

# Algorithmic Systems Ecology: Experiments on Multiple Interaction Types and Patches

Ozan Kahramanoğulları<sup>1(✉)</sup>, James F. Lynch<sup>2</sup>, and Corrado Priami<sup>1,3</sup>

<sup>1</sup> The Microsoft Research – University of Trento Centre for Computational and Systems Biology, Trento, Italy

ozan@cosbi.eu

<sup>2</sup> Department of Computer Science, Clarkson University, Potsdam, NY, USA

<sup>3</sup> Department of Mathematics, University of Trento, Trento, Italy

**Abstract.** Dynamic behavior in ecosystems can emerge as a result of multiple interactions of different types as well as movements of the ecosystem species between different patches. The extinction behaviors in ecosystem models, which can result from the small species numbers, bring stochasticity to the foreground as they are often not observable in deterministic representations. To this end, we demonstrate an integrated approach to ecosystem modeling from an algorithmic systems biology point of view. We use a modeling interface, called LIME, which allows us to give biologically intuitive models of a plant-pollinator system's descriptions with varying interaction types and patches. Our models, written in a narrative style, are automatically translated into stochastic programming languages. The discrete stochastic nature of the models brings about the possibility to analyze the models with respect to their simulations as well as various graph representations. Our analysis provides an assessment of the functional dynamics of ecosystems with respect to the influence of various interaction patterns and patch links.

**Keywords:** Ecosystem modeling · Plant-pollinator networks · Multiple interactions · Stochasticity

## 1 Introduction

Systems biology is established as a methodology that assists investigations for obtaining knowledge on dynamics and functioning of biological systems at various levels. Recent developments in computational methods now offer new avenues of investigation that provide contributions to long standing discussions also in ecology. Individual based models (IBMs) [3, 9, 10] have been instrumental in addressing different issues on ecosystems such as the relationship between patterns and processes [23]. Similar considerations that are centered around complex adaptive systems view of ecosystems have also been proven useful in addressing dynamical properties of ecosystems [19].

Until relatively recently, ecosystem modeling has been dominated by continuous deterministic representations that provide estimations of populations'

dynamics with respect to the species interactions. However, the greater availability of powerful computers made a departure from more traditional considerations feasible with the availability of new capabilities.

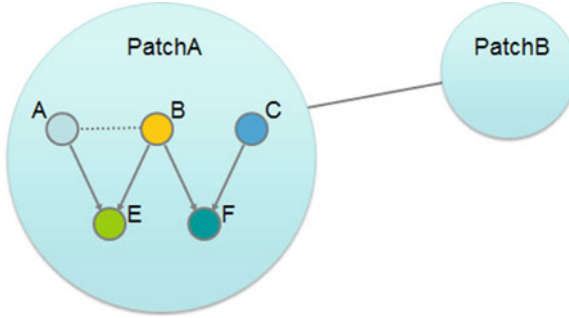
A point of interest in IBMs is on capturing certain sources of noise, which can be important in small populations [24]. This is because the same amount of living biomass can have a noisy behavior if it corresponds to an individual of an ecosystem rather than millions of individuals of the same system, for example, as in the relationship between whales and planktons. Stochasticity, which is exploited in some IBMs, is an instrumental feature for studying the inherent noise in ecosystems, both at individual level and at the population level. In contrast to deterministic population-level models, stochastic models make it possible to simulate actual extinction events. In this respect, generalistic discussions of ecosystem models are limited in both handling the noisy behavior of small populations and modeling extinction. However, the extinction of rare species is at the frontier of applied ecological research and conservation biology.

Another challenge that is confronting ecosystem models is the representation of different types of concurrent interspecific interactions [21]. In ecology literature the focus is on single interaction types, with an emphasis on food webs, and an increasing focus on plant-pollinator networks in isolation. However, in ecosystems, different kinds of interactions always happen in parallel. The capability to model them simultaneously is important for a better understanding of ecosystems.

In this paper, we present an algorithmic systems biology approach [26] for compositional stochastic modeling of ecosystems with different kinds of interactions between species and movement of species between patches, that is, spatial regions [14]. For our ecosystem models, we use the CoSBI Lab LIME (Language Interface for Modeling Ecology) [14], which is designed for modeling ecosystems with multiple interaction types and patches. The LIME language allows the user to give a biologically intuitive model description in a narrative style. This makes it possible to specify in the same model multiple parallel ecological interactions.

The LIME language is designed to perform static analysis on the model structures prior to the translation of the model description into a program suitable for stochastic dynamical simulation. These features make it possible to design, simulate and analyze ecosystem models with varying patterns of interactions and structures without dealing with the technicalities of specialized programming languages. In LIME, composability of the used language constructs play a key role in expressing the different kinds of interactions in a unified manner in a single model [15].

