Quantitative and Probabilistic Modeling in Pathway Logic

Alessandro Abate
EECS Department, UC Berkeley
aabate@eecs.berkeley.edu

Yu Bai
Biophysics Program, Stanford University
yubai@stanford.edu

Nathalie Sznajder
École Normale Superiéure
sznajder@lsv.ens-cachan.fr

Carolyn Talcott
CS Lab, SRI International
clt@csl.sri.com

Ashish Tiwari
CS Lab, SRI International
tiwari@csl.sri.com

Abstract

This paper presents a study of possible extensions of Pathway Logic to represent and reason about semiquantitative and probabilistic aspects of biological processes. The underlying theme is the annotation of reaction rules with affinity information that can be used in different simulation strategies. Several such strategies were implemented, and experiments carried out to test feasibility, and to compare results of different approaches. Dimerization in the ErbB signalling network, important in cancer biology, was used as a test case.

1 Introduction

Biological networks have complex interconnections, non-linear responses to stimuli and self-regulation. This presents clear challenges for modeling and studying their behavior, and it is important to efficiently organize and represent knowledge about biological networks at the modeling stage. Pathway Logic [6, 27] is an approach to modeling cellular processes based on symbolic logic. It allows one to model aspects of the structure and state of interacting components, to represent individual process steps (reactions) and to study possible ways a system can evolve using techniques based on logical inference. Reactions can be modeled at many levels of detail ranging from micro steps representing events such as phosphorlyation at specific sites or binding of protein domains, to macro steps such as the results of signaling or metabolic modules. The choice of the level of detail depends both on available data and the questions to be asked. This flexible approach allows one to study reaction networks of hundreds or even thousands of nodes.

Although detailed data concerning reaction rates is often still limited, there is much more data concerning time series and overall effects of changes in cellular signals and expression levels of different cellular components. Thus there is increasing interest in developing simulation and verification tools to handle quantitative, or at least semi-quantitative data. Moreover the uncertainty that affects the study of the entities into play suggests that it is important to address probabilistic aspects of biochemical processes. Thus one would like to ask a model semi-quantitative questions about different possible outcomes under different initial conditions or due to perturbations of ongoing processes, without sacrificing the ability to scale to moderately complex processes. For example, one might ask about the relative amounts of different phenotypes related to an overexpressed gene; or how the outcomes change when the network is perturbed, say by mutations or blocking activity of particular components.

In this paper we report on a study of different approaches to represent probabilistic information about approximate quantities and rates as a first step to extending the Pathway Logic modeling framework. The underlying theme is the use of probabilities and stochastic modeling as a flexible technique to account for unknown features and to incorporate different levels of quantitative information. Questions of interest include: how do different mathematical models represent rates and quantities? What are good choices for modeling randomness in these networks? What simplifications and abstractions are meaningful? Which techniques have efficient implementations that can scale to moderately complex modules?

We considered different but related approaches from the literature to represent probabilities and random events: stochastic simulations of chemically reacting systems, stochastic Petri nets and probabilistic boolean networks. We reinterpreted and customized these different techniques to the framework of Pathway Logic and several variations of these approaches were implemented and compared. Experiments were carried out using the ErbB dimerization network as a testbed. A first simple approach based on prioritizing rules and using a greedy discrete algorithm for simulation was developed and tested on a model of dimerization and activation for four ErbBs. The greedy algorithm was compared to an analysis using a model-checker for probabilistic systems. The feasibility of predicting the final/equilibrium state for a smaller model of dimerization of two ErbBs was first tested using a probability-based-rule-sampling approach, programmed in Matlab. Then, a probabilistic extension of the rewriting semantics underlying Pathway Logic was implemented in the Maude rewriting logic language and an extension of this small ErbB dimerization network with rules for internalization and degradation of the ErbBs was studied and compared to a previously published model.

2 Symbolic modeling and Pathway Logic

Symbolic modeling of cellular processes. Symbolic/logical models allow one to represent partial information and to model and analyze systems at multiple levels of detail, depending on information available and questions to be studied. Such models are based on formalisms that provide a language for representing the states of the system, mechanisms to model their changes, such as reactions, and tools for analysis based on computational or logical inference. Symbolic models can be used for simulation of system behavior. In addition properties of processes can be stated in associated logical languages and checked using tools for formal analysis. A variety of formalisms have been used to develop symbolic models of biological systems, including Petri nets [10, 20]; ambient/membrane calculi [19, 22]; statecharts [5]; live sequence charts [16]; and rule-based systems including P-systems [21]; and Pathway Logic [6, 27]. Each of the underlying formalisms was initially developed to model and analyze computer systems with multiple processes executing concurrently. A pi-calculus model for the receptor tyrosine kinase/mitogen-activated protein kinase (RTK/-MAPK) signal transduction pathway is presented in [24]. Tools such as BioSPI [23] and SPiM [3] have been developed for Monte Carlo simulation to obtain time-evolution of molecular concentrations. In addition to simulation, probablistic model checking techniques using tools such as PRISM [17] have been used to analyze such models [1, 11] A simple formalism for representing interaction networks using an algebraic rule-based approach very similar to the Pathway Logic approach is presented in [7]. The language has three interpretations: a qualitative binary interpretation much like the Pathway Logic models; a quantitative interpretation in which concentrations and reaction rates are used; and a stochastic interpretation. Queries are expressed in a formal logic called Computation Tree Logic (CTL) and its extensions to model time and quantities. CTL queries can express reachability (find pathways having desired properties), stability, and periodicity. Techniques for learning new rules to achieve a desired system specification are described in [2].

Pathway Logic. In Pathway Logic (PL) models of biological processes are developed using the Maude system [4] a formal language and tool set based on rewriting logic. Rewriting logic [18] is a logical formalism that is based on two simple ideas: states of a system are represented as elements of an algebraic data type; and the behavior of a system is given by local transitions between states described by *rewrite rules*. The process of application of rewrite rules generates computations (also thought of as deductions). In the case of biological processes these correspond to pathways.

A PL model includes representation of cellular components such as proteins and small molecules, their locations, protein state, and post translational modifications. It also includes representations, as rewrite rules, of basic process steps such as metabolic reactions or intra- and inter- cellular signaling. Execution of the rules allows one to represent and reason about dynamic assembly of complexes, cascading transmission of signals, feedback-loops, cross talk between subsystems, and larger pathways. Pathways are not predefined. Instead they are assembled by instantiating and connecting individual steps, starting from an initial state, subject to user-defined constraints. PL models are transformed into Petri nets for visualization and analysis using the Pathway Logic Assistant [26], a tool for interactive visualization and analysis of PL models.

