhibited in the model.

TGF-β and WNT Cross-talk. In the TGF-β and MAPK network, $TGFB$ $|[\ldots]|$ WNT, there are three types of cross-talk. *Signal flow:* the Smad7* protein degrades β-Catenin and the Axin protein degrades Smad7. *Gene expression:* the β-Catenin protein upregulates gene expression in the TGF-β pathway. *Intracellular communication:* the WNT pathway can cause the production of a ligand for the TGF-β pathway, and vice-versa. The inclusion of cross-talk with the WNT pathway can both provide an alternative gene expression reaction and inhibit Smad7 which can inactivate the receptor for the TGF-β pathway. Overall this causes the probability of the expression of Proteins within 5 time units, ψ_2, to increase to 0.76. The probability of the eventual expression of Proteins, ψ_1, is still < 1 due to the degradation of the β-Catenin protein.

TGF-β, WNT and MAPK Cross-talk. The TGF-β, WNT and MAPK network, $(TGFB\ |[\ldots]|\ MAPK)\ |[\ldots]|\ WNT$, is the union of the two cross-talk scenarios above. The effect of both WNT and MAPK cross-talk to the TGF-β pathway is additive. The probability of ψ_2 rises to 0.88, compared with the single cross-talks of WNT and MAPK with probability 0.76 and 0.73 respectively. The inclusion of the MAPK cross-talk provides sets of reactions that cause gene expression which cannot be inhibited and hence the probability of ψ_1 is 1.

We remark that we have categorised the complicated cross-talk in which Axin degrades Smad7 unambiguously as signal flow. Whereas, in [21] there is a suggestion that the cross-talk is receptor function because Axin degrades the receptor (via Smad7, an intermediate). However, our approach does not classify this cross-talk as receptor function cross-talk.

11. Discussion

Reversible reactions. We have simplified the biochemistry in this paper by only considering irreversible reactions, e.g. the activation reaction $X \rightarrow X^*$. If our models were to include deactivation reactions, e.g. $X^* \rightarrow X$, then the temporal logic properties would need to be strengthened. For example, the property characterising signal flow cross-talk expresses that at some point in time R_1 is inactive and $Protein_1$ is expressed. If the activation of R_1 is a reversible reaction then this property is too weak. The property could be satisfied if R_1 becomes active, $Protein_1$ is expressed and then R_1 becomes inactive. Thus, the correct property with reversible reactions is:

$\mathbf{E}\ [(R_1^* = 0)\ U\ (R_1^* = 0 \wedge Protein_1 = 1)]$.

Cross-talk formalisation. Our cross-talk formalisation depends on the set of modules being considered. One reason for this is that the modules act as a proxy for the cellular location. For example, in the definition for Gene Expression cross-talk, we disallow reactions from the Receptor module because gene expression occurs in the nucleus which is 'far' from the receptor. Future work will be to introduce a formalisation that is not so strongly tied to current set of modules. We expect to need, at the very least, a mapping from the set of modules to the location of the modules.

Cross-talk generation and pathway generation. Our method to generate all cross-talk models from a set of pathways (see Appendix A) could also be applied to generate all pathway models from a set of modules. However, to generate a pathway from a set of modules we would need to be careful that all modules are connected and sometimes connected together in a specific manner. Therefore, we would require a constraint to our method: a set of reactions in each module that must synchronise with at least one reaction in another module.

Quantitative detail. We have demonstrated our approach on models with a low level of quantitative detail. As such, the probability values resulting from CSL model checking can only be used to compare the models with each other. However, with more quantitative detail, further interpretation of our analysis results would be possible. For example, the properties concerning the probability of time-dependent gene expression between cross-talk models would become a meaningful assessment of the strength of the cross-talk.

Model-checking runtimes. The state spaces for all the models presented here are small, e.g. of the order of 10^2. Runtimes for checking properties are therefore trivial.