Besides mathematical approaches based on continuous differential equations, and experimental approaches of biology, which are however difficult in ecology, our algorithmic systems biology approach provides a discretized point of view of the models. This in return makes it easier to adapt computational analysis techniques for querying to the models and simulations with them. As an illustrative example for this, we discuss how an approach for discrete stochastic flux analysis can be used to obtain various graph representations of the models that display the resource flows in the modeled systems, and the causal reading of these flows.



**Fig. 1.** The graphical representation of a pollinator network on two patches, given with the LIME model in Fig. 2. The species A and B are pollinators for the plant E, whereas B and C are pollinators for the plant F. There is direct competition between the species A and B. The pollinator species can move between the two patches A and B.

As an example case study of an ecosystem with multiplicity of ecological interactions acting in parallel, we present models of a plant-pollinator system with varying interaction patterns and patch structures. Because ecological systems and processes are inherently variable, composability of the LIME language and the constructs of the underlying framework help to manage the emerging complexity. This way, we discuss and illustrate how these features can be used for the construction of complicated ecosystem models together with their simulation and analysis while keeping track of stochasticity and quantifying its consequences.

## 2 Algorithmic Modeling with LIME

We use a programming language that is based on process algebras. Process algebras are formal languages, which were originally introduced as a means to study the properties of complex reactive systems by providing rigorous discrete syntactic representations of these systems. In process algebra, concurrency, that is, the view of systems in which interacting computational processes are executing in parallel, is a central aspect. Stochastic extensions of these languages are now broadly used tools for describing the dynamics of biological systems [27], as they faithfully capture mass action kinetics with a continuous time Markov chain semantics. While continuous and discrete representations approximate each other, discrete representations are more realistic for the case of biological systems, because populations are described by discrete variables and both continuous and discrete representations are continuous in time. Moreover, discrete representations provide the expressive power to model otherwise challenging structures [2, 17].

The algebraic operators and the language constructs of the specialized stochastic process algebra languages allow them to capture the structural and functional aspects of the systems that they model. The biological system models written in these languages vary within a spectrum that spans from molecular

```

duration time 1.0
interactions
beeA pollinates flowerE
beeB pollinates flowerE
beeB pollinates flowerF
beeC pollinates flowerF
beeA and beeB compete

birth and death dynamics
beeA dies
beeB dies
beeC dies
flowerE dies
flowerF dies

patch dynamics
beeA moves from patchA to patchB
beeA moves from patchB to patchA
beeB moves from patchA to patchB
beeB moves from patchB to patchA
beeC moves from patchA to patchB
beeC moves from patchB to patchA

initial population
32 beeA in patchA
32 beeB in patchA
32 beeC in patchA
32 beeA in patchB
32 beeB in patchB
32 beeC in patchB

1000 flowerE in patchA
1000 flowerF in patchA
1000 flowerE in patchB
1000 flowerF in patchB

```

**Fig. 2.** LIME description of the model depicted in Fig. 1. The rates values are not specified, they are thus set to the default value 1.0 for this case.

biology to ecology with respect to their levels of abstraction. However, the mathematical syntax of these languages makes them difficult to make modifications on the models. When the modifications on the models involve functional phenomena such as patterns of interactions, for example, as in ecosystem models, the syntax of these languages becomes even more challenging. This creates obstacles for these languages to be used effectively in experiments that involve variations of the same model. As a result of this, specialized high level languages that provide user friendly interfaces to these languages are developed and used for modeling biological systems at these different levels of biological systems for expressing various phenomena [11, 13, 14].

We describe our model in the language of LIME (Language Interface for Modeling Ecology) [14, 15]. After performing static analysis on the model structure, the LIME translation software tool translates the model description into the stochastic process algebra languages *BlenX* [6] and *SPiM* for simulation. In Appendix A, we give a short introduction to *BlenX*. For an indepth exposure to *BlenX* we refer to [5, 6], and for *SPiM* to [1, 13, 17, 22]. A graphical representation of a model is depicted in Fig. 1, and its complete LIME description is given in Fig. 2. There, the rates are not specified, they are thus set to the default value of 1.0 for this case.