In the following we use the EGFR family of receptor tyrosine kinases (ErbBs), important in the study of cancer tumor cells, as the basis of our case studies. These receptors form a multiplicity of homo- and hetero-dimers [28]. As an example, the following are rules (in simplified form, represented using Maude syntax [4]) for the homo- and hetero-dimerization of two receptors `ErbB1` (also known as `EGFR`) and `ErbB2` (also known as `HER2`).

```
rl[r1]: ErbB1 ErbB1 => ErbB1:ErbB1
rl[r2]: ErbB1 ErbB2 => ErbB1:ErbB2
rl[r3]: ErbB2 ErbB2 => ErbB2:ErbB2
```

The first rule (labeled `r1`) says that if `ErbB1` is present in the system in multiple copies, then two can bind together to form a homo-dimer `ErbB1:ErbB1`. When this rule fires two occurrences of `ErbB1` are removed from the state and `ErbB1:ErbB1` is added. The second rule describes the hetero-dimerization of `ErbB1` and `ErbB2`, and the third rule describes homo-dimerization of `ErbB2`.

Sample PL models, tutorial material, papers and presentations are available from the PL web site, `http://pl.csl.sri.com/`, along with the Pathway Logic Assistant [26].

3 Prioritized rule modeling of ErbBs

We started with a very simple idea for semi-quantitative reasoning, namely to assign priorities to rules. The priorities can be thought of as affinities or an abstraction of the thermodynamics of the system. We used these priorities in two ways: as parameters to a greedy algorithm for choosing which rule to fire next; and as parameters of a probabilistic model. This idea was tested on a simplified set of rules for the four members of the ErbB family of receptors. These rules model the (homo- and hetero-) dimerization and resulting cross phosphorylation steps, assuming the receptors that need ligands are initially ligand bound.

```
rl[1]: E2 E3B => E3Bp E2d      rl[5]: E4B E4B => E4Bp E4Bp
rl[2]: E2 E1B => E2p E1Bp      rl[6]: E1B E3B => E3Bp E1Bd
rl[3]: E2 E4B => E2p E4Bp      rl[7]: E1B E4B => E1Bp E4Bp
rl[4]: E1B E1B => E1Bp E1Bp    rl[8]: E3B E4B => E3Bp E4Bd
```

In these rules `E1B` represents ligand bound ErbB1, and similarly for `E3B` and `E4B`. `E2` represents ErbB2 which has no ligand. `E1Bp` is bound phosphorylated ErbB1, implicitly dimerized with its phosphorylation partner, and `E1Bd` represents ErbB1 that is dimerized but not phosphorylated. Similarly, for the other ErbBs. The rules express the known biochemistry of the ErbB dimers. In particular ErbB3 has no kinase activity and thus can not cross phosphorylate its dimerization partner. Also rules for homo-dimerization of ErbB2 and ErbB3 have been omitted. For simulation purposes an execution state is a set of pairs `(n, e)`, where `e` is one of the ErbB symbols and `n` is the number of molecules of `e`. In the following subsections we explain the two uses of priorities, give results from some test cases, and compare the two methods.

Greedy algorithm. The greedy algorithm for using the priorities is the following: all rules of highest priority are applied until none can be applied. Then rules of the next highest priority are considered, until all of the priority levels are exhausted. Note that because there are no cycles (no rule produces something another rule can use) if a rule is applied as much as possible, application of other rules will not result in a state where the rule is again applicable.

The table below summarizes the results of 4 test cases: two starting states and two assignments of rule priorities. The two starting states are `((10000, E1B) (100000, E2) (100000, E3B) (n, E4B))`, where `n` is 0 or 10000. The first assignment gives all rules have same priority and second assignment is the following

```
rl[1] -> 1, rl[2] -> 2, rl[3] -> 2, rl[4] -> 3,
rl[5] -> 3, rl[6] -> 4, rl[7] -> 4, rl[8] -> 4.
```

where lower numbers correspond to higher probability, reflecting the experimental observation that ErbB2 is the preferred dimerization partner of the ErbBs.

The row labels code the test case, `eq` is the same priority case and `neq` is the varied priority case. The `+/-` corresponds to presence or absence of ErbB4 in the initial state.

	E1Bd	E1Bp	E2d	E2p	E3Bp	E4Bd	E4Bp
eq-	3334	6668	96666	3334	100000	0	0
eq+	2500	7500	95000	5000	100000	2500	7500
neq-	0	10000	100000	0	100000	0	0
neq+	0	10000	100000	0	100000	0	10000

With equal priority rules the presence of ErbB4 effects the outcome for others by competing with ErbB3 for dimerization resulting in more phosphorylation of other ErbBs. With highest probability assigned to dimerization of ErB2 with ErbB3 this reaction uses all the ErbB2 and ErbB3 and ErbB1 and ErbB4 will dimerize with partners that can cross phosphorylate.

Probabilistic Model. A simple algorithm was used to convert rule priorities into probabilities. Namely, starting with the highest priority, for each group of priorities, we give half of the probability left to that group and divide that amount equally among rules in the group. Thus, for the priority assignment above, we get the following probability assignment

```
rl[1] -> 1: .5
rl[2] -> 2, rl[3] -> 2: .25/2 = .125
rl[4] -> 3, rl[5] -> 3: .125/2 = .0625
rl[6] -> 4, rl[7] -> 4, rl[8] -> 4: .125/3 = .0416
```

The problem now is, for a given starting state, to determine the probabilities for each of the different ErbBs in the final state. In particular we want the mean of random variables representing the number of different forms of ErbBs. That is we want $E(e) = \Sigma_p p \cdot P(|e| = p$ in a final state), where $|e|$ is the quantity of the named ErbB state.

To determine the distribution in a final state we formulated a series of formulas in Probabilistic Computational Tree Logic (PCTL) about the probability that the value of each random variable is in a given range and used the PRISM model checker [12, 15] to determine the corresponding probabilities. As this is a much more complex process than the discrete case, we used a scaled down version of the initial state for the discrete case study with no ErbB4 as a test case: `((10,E1B) (100,E2) (100,E3B))`. The table below compares the results using PRISM (row labeled `probl`) with those using the greedy discrete algorithm (row labeled `greedy`) to find the distribution in the final state.