Feature interaction. There is an interesting analogy with feature interactions in telecommunications and software systems. Features, or services, in these systems are functionality additional to the core, added incrementally, by various developers, at various times (e.g. due to market deregulation). One consequence of these uncoordinated additions is interactions between the new features themselves, or with the core system, causing some features or the core to behave in new, sometimes undesirable ways. One inspiration for the approach to pathway cross-talk presented here is work on using temporal logics to detect and characterise feature interactions [26]. An open question is whether techniques developed to model and detect features and interactions may be applicable to pathway cross-talk. For example, a common problem is lack of universal definition of pathway/feature; it would be interesting to investigate if concepts such as the feature construct of [27] would be useful in the pathway paradigm. Finally, we note that in telecommunications, 3-way feature interactions (a interaction between three features, that does not occur between only two features) are very rare: most detection algorithms depend on a pairwise analysis. This parallels cross-talk in which we have not found a single example of cross-talk that is between three pathways.

Cross-talk categorisation. An interesting question that also plagues the feature interaction community is what is a feature and a feature interaction? This is analogous to what is a pathway and a cross-talk, which begs the question, is our cross-talk categorisation complete? We recall discussions with a domain expert [25] that suggest a pathway is a *dominant* behaviour whereas cross-talk is a side effect, which leads us to believe this is *future* work for the biological, rather than the formal computer science, community.

The Molecular Nose research project. This work has been developed as part of the Molecular Nose research project (see *Acknowledgements*). The project aims to develop new in vivo sensor technologies for analysing and interpreting cellular signal transduction networks. The term “molecular nose” refers to sensor technology “sniffing out” pathways within a cell. Long term, we aim to generate hypotheses about the structure of pathways and networks that may explain cross-talks or pathway structures, comparing those in normal cells with the same in diseased cells.

12. Conclusions and future work

We have defined a rigorous approach to modular modelling of pathways and their cross-talk, based on generic modules and composition with synchronisation, variable sharing, and reaction renaming.

Our modelling style is reagent-centric, implemented in a state-based language, which means that every reagent is mapped to a variable that denotes a concentration level in an underlying CTMC with levels. While in standard reagent-centric modelling modules represent processes, here we define generic modules that represent commonly occurring behavioural patterns in signalling pathways. We then compose instances of modules, using the standard (synchronising) parallel composition operator, to define a pathway. Since composition is associative, this extends to pathways composed of any number of modules.

A key aspect of a pathway is the distinction between external reactions and variables, that are visible in the interface, and internal reactions and variables, that are not visible. We then compose instances of pathways, again using the standard (synchronising) parallel composition operator, to define a network of pathways. If the intersection of the interfaces of the two pathways is empty, then the two pathways are considered independent. The pathways can be "wired" together, i.e. to make the pathways cross-talk, in a combinatorial manner by renaming and synchronising external reactions and by sharing variables. Again, by associativity, this extends to networks of any number of pathways. Also, it is easy to remove any results that are clearly biochemically infeasible. This approach allows us to both explore all possible cross-talks between pathways and to explain a given network in terms of pathways.

We have defined 5 different types of cross-talk, these have been alluded to in the literature but not previously defined formally. Generation of the different types of cross-talk is given by two simple algorithms; detection and characterisation of the different types of cross-talk is by qualitative and quantitative logic property model checking. We are able to show, from a basic set of modules, that every type of cross-talk can be generated and detected. Throughout, we use the state based PRISM modelling language, a language of guarded commands with process algebraic composition, with two minor extensions.

We apply the approach to a prominent example of cross-talk between the three pathways: TGF-β, WNT and MAPK. We both detect and categorise 9 cross-talks.