A LIME input file can consist of five parts that describe different aspects of the model. Some of these parts are optional, therefore they can be excluded in models. However, the model in Fig. 2 includes all these parts. The first part,

```

duration time 1
interactions
  beeA pollinates flowerE with rates
    0.00294651236 and 0.00240000000
  beeB pollinates flowerE with rates
    0.0146105379 and 0.00191794978
  beeB pollinates flowerF with rates
    0.0163731225 and 0.0000820502154
  beeC pollinates flowerF with rates
    0.00301982726 and 0.00240000000
birth and death dynamics
  beeA dies with rate 2.4
  beeB dies with rate 2.0
  beeC dies with rate 2.4
  flowerE dies with rate 0.7
  flowerF dies with rate 0.7
initial population
  32 beeA 32 beeB 32 beeC
  1000 flowerE 1000 flowerF

```

**Fig. 3.** A model obtained from the one depicted in Fig. 1, however on a single patch, without competition interactions, and with explicit rate values.

initiated with the keyword `duration`, is a single statement on the simulation duration. In this model, the simulations are specified to run until time point 1.0.

The second part, initiated with the keyword `interactions`, consists of the sentences that describe the interactions of the individuals of the modeled ecosystem. Each sentence describes an interaction in the ecosystem together with the ecological patch where it happens and its rate. The interactions can be of four different kinds: *predator-prey*, *plant-pollinator*, *direct competition*, and *facilitation*. The model in Fig. 2 contains only plant-pollinator and direct competition interactions. There are three pollinator species, that is, `beeA`, `beeB`, `beeC`, and two flower species `flowerE` and `flowerF`. The species `beeA` is a pollinator for the `flowerE`, and `beeC` is a pollinator for the `flowerE`, whereas `beeB` is a pollinator for both `flowerE` and `flowerF`. `beeA` and `beeB` have a direct competition interaction between each other.

The optional third part of the input file, initiated with the keywords `birth and death dynamics`, collects the information on the birth and death rates of the species. Each sentence in this part describes the birth rate or the death rate of a species in each habitat patch. Without habitat patch specification, a rate is distributed and applies to all patches: this way, general rates can be defined. In the model in Fig. 2, death rates for all the species `beeA`, `beeB`, `beeC`, `flowerE` and `flowerF` are defined as the default rate value 1.0.

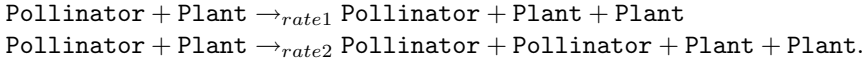
The optional fourth part of the input file, initiated with the keywords `patch dynamics`, contains the information on the rates of the movements between patches of the ecosystem: each sentence here describes the migration rate between two particular patches of a given species. In the model in Fig. 2, the bee species can move between connected patches as depicted in Fig. 1. `patchA` and `patchB` are connected such that `beeA`, `beeB` and `beeC` can move between them.

The fifth part, initiated with the keywords `initial population`, provides the information on the initial population sizes at the beginning of a simulation. In the model in Fig. 2, there are 1000 flowers and 32 bees of each species in each patch at the initial state.

LIME translation algorithm maps the sentences of the model into program code that implements the corresponding reactions. For example, each sentence of the form

Pollinator pollinates Plant with rates *rate1* and *rate2*

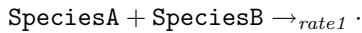
is mapped to two reactions



Each sentence of the form

SpeciesA and SpeciesB compete with rate *rate1*

is mapped to a reaction



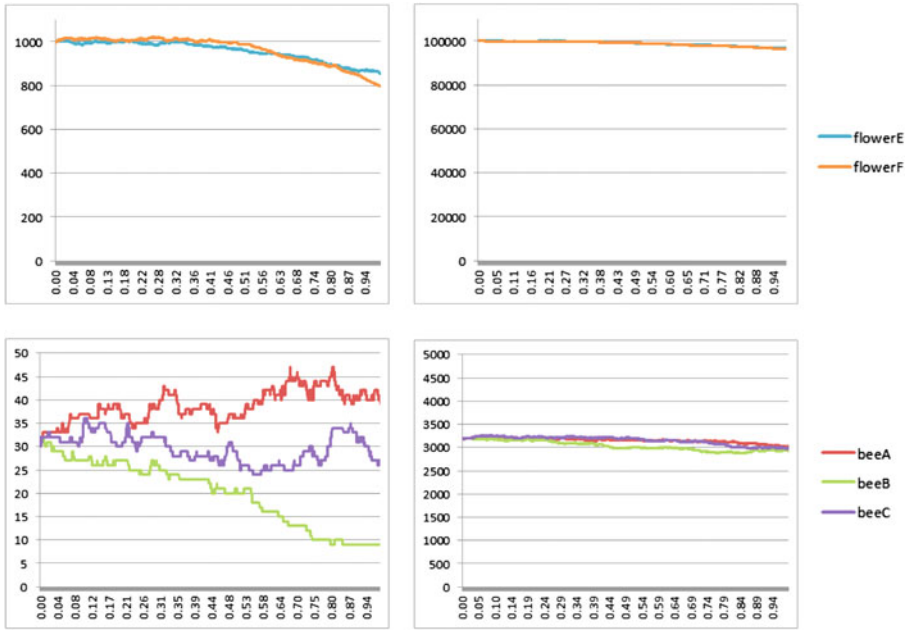
In the model, when movement of species between spatial regions, given with patches, are defined or interactions are defined specific to certain patches, the LIME translation algorithm distributes these patches to the reactions with respect to the model.