	E1BD	E1BP	E2D	E2P	E3B	E3BP
probl	0.22	9.78	93.48	6.52	6.31	93.69
greedy	0	10	100	0	0	100

The results turn out to be quite close. We find that the probabilistic computation produces small non-zero values whenever the discrete computation produces zeros. This is consistent with the hypothesis that the probabilistic approach models the stochastic nature of the system. In the case that small non-zero values can safely be ignored, the discrete greedy algorithm is a better choice as it is substantially more efficient. However, if there is a chance that the small amounts could be amplified in a larger context (networks with positive feedbacks or non-symmetric structure) then a probabilistic model is more reliable.

In the following section, we explore more sophisticated choices in extending Pathway Logic to model stochastic behavior. We will analyze the resulting models using stochastic simulation techniques, which fall in between the two extremes of the simplistic “greedy method” and the exhaustive “probabilistic model-checking method” used in this section.

4 Beyond PL: Quantitative and Probabilistic Modeling

We extend the PL modeling formalism in two directions, which we eventually bring together: incorporating quantitative information and the notion of time; and incorporating stochastic information into the models. We note here that most stochastic modeling approaches integrate the notion of probability and time and exploit quantitative information to define and handle these concepts. For instance, the classic chemical master equation (CME) [8] describes the time evolution of a probability density function using partial differential equations. Other stochastic modeling formalisms, such as stochastic Petri nets [10], are also given semantics using the CME. As a result, simulation engines for stochastic models produce time series data of species concentrations.

The notion of time elapse and probabilistic transition are inherently coupled. However, as Gillespie points out in his seminal paper [8], while performing a stochastic simulation of CME, there is a certain decoupling between the choice of the next reaction to fire and the time that elapses before the effects of the reaction are observed. This is reflected in Gillespie’s Direct Method stochastic simulation algorithm (SSA), where the algorithm samples two distinct random variables for these two purposes.

In our extension of PL, we keep the two aspects separate and make the model modular with respect to the choices for these two parts. As a result, we get a natural and flexible modeling language that is more useful as a modeling and prototyping formalism. We also have the possibility of using different options for defining probabilistic rule firing as well as different choices for specifying the timing behavior.

The syntax of PL changes only very slightly when it is extended with probabilities. With each rule, we now associate a scalar value `ai`, called *weight* or *affinity*.

```
rl[r1]: ErbB1 ErbB1 => ErbB1:ErbB1  a1 = 1
rl[r2]: ErbB1 ErbB2 => ErbB1:ErbB2  a2 = 10
rl[r3]: ErbB2 ErbB2 => ErbB2:ErbB2  a3 = 10
```

There are different ways of interpreting these scalar affinities. In any state, the likelihood of a reaction will be *proportional* to the product of its affinity and the number of each reactant[1]: given a state $\mathbf{s}$ (represented as a function from species to its number) and a rule r with reactants $r_1, r_2, \ldots$, let $f_r(\mathbf{s})$ denote the product $a_r \Pi_i \mathbf{s}(r_i)$.

Let $\mathit{Enabled}(\mathbf{s})$ denote the set of all rules that are *enabled* in the state $\mathbf{s}$, i.e. all the reactions that can possibly occur and change the state of the system. The semantics of PL specifications extended with affinities can be given as a Markov chain. There are different choices for defining such a Markov chain, although in all cases reactions to fire are ultimately chosen by sampling from a uniform distribution. We describe three choices that we have explored here, *exactly one rule*, *at most one rule*, and *multiple rules*.

Exactly one rule at a time. One natural way is to assume that in any state, the events set in the probability space consists of all the enabled rules. If we assume that these rules are exclusive–that is no more than one reaction can happen at the same time–and independent, then the probability that a rule r fires in state $\mathbf{s}$ is exactly equal to

$$\text{Probability of firing } r \text{ in state } \mathbf{s} \doteq \frac{f_r(\mathbf{s})}{\Sigma_{r' \in \mathit{Enabled}(\mathbf{s})} f_{r'}(\mathbf{s})}.$$

Note that if exactly one rule is enabled in a given state, then that rule is fired with probability 1.

At most one rule at a time. If we assume the existence of a maximum constant M such that $\Sigma_{r \in \mathit{Enabled}(\mathbf{s})} f_r(\vec{s}) < M$ for all states $\vec{s}$, then the probability that a rule r fires in state $\mathbf{s}$ can be given to be $f_r(\mathbf{s})/M$. In contrast with the previous case, in this case we may have a nonzero probability of no rule firing.

Possible multiple rules. In this case, we allow the rules to fire simultaneously. This is done to account for possible co-occurrence of the rule firing events, as is the case in Probabilistic Boolean Networks [25]. In this case, the event space in state $\mathbf{s}$ consists of $2^{|\mathit{Enabled}(\mathbf{s})|}$ elements: each rule $r \in \mathit{Enabled}(\mathbf{s})$ may or may not fire. The probability of each event is the product of the probabilities that each reaction happens or does not happen in the next time step. Note here the higher computational burden of this method compared to the first two. The choice of a particular one may depend on the particular network under study, and on the available information (i.e., whether we are given general firing likelihoods, or affinities, or reaction rates).

We have specified above a *time-abstract* semantics for extended PL specifications. We can incorporate time in the semantics. To do this, we need to give a time-dependent interpretation to the affinities. Again, there are a few options here, and we describe two of them below: *exponential random* and *deterministic amortized* variables.

[1] For completeness, we mention here that the “product of reactants” can be replaced by the more accurate “number of different possible combinations between the reactants” as suggested in [9].

Exponential random variables. Inspired by [8, 9], while in state $\mathbf{s}$, we assume that the time that elapses before a reaction fires is given by an exponential random variable with decay constant $K \doteq \Sigma_{r \in Enabled(\mathbf{s})} f_r(\mathbf{s})$. In other words, the probability that t time units elapse while in state $\mathbf{s}$ is given by $p_1(t|\mathbf{s}) = e^{-Kt}$.

Deterministic amortized variables. We can consider a deterministic approach for computing the time elapse by assuming that the rate of change of a species' concentration is given by the difference between its propensity to be created (by rules that create that species) and its propensity to be destroyed (by rules that use up that species). Given a state $\mathbf{s}$, let p denote a species that is produced in the rules $Prod(p)$ and consumed in the rules $Cons(p)$, where $\{Prod(p), Cons(p)\} \in Enabled(\mathbf{s})$. Mathematically, we can say that

$$dp/dt \quad \doteq \quad \Sigma_{r \in Prod(p)} f_r(\mathbf{s}) - \Sigma_{r \in Cons(p)} f_r(\mathbf{s})$$

Now the time interval Δt between two adjacent states, $\mathbf{s}$ and $\mathbf{s}'$, *from the point of view of species* p can be computed by approximating the above expression via a first-order Taylor expansion, and solving for the time step Δt.

$$\Delta t \quad = \quad \frac{(\mathbf{s}'(p) - \mathbf{s}(p))}{\Sigma_{r \in Prod(p)} f_r(\mathbf{s}) - \Sigma_{r \in Cons(p)} f_r(\mathbf{s})}$$

We do not have a notion of global time in this case. Each species has its own clock.