Several future directions have been identified. As suggested earlier, longer term, we will apply our approach to models with a higher level of quantitative detail: to make better predictions and gain further insights into the biological effects of cross-talk. We will also consider the relationship between cross-talks and standard pathway motifs. For example, is there a reasonable alteration of a pathway model (e.g. addition of a feedback loop) that gives the same behaviour as a potential cross-talk? We will also assess how the effectiveness of pathway intervention techniques such as drugs and gene knockouts change with the addition of cross-talk. Finally, a larger question is how the temporal ordering of signals affects the detectability and behaviour of cross-talk; for example, do pathways hold a "biochemical history" of signalling events?

The cross-talk generation program (incl. source code), PRISM models and logic properties used in this paper can be found at:
`www.dcs.gla.ac.uk/~radonald/jtb2011/`.

Acknowledgements. This research is supported by the project *The Molecular Nose*, funded by the Engineering and Physical Sciences Research Council (EPSRC). We thank Professor Rainer Breitling for advice on the development of the case-study model.

References

[1] V. Danos, J. Feret, W. Fontana, R. Harmer, J. Krivine, Rule-based modelling of cellular signalling, invited paper, in: L. Caires, V. Vasconcelos (Eds.), Proceedings of the Eighteenth International Conference on Concurrency Theory, CONCUR '2007, Lisbon, Portugal, volume 4703 of *Lecture Notes in Computer Science*, Springer, Berlin, Germany, Lisbon, Portugal, 2007, pp. 17–41.

[2] M. Heiner, D. Gilbert, R. Donaldson, Petri Nets in Systems and Synthetic Biology, in: Schools on Formal Methods (SFM), Springer LNCS 5016, 2008, pp. 215–264.

[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 VII 4230 (2006) 1–23.

[4] M. Calder, V. Vyshemirsky, R. Orton, D. Gilbert, Analysis of Signalling Pathways using Continuous Time Markov Chains, Trans. on Computat. Syst. Biol. VI 4220 (2006) 44–67.

[5] O. Tymchyshyn, M. Kwiatkowska, Combining intra- and inter-cellular dynamics to investigate intestinal homeostasis, in: Proc. Formal Meth-

ods in Systems Biology (FMSB'08), volume 5054 of *LNCS*, Springer, 2008.

[6] P. Degano, D. Prandi, C. Priami, P. Quaglia, Beta-binders for Biological Quantitative Experiments, Electronic Notes in Theoretical Computer Science 164 (2006) 101–117.

[7] N. Chabrier-rivier, M. Chiaverini, V. Danos, F. Fages, V. Schächter, Modeling and querying biomolecular interaction networks, Theoretical Computer Science 325 (2003) 25–44.

[8] C. M. Taniguchi, B. Emanuelli, R. C. Kahn, Critical nodes in signalling pathways: insights into insulin action, Nat Rev Mol Cell Biol 7 (2006) 85–96.

[9] I. Catt, Crosstalk (Noise) in Digital Systems, Electronic Computers, IEEE Transactions on EC-16 (1967) 743–763.

[10] M. A. Schwartz, M. H. Ginsberg, Networks and crosstalk: integrin signalling spreads, Nat Cell Biol 4 (2002) E65–E68.

[11] M. Hatakeyama, S. Kimura, T. Naka, T. Kawasaki, N. Yumoto, M. Ichikawa, J. Kim, K. Saito, M. Saeki, M. Shirouzu, S. Yokoyama, A. Konagaya, A computational model on the modulation of Mitogen-Activated Protein Kinase (MAPK) and Akt pathways in heregulin-induced ErbB signalling, Biochemical Journal 373 Pt. 2 (2003) 451–463.

[12] S. N. Sreenath, R. Soebiyanto, M. D. Mesarovic, O. Wolkenhauer, Co-ordination of crosstalk between MAPK-PKC pathways: an exploratory study, Systems Biology, IET 1 (2007) 33–40.

[13] M. N. N. McClean, A. Mody, J. R. R. Broach, S. Ramanathan, Cross-talk and decision making in MAP kinase pathways, Nat Genet (2007).

[14] M. Heiner, I. Koch, J. Will, Model validation of biological pathways using Petri nets–demonstrated for apoptosis, Journal BioSystems 75 (2004) 15–28.