Due to these features, LIME models can be written, extended and modified with a great ease with almost no prior knowledge of this language. As we illustrate below, this capability makes it very easy to experiment with variations of models.

### 3 In Silico Experiments

As a first step for an analysis of the model, we work with a single-patch model, given in Fig. 3, as in `PatchA` of the model depicted in Fig. 1, however we exclude the direct competition between `beeA` and `beeB`. That is, there are only plant-pollinator interactions, where `beeA` is a pollinator for `flowerE`, `beeC` is a pollinator for `flowerF`, and `beeB` is a pollinator for both `flowerE` and `flowerF`.

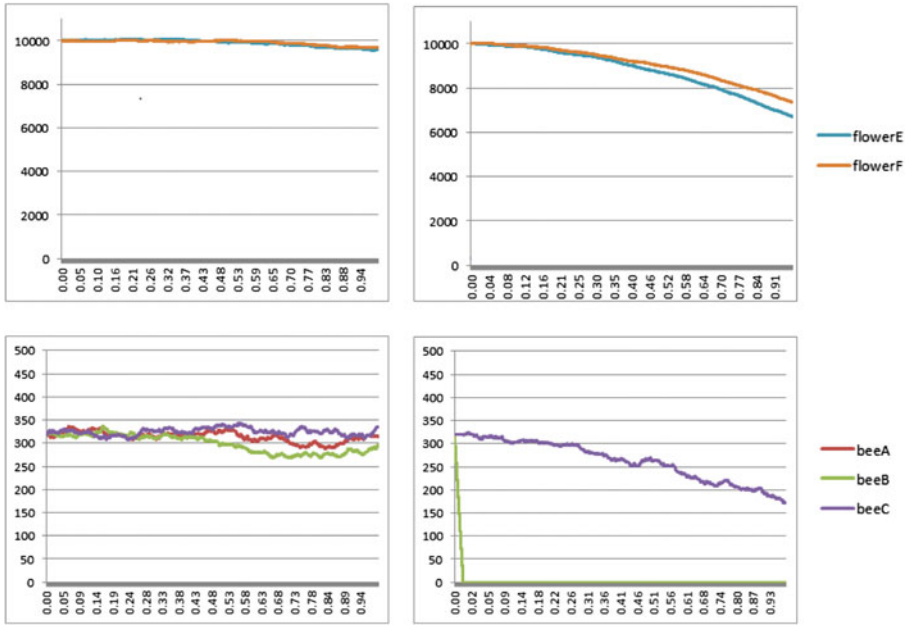
We instantiate the rates of the model such that the simulations display a close to stable behavior. For this, we obtain the rates by fitting the ODE representation of the model to steady state, where the death rates and initial quantities are fixed as in Fig. 3. For the fitting, we used the least-squares method provided by the PET tool [28]. We assume that the time unit of the simulations is one year. This provides realistic death rates for the bee populations in accordance with their average lifespans within a season, and allows us to observe the population behavior in response to interactions within that season. With stochastic simulations, we obtain a close to stable level for the time interval from 0 to 1, however as the simulation continues, the levels of the species depart from the steady state values due to the fluctuations in the system. These simulations result either in an exponential growth of the numbers of some or all of the species or in extinctions, depending on the initial bias given by the fluctuations.



**Fig. 4.** The plots on the left are from a simulation with the model in Fig. 3. The plots on the right are from a simulation with a version of the model, which is scaled by 100. The upper plots display the behavior of the flower species, whereas the lower plots display the behavior of the bee species.

As we scale the model by multiplying the initial numbers by a scaling factor, we observe extended stability with respect to simulation time as depicted in Fig. 4, however with an increased tendency towards extinctions as the simulation continues. This contrasts with the exponential increases often observed with lower species numbers in the non-scaled model. Because our focus of interest is the behavior within a season, the model does not factor for the seasonal changes between two years. The model thus gives an estimate of the tendencies within one time unit after the start of the simulation that are delivered by the interaction patterns and also patch structures as we analyze below.