As we mentioned earlier, the explicit decoupling of the state change aspect of a transition (or, equivalently, of the rule selection) from its timing aspects leads to greater flexibility in modeling and simulating PL models with affinities. It allows simulations to be first performed in a time abstract way and then, *if required*, to embed timing information in the simulation *a posteriori*, that is after the simulation has been performed.

Other approaches in the literature.

We review here three related approaches for stochastic modeling that have also partly inspired our choices above. The main difference between PL schemes extended with affinities and the following models is that the extended PL models and their simulation engine explicitly decouple time elapse and probabilistic state transition features and allows for more choices in each feature.

The Stochastic Simulation Algorithm. Gillespie's Stochastic Simulation Algorithm (SSA) [8] aims at simulating the evolution of a set of N chemical species interacting through M possible different reactions within a fixed volume V. Unlike the classical reaction-rate approach, which sets up a deterministic system of differential equations based on the "law of mass action", Gillespie's algorithm simulates the Chemical Master Equation (CME), which describes *both* the time and the probabilistic transition behavior of the system. Specifically, if $P(\mathbf{s}, t|\mathbf{s_0}, t_0)$ denotes the probability that the state is $\mathbf{s}$ at time t, given it was $\mathbf{s_0}$ at time t_0, the CME is given as:

$$\frac{\partial}{\partial t} P(\mathbf{s}, t|\mathbf{s_0}, t_0) = \\ \Sigma_r [f_r(\mathbf{s} - \nu_r) P(\mathbf{s} - \nu_r, t|\mathbf{s_0}, t_0) - f_r(\mathbf{s}) P(\mathbf{s}, t|\mathbf{s_0}, t_0)]$$

where ν_r is the vector of the change in the number of molecules of each species caused by a firing of reaction r.

Gillespie shows that his stochastic simulation algorithm (SSA) exactly simulates the above chemical master equation, [9]. At each step, the algorithm chooses two quantities:
(i) the time delay τ for the next reaction to occur;
(ii) the reaction r, among the enabled reactions, that will occur next.
The time step τ is a sample of the exponential random variable with decay constant $\Sigma_{r \in Enabled(\mathbf{s})} f_r(\mathbf{s})$. Note that this is the first option in the two choices for incorporating time in extended PL models described above. The reaction r to fire is chosen by sampling an integer random variable on $[1, M]$ with point probabilities $f_r(\mathbf{s}) / \Sigma_{r' \in Enabled(\mathbf{s})} f_{r'}(\mathbf{s})$. Note again that this is the first option, among the three options, for defining the next time abstract transition in extended PL models described above. As noted by Gillespie, this particular combination of choices for simulation (and the CME dynamics) is a consequence of the assumption that the propensity function, $f_r(\mathbf{s})$ for reaction r in state $\mathbf{s}$ is such that $f_r(\mathbf{s})dt$ defines the probability that the reaction r will fire once within the next infinitesimal time interval $[t, t + dt)$ given that the state is $\mathbf{s}$ at time t.

Stochastic Petri Networks. In Section 2, we mentioned that a PL model essentially encodes a Petri net. Stochastic Petri nets are an extension of Petri nets that incorporate random events. There are many variants of stochastic Petri nets in the literature. The standard stochastic Petri nets can also be given semantics using a chemical master equation, and hence the stochastic simulation algorithm can be used to perform simulations on such models [10]. The literature on stochastic Petri nets also considers other semantics for time elapse given by non-exponential random variables. Moreover, there are variants, called general stochastic Petri nets, that allow immediate and time delayed transitions.

Probabilistic Boolean Networks. Probabilistic Boolean Networks (PBN) have been introduced to model and simulate regulatory networks through a rule-based approach [25]. They are described by a set of boolean-valued nodes and functions (rules) that update these nodes. The rules to be fired are chosen probabilistically allowing for possible multiple rule firings. Unlike the Gillespie's SSA, they do not embed any notion of time, but just sequentially execute a particular set of rules.

5 Implementation and Simulations

Simple ErbB dimerization network (sErbB):

We consider the simple biological system containing three second-order forward reactions first described in Sec. 2 and enhanced with the introduction of affinities, as in Sec. 4. No reverse reaction is modeled in this simple network. Considering the corresponding reaction equations, the quantitative parameters are the concentrations of each reactant ($E1 \doteq \mathtt{ErbB1}$ and $E2 \doteq \mathtt{ErbB2}$) and dimerized products ($E1E1 \doteq \mathtt{ErbB1{:}ErbB1}$, $E1E2 \doteq \mathtt{ErbB1{:}ErbB2}$, $E2E2 \doteq \mathtt{ErbB2{:}ErbB2}$). Associated to each reaction are the three "association affinities" $\mathtt{ai}$.

If we conceive the aforementioned affinities as reaction rates, we can come up with a system of ordinary differential equations (ODEs) for solving the above network kinetics:

$$\frac{dE1E1}{dt} = \mathtt{a1} \cdot E1 \cdot E1; \quad \frac{dE1E2}{dt} = \mathtt{a2} \cdot E1 \cdot E2;$$
$$\frac{dE2E2}{dt} = \mathtt{a3} \cdot E2 \cdot E2; \quad \frac{dE1}{dt} = -2\mathtt{a1} \cdot E1E1 - \mathtt{a2} \cdot E1E2;$$
$$\frac{dE2}{dt} = -2\mathtt{a3} \cdot E2E2 - \mathtt{a2} \cdot E1E2.$$

Extended ErbB interaction network (eErbB):

A more complete ErbB interaction network was constructed based on an earlier work (see [13],[14]) and reported in Appendix 2. The *dissociations* (reactions $4, 5, 6$) and *internalizations* (which products are denoted by subscripts *in*) of the dimerized products are additional reactions in this network, compared to sErbB. Rate constants are named according to [13] and are assumed to be known.

Quantitative and Probabilistic Rule Selection.