[15] J. Fisher, N. Piterman, A. Hajnal, T. A. Henzinger, Predictive Modeling of Signaling Crosstalk during C. elegans Vulval Development, PLoS Computational Biology 3 (2007) e92+.

[16] R. Donaldson, M. Calder, Modelling and Analysis of Biochemical Signalling Pathway Cross-talk, EPTCS 19 (2010) 40–54.

[17] M. Calder, J. Hillston, Process algebra modelling styles for biomolecular processes, Trans. on Computat. Syst. Biology XI, LNBI 5750 (2009) 1–25.

[18] F. Ciocchetta, A. Degasperi, J. Hillston, M. Calder, Some Investigations Concerning the CTMC and the ODE Model Derived From Bio-PEPA, Electron. Notes Theor. Comput. Sci. 229 (2009) 145–163.

[19] R. Alur, T. Henzinger, Reactive Modules, Formal Methods in System Design 15 (1999) 7–48.

[20] PRISM Website, PRISM - Probabilistic Symbolic Model Checker, `http://www.prismmodelchecker.org`, (2011).

[21] X. Guo, X.-F. Wang, Signaling cross-talk between TGF-β/BMP and other pathways, Cell Research 19 (2009) 71–88.

[22] NIH, “Insulin Signaling And Receptor Cross-Talk” Program Announcement PA-07-058, (2006).

[23] B. Katzenellenbogen, Estrogen receptors: bioactivities and interactions with cell signaling pathways, Biol. Reprod. 54 (1996) 287–293.

[24] K. D. Bosscher, W. V. Berghe, G. Haegeman, Cross-talk between nuclear receptors and nuclear factor κb, Oncogene 25 (2006) 6868–6886.

[25] R. Breitling, Private correspondence, (2009).

[26] M. Calder, A. Miller, Feature interaction detection by pairwise analysis of LTL properties, Formal Methods in System Design 28 (2006) 213–261.

[27] M. Plath, M. Ryan, The feature construct for SMV: Semantics, Feature Interactions in Telecommunications and Software Systems VI (2000) 129–144.

Appendix A. Cross-talk generation

We describe below a method to generate every cross-talk between two pathways P_1 and P_2. First, we consider substrate availability cross-talk and then, all other types of cross-talk.

To generate every substrate availability cross-talk we share between pathways every pair of (external) variables.

```
for variable $v \in ext_v(P_1)$ do
  for variable $w \in ext_v(P_2)$ do
    $P_1 \ \{renamings_1\} \ |[E \cup U, \ V]| \ Pathway_2 \ \{renamings_2\}$ where $U = aux(P_1) \cup aux(P_2)$, $E = \emptyset$, $renamings_1 = \emptyset$, $renamings_2 = \emptyset$ and $V = \{(v, w)\}$
  end for
end for
```

To generate every cross-talk of all other types we create all possible *candidate* cross-talks by synchronising up to k (external) reactions.

```
for $i \geq 1, j \geq 1$ such that $i + j \leq k$ do
  for $X$ = choose $i$ reactions from $ext_r(P_1)$ do
    for $Y$ = choose $j$ reactions from $ext_r(P_2)$ do
      if $X \cup Y$ contains only modifiers then skip else
      $P_1 \ \{renamings_1\} \ |[E \cup U, \ V]| \ P_2 \ \{renamings_2\}$ where
      $renamings_1$ such that $\forall x \in X.x \leftarrow r_{new}$, $renamings_2$
      such that $\forall y \in Y.y \leftarrow r_{new}$, $E = \{r_{new}\}$ and
      $U = (aux(P_1 \ \{renamings_1\}) \cup aux(P_2 \ \{renamings_2\})) \setminus mapped(r_{new})$,
    end for
  end for
end for
```

We then filter the candidate cross-talks according to the categorisation – those cross-talks that are not categorised are removed.