In order to analyze the effect of direct competition to the system behavior, we introduce a direct competition interaction between the species `beeA` and `beeB` by adding the following sentence to the interactions of the model: “`beeA` and `beeB` compete with rate 0.1”. The direct competition interactions can be seen as a mechanism that stresses `beeA` and `beeB` concomitantly when they are present at the same patch. With the introduction of this interaction, as depicted in Fig. 5, we observe that in the simulations the numbers of the species `beeA` and `beeB` reduce until at least one of them goes extinct. As a result of this the numbers of the species `flowerE` and `flowerF` reduce as their pollination by `beeA` and `beeB` becomes hampered. When we increase the rate of the direct competition inter-

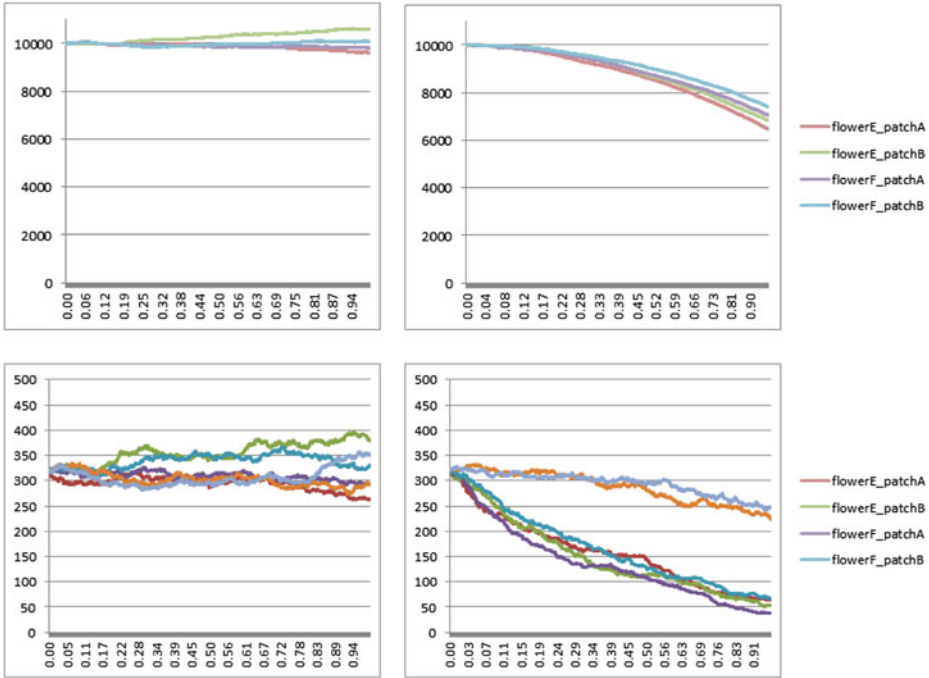


**Fig. 5.** The plots on the left are from a simulation with the model in Fig. 3. The plots on the right are from a simulation with a model that extends the model in Fig. 3 with competition between **beeA** and **beeB**. The upper plots display the behavior of the flower species, whereas the lower plots display the behavior of the bee species. The simulations are with the versions of the models, which are scaled by 10.

action from 0.1 up to ten orders of magnitude, we observe that this increases the extinction of the competing species within the considered time interval, however it does not accelerate the extinction drastically in comparison to original rate. Decreasing the competition rate more than two orders of magnitude brings the behavior of the system close to the levels that are observed without competition.

In order to observe the effect of the availability of patches, between which the bee species can move, we extend the model to two patches. The simulations with the different versions of this model are depicted in Fig. 6. In the first version of this extension, we consider a model without competition. This model is obtained by extending the model in Fig. 3 with the sentences in Fig. 7. The simulation plot, depicted on the left-hand-side of Fig. 6, is with this model. As a second step for a consideration of patch dynamics, we extend this model with direct competition. The plot on the right-hand-side of Fig. 6 is with the model, which extends patch dynamics with direct competition between **beeA** and **beeB** as above. Both models behave similar to their single patch versions as can be seen by comparing these plots with those in Fig. 5. However, for the case of the model without competition, the movement between patches provides a minor stabilizing effect to the pollination in comparison to the single patch version.





**Fig. 6.** Simulation plots with the model that extends the model in Fig. 3 to two patches, named A and B. The plots on the left are without competition, whereas the ones on the right include the competition between *beeA* and *beeB*. The upper plots display the behavior of the flower species and the lower plots display the bee species. The simulations are with the versions of the models, which are scaled by 10.