We implemented a Probabilistic Rewriting Module (PRR) in Maude [4]. The following direct scheme describes the set of Maude procedures that is executed, where meta-level Maude programming is used.
1. A multiset of reactants and products with their associated quantities is defined as a "state".
2. A chemical reaction is represented by a "rule" where a multiset of reactants turn into a multiset of products. Applying a rule yields a decrease of the quantity of reactants and and increase of the quantity of products according to the reaction stoichiometry.
3. The firing likelihood of a rule is computed with a probability that depends on one of the approaches in Sec. 4.
4. At each step of the PRR simulation a rule is selected according to its probability; the selected rule is applied; the state is updated and specific new probabilities are computed. The rule rewriting and the state update are performed at the meta level.
5. Simulation terminates when there is no rule that can be selected according to the current state. For a network where no reverse reaction is considered (e.g., $sErbB$), PRR stops at the end of reactants usage. For network with feedback loops (e.g., $eErbB$) PRR stops when the steady state is reached, which can be inferred from the stabilization of the quantities for each of the species.
6. As a post processing step, time information is optionally embedded into the simulation data by using one of the approaches in Sec. 4. Further details can be found in Appendix 1.

5.1 Discussion of Results.

We first analyzed the simple network of growth factor receptor dimerization $sErbB$ to test the methodology and the performance of the PRR module in predicting the steady state of the biological pathway. We then enabled the time embedding feature (see Sec. 4) to add some dynamics to the simulations. Furthermore, considering the extended $eErbB$ interaction network, both sequential and time-dynamic simulations were performed. As we shall see, the results are in good agreement with experiments and previous computational studies.

5.1.1 Simulating sErbB dimerization by PRR module.

The deterministic simulation of the $sErbB$ network was obtained by solving the corresponding kinetic rate equations (see beginning of Sec. 5) in MATLAB and is reported in Fig. 2. The PRR procedure was executed via routines written in Maude. The initial concentrations of reactants (ErbB1, ErbB2) and products (the homodimers ErbB1:ErbB1, ErbB2:ErbB2 and the heterodimer ErbB1:ErbB2), as well as the affinities (a1, a2, a3) were varied systematically to examine the response of the network.

Fig. 1 shows PRR simulations for two of the probability models[2](exactly one rule at a time and possible multiple rules). The predictions from the two probability models are similar: the final state contains approximately 500 heterodimers ErbB1:ErbB2 (Fig. 1, left panels, cyan line)

[2] Concentration of reactants and products has a unit of number-per-volume and a volume 1 is assumed for all reactions studied in this work. The rate constants are the inverse of the concentration, times seconds, for the second-order dimerization reaction, and the inverse of seconds for the first order dissociation or internalization reactions.

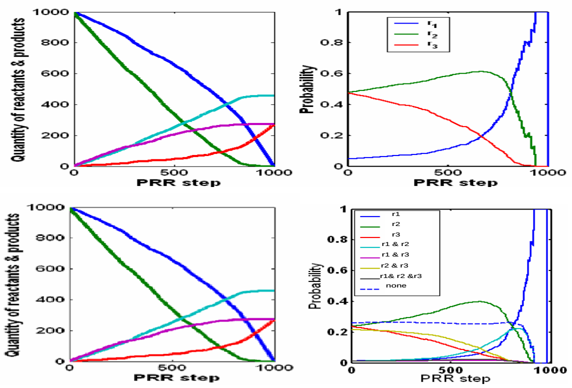


Figure 1. PRR simulation of $sErbB$ interaction network (ErbB1: blue, ErbB2: green, ErbB1:ErbB1: red, ErbB1:ErbB2: cyan, ErbB2:ErbB2: magenta) using the single rule (top) and multi-rule (bottom) probability models, given initial state ErbB1 = ErbB2 = 1000, ErbB1:ErbB1 = ErbB1:ErbB2 = ErbB2:ErbB2 = 0, and a1 = 1, a2 = a3 = 10. The change of quantities of the reactants and products (left plots) and the probabilities of the reactions are plotted along the simulation evolution (right plot).

and approximately 200 homo-dimers ErbB1:ErbB1 and ErbB2:ErbB2 (Fig. 1, left panels, red and magenta lines). The statistical significance of this similarity was confirmed from 500 independent runs of the simulations. The similar values for the outputs ErbB1:ErbB1 ≈ ErbB2:ErbB2 are due to the equality of the inputs ErbB1 = ErbB2 and the symmetric topology of the network. The probability values of the reactions in the network are plotted along the steps of PRR simulation for both probability models (Fig. 1, right plot, top for the single-rule model and bottom for multiple-rules model). The last rule selection model illustrates that the concurrency of a set of reactions, as a product of multiple probabilities, is insignificant in general compared to the individual reaction for this particular example, where symmetry and high quantities play an important role. Yet a non-negligible probability of concurrent reactions could occur if the concurrent reactions have considerably higher likelihood than the single reactions, as shown in Fig. 1.

Response of the network to changes in input variables.

The response of the $sErbB$ network to changes on the initial reactants concentration or the affinity of the reactions are explored in Figure 3, in Appendix 1. To summarize the outcomes, the observed non-sensitive response of the network to affinities under the “over expression” condition suggests a possibility to predict a system behavior with incomplete knowledge. In other words, accurate measurements of affinities are not strictly discriminating in this case. The less strict requirement of the input parameter would be extremely beneficial in analyzing biological data as experimental accuracy is very limited in in-vivo studies.

Incorporating time into the PRR module.

It is possible to enhance the outcomes of the PRR procedure by time-scaling the sequential data: as discussed in Sec. 4, this can be achieved either by post-simulation Taylor expansion (amortized approach), or by the application of the Gillespie algorithmic idea (exponential assumption). As shown in Fig. 2, plotting PRR prediction against the time by the amortized method (left) or by the exponential embedding (right) yields behaviors that agree with those from the ODE approach, as expected. As the network contains large numbers of reacting molecules (as expected for experimental conditions of the ErbBs dimerization), the time interval between two successive reaction events is small enough to validate the the first order Talyor expansion. Similarly, with these large numbers of reactants, the stochastic simulation (Gillespie algorithm) converges to the ODEs trajectories. Note that the apparent inconsistent termination in the time-resolved kinetics of reactants or products in the traces of ErbB2, ErbB1:ErbB2 and ErbB2:ErbB2 in Fig. 2(left) is explained by our a-posteriori method for embedding time, whereby local clocks for consumed reactants stop. As ErbB2 is consumed faster and is the first to be used up due to the higher dimerization rate, the products (ErbB1:ErbB2, ErbB2:ErbB2) run out accordingly from the reactions that require ErbB2 as reactant. The procedure thus treats this behavior as the termination of the kinetics for ErbB2, ErbB1:ErbB2 and ErbB2:ErbB2.
To conclude, a note on the computational burden. While the running time of the PRR with a single-rule approach is comparable to that of the rate-equations, the implementation of the multi-rule probability mode requires handling a wider range of possible outcomes and, hence, is affected by a longer running time.