In order to model the different conditions on different patches, we restrict the competition between *beeA* and *beeB* to patch A. We then performed experiments, where we restricted the movement of species. These variations of the model are as follows: (1) only *beeA* and *beeB* can move between patch A and patch B, and *beeC* is stationary on both patches; (2) only *beeA* moves between patch A and patch B; (3) only *beeB* moves between patch A and patch B. Example simulation plots with these variations of the models are depicted in Fig. 8, where the plot on the left is for the case where all the bee species can move between two patches, whereas the plot on the right is for the case where only *beeA* and *beeB* can move between patches.

In all these variations of the model, the restriction of competition to a single patch provides a safe environment for the pollination in the patch without competition, which is patch B. Apart from this, the two models with the simulations in Fig. 8, have similar behaviors since the movement of *beeC* between two patches does not strongly hamper the competition between *beeA* and *beeB*,

however their movement has a minor influence that reduces the effect of the competition. The cases where either only `beeA` or only `beeB` can move between two patches display similar behaviors to the simulation depicted on the left-hand-side of Fig. 8.

```

patch dynamics
beeA moves from patchA to patchB with rate 0.1
beeA moves from patchB to patchA with rate 0.1
beeB moves from patchA to patchB with rate 0.1
beeB moves from patchB to patchA with rate 0.1
beeC moves from patchA to patchB with rate 0.1
beeC moves from patchB to patchA with rate 0.1
    
```

Fig. 7. The sentences that extend the model in Fig. 3 with patch dynamics

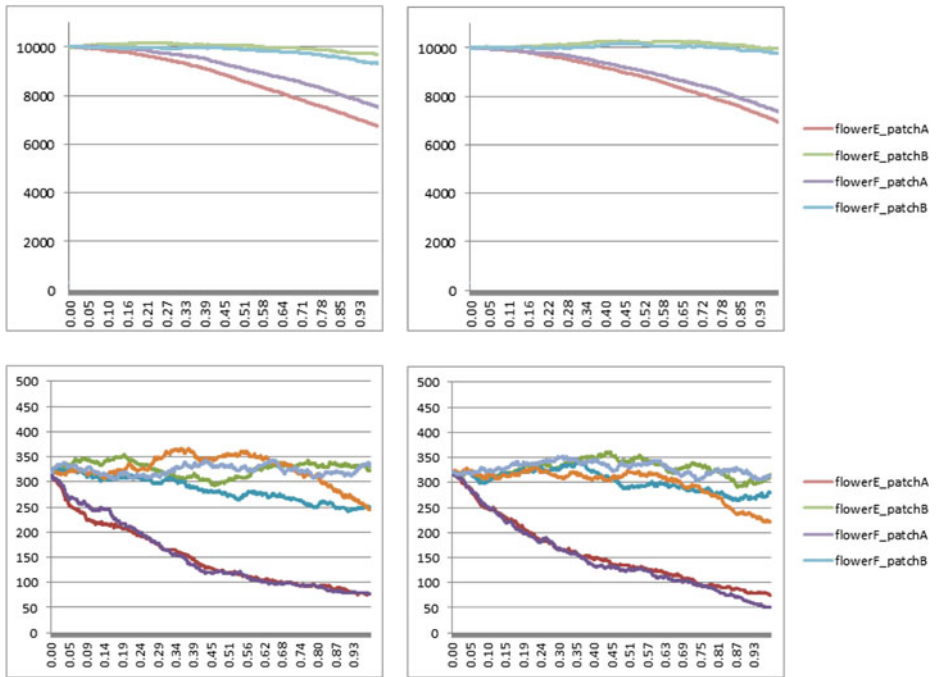


Fig. 8. Simulation plots with the model that extends the model in Fig. 3 to two patches, named A and B. In these models, there is competition between `beeA` and `beeB`, however only in patch A. The simulation plots on the left are with a model where all the bee species can move between both patches, whereas the plots on the right are with a model where the movement between patches is restricted to only `beeA` and `beeB`, and `beeC` are stationary on both patches. The upper plots display the behavior of the flower species, whereas the lower plots display the behavior of the bee species. The simulations are with the versions of the models which are scaled by 10.