5.1.2 Applying PRR to the more complex eErbB dimerization network.

We applied the stochastic simulations to the more complex $eErbB$ biological network which, based on the work in [13], additionally models dissociation of the dimers and internalization of monomers and dimers; both steady state and kinetic studies were performed and compared to the results obtained there; the study of the equilibrium (Appendix 2, Fig. 4, left) has showed, under the substrate saturating conditions, that the relative amount of hetero-(ErbB1:ErbB2) and homo-(ErbB1:ErbB1,

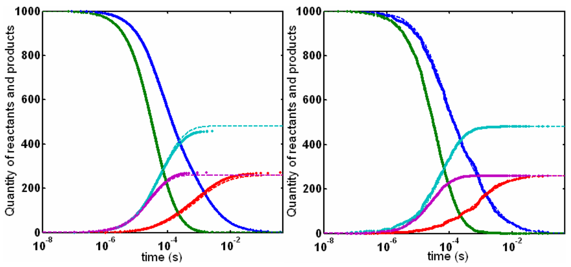


Figure 2. *sErbB* network kinetics (`ErbB1`: blue, `ErbB2`: green, `ErbB1:ErbB1`: red, `ErbB1:ErbB2`: cyan, `ErbB2:ErbB2`: magenta) predicted by time-resolved PRR (dots) via Taylor approximation (left) and exponential firing (right), in comparison to ODEs traces (dashed lines). Initial state are `ErbB1` $=$ `ErbB2` $= 1000$, `ErbB1:ErbB1` $=$ `ErbB1:ErbB2` $=$ `ErbB2:ErbB2` $= 0$, and `a1` $= 1$, `a2` $=$ `a3` $= 10$.

`ErbB2:ErbB2`) dimers varies according to the initial expression levels of `ErbB1` and `ErbB2` (named, respectively, `EGFR` and `HER2` in [13]).

Four extreme conditions were studied, corresponding to the four corners of the horizontal plane in Fig. 4 (left): 1. Both `ErbB1` and `ErbB2` are normally expressed, $3 \cdot 10^4$ per cell; 2. `ErbB1` is over-expressed, $6 \cdot 10^5$ per cell; 3. `ErbB2` is over-expressed, $6 \cdot 10^5$ per cell; 4. Both `ErbB1` and `ErbB2` are over expressed. We then performed the sequential PRR simulations with similar initializations: 1. `ErbB1` $=$ `ErbB2` $= 10^3$; 2. `ErbB1` $= 2 \cdot 10^4$, `ErbB2` $= 10^3$; 3. `ErbB1` $= 10^3$, `ErbB2` $= 2 \cdot 10^4$; 4. `ErbB1` $=$ `ErbB2` $= 2 \cdot 10^4$. Note that we scaled down the absolute quantities of `ErbBs` in this simulation for coherence with previous sections, but we kept the same ratio of over-expression and normal expression, i.e. 20-fold; in addition, we simplified the network by assuming an EGF saturation condition such that any monomer or dimer species without EGF bound is eliminated. According to these modifications, reproducing exactly the outcome in [13]) with a scaling factor of 20 is not to be precisely expected. Nevertheless, our results shown in the Table below are consistent with the outputs in Appendix 2, Fig. 4, left.

We also performed kinetic simulations with both `ErbBs` normally expressed (initial state 1.) and `ErbB2` over-expressed (initial condition 2.). Signaling homo-dimer (`ErbB1:ErbB1`) and hetero-dimer (`ErbB1:ErbB2`) were calculated by time-stretched PRR simulations (Appendix 2, Fig.5). The outcomes in Fig. 5 under these two circumstances are analogous to those in Fig. 4, right: when `ErbB1` and `ErbB2` are both $3 \cdot 10^4$ (normal expression), a low output of homo-dimer (`ErbB1:ErbB1`) and hetero-dimer (`ErbB1:ErbB2`) is observed; when `ErbB2` is 20-fold overexpressed over `ErbB1`, the hetero-dimers are greatly enhanced and dominate the signaling species, while the homo-dimers are suppressed down to ground level. The good agreement with [13](Appendix 2, Fig. 4, right) indicates the adequacy of the PRR module in handling these network kinetics.

`ErbB1`	`ErbB2`	`ErbB1:ErbB2`− `ErbB1:ErbB1`	`ErbB1:ErbB2`− `ErbB1:ErbB1` Fig.4, left; from[13]
10^3	10^3	6	$7.6 \cdot 10^3$
10^3	$2 \cdot 10^4$	260	$1.5 \cdot 10^4$
$2 \cdot 10^4$	10^3	-5019	$-2.3 \cdot 10^5$
$2 \cdot 10^4$	$2 \cdot 10^4$	2971	$1.6 \cdot 10^5$

6 Conclusion and Future Directions

We have discussed the underlying principles and several approaches for extending Pathway Logic with the ability to represent and reason about semi-quantitative and probabilistic aspects of biological processes. In summary our conclusions are as follows:

– There is a wide range of options to consider when analyzing a model containing some quantitative data. Depending on the number of reactants present in the environment, the accuracy of available quantitative information (the values for the reaction affinities, for instance, which are often only imprecisely known), the question of interest, and the abstraction level, one could use fast greedy approaches, or use more accurate stochastic simulation based approaches.

– The presented probability-based rule-selection strategies represent a reasonable approach to incorporate semi-quantitative information into Pathway Logic. A flexible approach to modeling temporal aspects was developed, allowing simulations to be either time sensitive, or to account for time a posteriori.

– Traditionally only the evolution of the quantities of reactants are observed when carrying out simulations. We observed that it is also interesting to observe the evolution of reactions probabilities.

There are a number of interesting questions left for future work. Experiments showed that simulations in which multiple reactions occur simultaneously yield results very close the simulations based on one reaction occurring at each step. This apparently is due to the product of probabilities of the single reactions being too small to have a substantial effect. Using a product to compute the probabilities of multi-reaction steps may not be appropriate as it corresponds to synchronous interaction, while in fact multiple independent reactions occur concurrently and asynchronously. A challenging problem is to develop a theory of truly concurrent probabilistic systems that accounts for asynchronicity.

A future step is to develop hybrid approaches capable of switching between different simulation methods according to suitable conditions. A possible technique would be to use generalized stochastic Petri nets as a formal representation.