1 : beeA+ →	x 70
2 : beeA + flowerE → beeA + beeA + flowerE + flowerE	x 71
3 : beeA + flowerE → beeA + flowerE + flowerE	x 72
4 : beeB →	x 58
5 : beeB + flowerE → beeB + beeB + flowerE + flowerE	x 42
6 : beeB + flowerE → beeB + flowerE + flowerE	x 333
7 : beeB + flowerF → beeB + flowerF + flowerF	x 389
8 : beeC →	x 67
9 : beeC + flowerF → beeC + beeC + flowerF + flowerF	x 49
10 : beeC + flowerF → beeC + flowerF + flowerF	x 57
11 : flowerE →	x 669
12 : flowerF →	x 642

**Fig. 9.** The reactions that occur during a simulation and their numbers of occurrences.

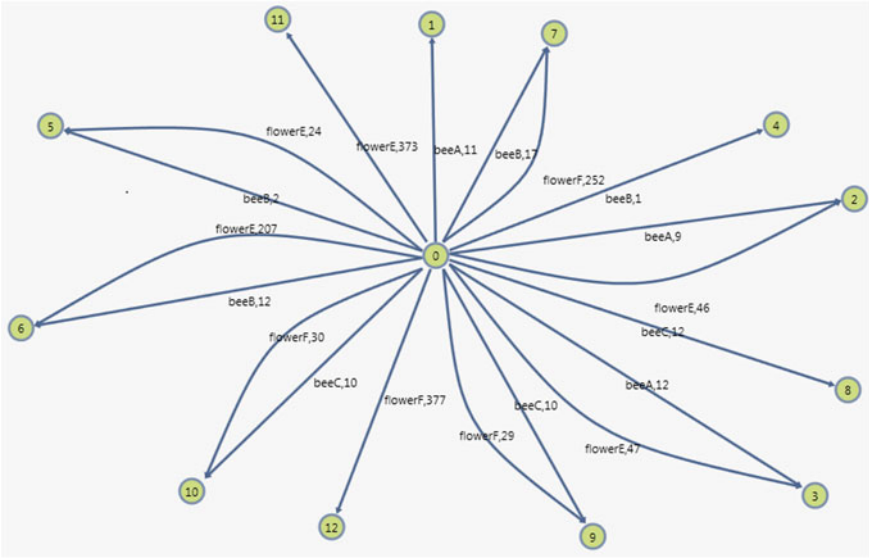
## 4 Flux Analysis

The reactions that are given by a LIME model are interpreted by the stochastic simulation algorithm as a continuous time Markov chain (CTMC). A simulation with a model can thus be seen as reduction of this complex structure, that is, the model, into a simpler structure, that is, the simulation trajectory, which is a path in the CTMC. However, during this process, some of the information on the simulation can be lost. In particular, the causality between the reactions with respect to the flow of resources during simulation remains hidden in the total order of the reaction instances of the simulation trajectory, which is emphasized by the unique time stamps of these reaction instances.

In [16], Kahramanoğulları and Lynch have introduced a method for flux analysis on stochastic simulations with reaction networks, where flux is the flow of resources between reactions of the network. The approach of [16] is based on inspecting the reaction instances from the point of view of their dependencies on one another during simulation, and relaxing their total order into a partial order structure. The method then uses this partial order as a representation of causal dependencies in the simulation, and processes it further to observe the flux in the network with respect to the flow of the resources from a reaction to another in arbitrary time intervals. The notion of flux obtained this way does not only approximate the notion of flux given by continuous deterministic representations, but also reveals other information that are not given by differential equation analysis.

We perform flux analysis for the simulation time interval from 0 to 1. Figure 9 gives a list of the reactions together with their ids and their number of occurrences during a simulation. The flux graphs of this simulation within the considered time interval are depicted in Figs. 10, 11, 12.

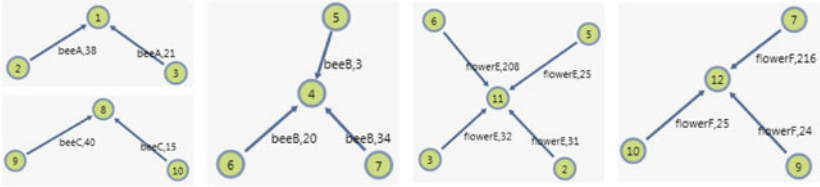
Figure 10 shows how resources present at the beginning of the simulation are distributed to the reactions of the network. Here, 0 denotes the initial state, and the directed edges of the graph display how resources at the initial state are