References

[1] M. Calder, V. Vyshemirsky, D. Gilbert, and R. Orton. Analysis of signalling pathways using the PRISM model checker. In G. Plotkin, editor, *Proceedings of the Third International Conference on Computational Methods in System Biology (CMSB 2005)*, 2005.

[2] L. Calzone, N. Chabrier-Rivier, F. Fages, L. Gentils, and S. Soliman. Machine learning bio-molecular interactions from temporal logic properties. In G. Plotkin, editor, *Proceedings of the Third International Conference on Computational Methods in System Biology*, 2005.

[3] L. Cardelli and A. Phillips. A correct abstract machine for the stochastic pi-calculus. In *BioConcur'04*, 2004.

[4] M. Clavel, F. Durán, S. Eker, P. Lincoln, N. Martí-Oliet, J. Meseguer, and C. L. Talcott. The Maude 2.0 system. In R. Nieuwenhuis, editor, *Rewriting Techniques and Applications (RTA 2003)*, volume 2706 of *Lecture Notes in Computer Science*, pages 76–87. Springer-Verlag, 2003.

[5] S. Efroni, D. Harel, and I. Cohen. Towards rigorous comprehension of biological complexity: Modeling, execution and visualization of thymic t-cell maturation. *Genome Research*, 2003. Special issue on Systems Biology, in press.

[6] S. Eker, M. Knapp, K. Laderoute, P. Lincoln, J. Meseguer, and K. Sonmez. Pathway Logic: Symbolic analysis of biological signaling. pages 400–412, January 2002.

[7] F. Fages, S. Soliman, and N. Chabrier-Rivier. Modelling and querying interaction networks in the biochemical abstract machine BIOCHAM. *Journal of Biological Physics and Chemistry*, 4(2):64–73, 2004.

[8] D. Gillespie. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *Computational Physics*, 22:403–434, 1976.

[9] D. Gillespie. Exact stochastic simulation of coupled chemical reactions. *Physical Chemistry*, 81,25:2340–2361, 1977.

[10] P. J. Goss and J. Peccoud. Quantitative modeling of stochastic systems in molecular biology using stochastic Petri nets. *Proc. Natl. Acad. Sci. U. S. A.*, 95:6750–6755, 1998.

[11] J. Heath, M. Kwiatkowska, G. Norman, D. Parker, and O. Tymchyshyn. Probabilistic model checking of complex biological pathways. In *Computational Methods in Systems Biology (CMSB'06)*, 2006.

[12] J. Heath, M. Kwiatkowska, G. Norman, D. Parker, and O. Tymchyshyn. Probabilistic model checking of complex biological pathways. In *International Workshop on Computational Methods in Systems Biology*, volume 4210 of *Lecture Notes in Computer Science*, pages 32–47. Springer, 2006.

[13] B. Hendriks, L. Opresko, H. Wiley, and D. Lauffenburger. Quantitative analysis of her2-mediated effects on her2 and epidermal growth factor receptor endocytosis. *Biological Chemistry*, 278, 26:23343–23351, 2003.

[14] B. Hendriks, G. Orr, A. Wells, H. Wiley, and D. Lauffenburger. Parsing erk activation reveals quantitatively equivalent contributions from epidermal growth factor receptor and her2 in human mammary epithelial cells. *Biological Chemistry*, 280, 7:6167–6169, 2005.

[15] A. Hinton, M. Kwiatkowska, G. Norman, and D. Parker. PRISM: A tool for automatic verification of probabilistic systems. In H. Hermanns and J. Palsberg, editors, *12th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS06)*, volume 3920 of *Lecture Notes in Computer Science*, pages 441–444. Springer, 2006.

[16] N. Kam, D. Harel, H. Kugler, R. Marelly, A. Pnueli, J. Hubbard, and M. Stern. Formal modeling of C.elegans development: A scenario-based approach. In *First International Workshop on Computational Methods in Systems Biology*, volume 2602 of *Lecture Notes in Computer Science*, pages 4–20. Springer, 2003.

[17] M. Kwiatkowska, G. Norman, D. Parker, O. Tymchyshyn, J. Heath, and E. Gaffney. Simulation and verification for computational modelling of signalling pathways. In L. F. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol, and R. M. Fujimoto, editors, *2006 Winter Simulation Conference*, 2006.

[18] J. Meseguer. Conditional Rewriting Logic as a unified model of concurrency. *Theoretical Computer Science*, 96(1):73–155, 1992.

[19] F. Nielson, H. R. Nielson, C. Priami, and D. Rosa. Control flow analysis for bioambients. In *BioConcur*, 2003.

[20] J. S. Oliveira, C. G. Bailey, J. B. Jones-Oliveira, D. A. Dixon, D. W. Gull, and M. L. Chandler. A computational model for the identification of biochemical pathways in the Krebs cycle. *J. Computational Biology*, 10:57–82, 2003.

[21] M. Prez-Jimnez and F. Romero-Campero. Modelling EGFR signalling cascade using continuous membrane systems. In G. Plotkin, editor, *Proceedings of the Third International Conference on Computational Methods in System Biology (CMSB 2005)*, 2005.

[22] A. Regev, E. Panina, W. Silverman, L. Cardelli, and E. Shaprio. Bioambients: An abstraction for biological compartments, 2003. to appear TCS.

[23] A. Regev and E. Shapiro. The pi-calculus as an abstraction for biomolecular systems. *Modelling in Molecular Biology*, pages 219–266, 2004.

[24] A. Regev, W. Silverman, and E. Shapiro. Representation and simulation of biochemical processes using the pi-calculus process algebra. In R. B. Altman, A. K. Dunker, L. Hunter, and T. E. Klein, editors, *Pacific Symposium on Biocomputing*, volume 6, pages 459–470. World Scientific Press, 2001.

[25] I. Shmulevich, E. R. Dougherty, S. Kim, and W. Zhang. Probabilistic boolean networks: A rule-based uncertainty model for gene regulatory networks. *Bioinformatics*, 18:261–274, 2002.

[26] C. Talcott and D. L. Dill. The pathway logic assistant. In G. Plotkin, editor, *Third International Workshop on Computational Methods in Systems Biology*, pages 228–239, 2005.

[27] C. Talcott, S. Eker, M. Knapp, P. Lincoln, and K. Laderoute. Pathway logic modeling of protein functional domains in signal transduction. In *Proceedings of the Pacific Symposium on Biocomputing*, January 2004.

[28] Y. Yarden and M. X. Sliwkowski. Untangling the erbb signalling network. *Nat. Rev. Mol. Cel. Biol.*, 2:127–137, 2001.

Appendix 1: sErbB Network

Extracting Time Tags in the sErbB Model. As mentioned in Sec. 4, introducing quantitative information in the models allows further analysis of its dynamics and the possibility of calculating the actual time flow, as opposed to just keeping track of the sequential operations that are performed. In the case of the $sErbB$ model, the time interval between two adjacent states (the i^{th} and $(i+1)^{th}$) of a specific species in the network (say $E1$), that is $\Delta t_{E1_i} = t_{i+1} - t_i$, is the time interval when reactant $E1$ changes from state $E1_i$ to $E1_{i+1}$; it can be deterministically approximated by the "amortized" approach by discretizing the corresponding ODEs and solving for the time stamps:

$$\begin{aligned}
\Delta t_{E1_i} &= \frac{(E1_{i+1} - E1_i)}{(-2\mathtt{a1} \cdot E1_i \cdot E1_i - \mathtt{a2} \cdot E1_i \cdot E2_i)};\\
\Delta t_{E2_i} &= \frac{(E2_{i+1} - E2_i)}{(-2\mathtt{a3} \cdot E2_i \cdot E2_i - \mathtt{a2} \cdot E1_i \cdot E2_i)};\\
\Delta t_{E1E1_i} &= (E1E1_{i+1} - E1E1_i)/(\mathtt{a1} \cdot E1_i \cdot E1_i);\\
\Delta t_{E1E2_i} &= (E1E2_{i+1} - E1E2_i)/(\mathtt{a2} \cdot E1_i \cdot E2_i);\\
\Delta t_{E2E2_i} &= (E2E2_{i+1} - E2E2_i)/(\mathtt{a3} \cdot E2_i \cdot E2_i).
\end{aligned}$$

Responses of sErbB network to change of input variables studied by PRR. The outcomes of the simulations are reported in Figure 3. If the initial quantity of ErbB1 and ErbB2 are comparable (ErbB1 $\approx$ ErbB2), that is, there is no preferential cellular expression in either of the ErbBs, the final quantities of products largely depends on the value of the affinity (Fig. 3, top). However, when one of the ErbBs is over-expressed, e.g., ErbB2, the influence of the affinities is only moderate or weak. The ErbB2:ErbB2 homo-dimer greatly dominates the final state, regardless of the change in the affinities (Fig. 3, bottom). Furthermore, ErbB1:ErbB1 and ErbB1:ErbB2 have insignificant changes (for instance, the hetero-dimer ErbB1:ErbB2 varies only 2-fold in Fig. 3, cyan lines).

Appendix 2: eErbB Network

The Extended eErbB Network and its analysis via PRR. We report the reactions involved in this network. The units for kc are $[(\#/\text{cell min})^{-1}]$, while for the other rates are $[\text{min}^{-1}]$.

```
[r1]  : ErbB1 ErbB1 => ErbB1:ErbB1   kc = 0.001
[r2]  : ErbB1 ErbB2 => ErbB1:ErbB2   kc = 0.001
[r3]  : ErbB2 ErbB2 => ErbB2:ErbB2   kc = 0.001
[r4]  : ErbB1:ErbB1 => ErbB1 ErbB1   ku11 = 10
[r5]  : ErbB1:ErbB2 => ErbB1 ErbB2   ku12 = 10
[r6]  : ErbB2:ErbB2 => ErbB2 ErbB2   ku22 = 10
[r7]  : ErbB1 => ErbB1in   kin11 = 0.28
[r8]  : ErbB1:ErbB1 => ErbB1:ErbB1in   kin11 = 0.28
[r9]  : ErbB2 => ErbB2in   kin2- = 0.01
[r10] : ErbB2:ErbB2 => ErbB2:ErbB2in   kin2- = 0.01
[r11] : ErbB1:ErbB2 => ErbB1:ErbB2in   kin12l = 0.1
```

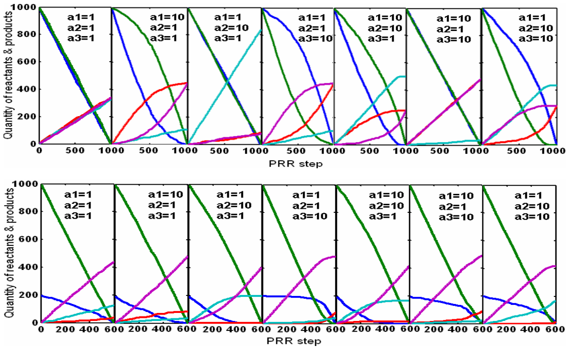
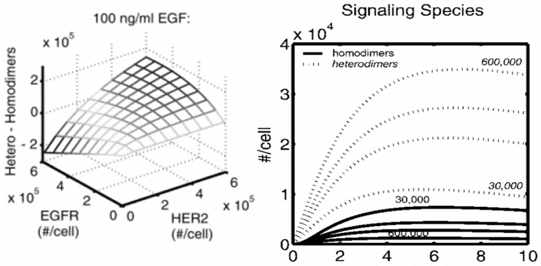


Figure 3. PRR simulation of sErbB network (ErbB1: blue, ErbB2: green, ErbB1:ErbB1: red, ErbB1:ErbB2: cyan, ErbB2:ErbB2: magenta) with altered reaction affinities. Top: the initial condition is ErbB1 = ErbB2 = 1000, ErbB1:ErbB1 = ErbB1:ErbB2 = ErbB2:ErbB2 = 0. Bottom: The initial condition is ErbB1 = 200, ErbB2 = 1000, ErbB1:ErbB1 = ErbB1:ErbB2 = ErbB2:ErbB2 = 0. The affinites are indicated within the single plots.



Hendrisk et al. 2003, JBC

Figure 4. A steady state simulation and a kinetic study of $eErbB$ presented in [13].

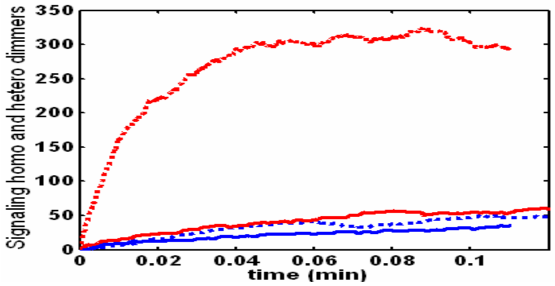


Figure 5. Quantities of signaling homodimer (ErbB1:ErbB1, blue) and heterodimer (ErbB1:ErbB2, red) under normal expression (initial values ErbB1 = ErbB2 = 1000, solid lines) or over-expression (initial values ErbB1 = 1000, ErbB2 = 20,000, dotted lines) condition, predicted by time-resolved PRR.