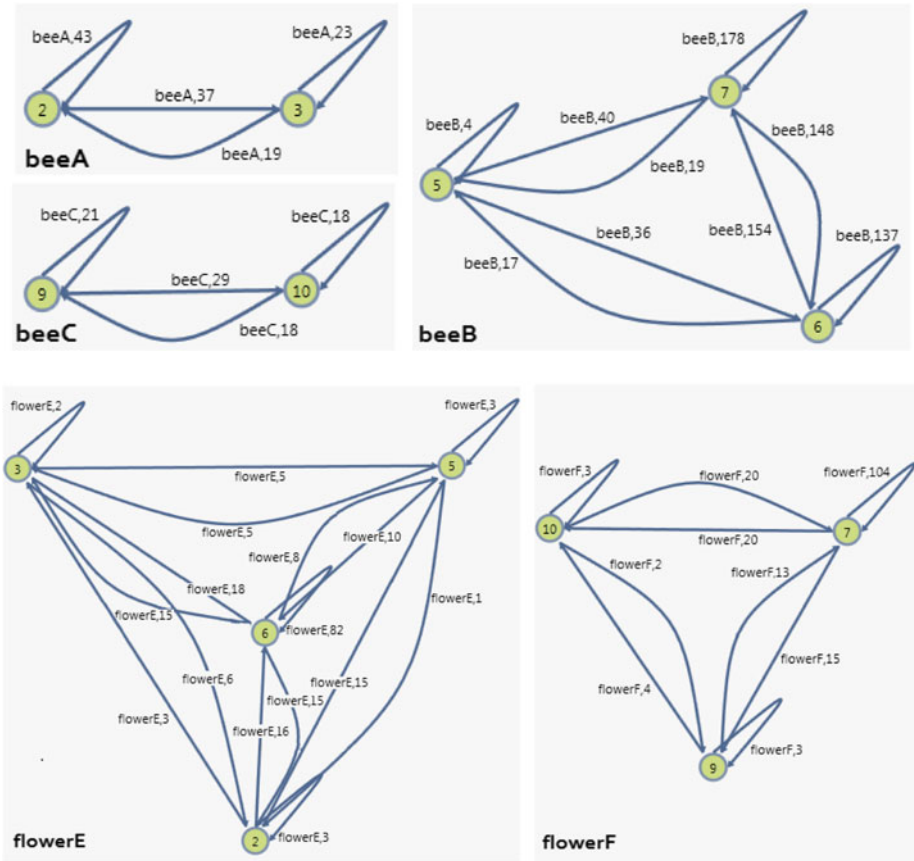
**Fig. 10.** Flux analysis graph for the simulation time interval from 0 to 1, indicating the quantity of the resources distributed from the initial state to the reactions of the network. Here, 0 denotes the initial state, and Fig. 9 gives a list of the reactions together with their ids.

consumed by the reactions listed in Fig. 9. As shown by this graph, 11 of **beeA**, 1 of **beeB**, 12 of **beeC**, 373 of **flowerE** and 377 of **flowerF** die without participating in any pollination interaction. An implication of this observation is that substantial fractions of the **beeA** and **beeC** populations die without participating in any pollination interaction, which removes one third of these species. However, almost all of the **beeB** participate in the reactions. A possible interpretation of these facts is that **beeB** is the most important bee species for maintaining the system. In order to check this hypothesis, we modified the pollination rates of **beeA**, **beeB**, and **beeC** in three different versions of the model. The simulations showed that lowering the rates of **beeA** and **beeC** by a third of the original rates does not affect the stability as much as doing the same to **beeB** as this causes a rapid decline in the levels of all the species. This observation agrees with the evidence provided by the flux analysis indicating that **beeB** plays a more important role in maintaining the system.

Figure 11 shows for different species the resources flowing from other reactions to death reactions. We observe that 59 of **beeA** participate in a pollination before they die, however 38 of these pollination interactions result in a reproduction of **beeA**. Similarly, 55 of **beeC** participate in a pollination before they die, and 40 of these pollination interactions result in a reproduction of **beeC**. On the other hand, while 57 of **beeB** participate in a pollination before they die, only 40 of these pollination interactions result in a reproduction of **beeC**. From these observations, it follows that although comparable numbers of **beeA**, **beeB** and



**Fig. 11.** Flux analysis graphs for the simulation time interval from 0 to 1, indicating the quantity of species flowing from other reactions of the network, excluding the initial state, to those for the death of the species. These are, respectively, for **beeA**, **beeC**, **beeB**, **flowerE** and **flowerF**. Figure 9 gives a list of the reactions and their ids.



**Fig. 12.** Flux analysis graphs of different species for the simulation time interval from 0 to 1. The graphs indicate the quantity of species flowing between the pollination reactions of the network. These exclude the initial state and the reactions for the death of the species. Figure 9 gives a list of the reactions and their ids.

**beeC** die after participating in a pollination interaction, **beeB** reproduces less frequently as a result of these pollinations in comparison to **beeA** and **beeC**, which can explain the critical role played by **beeB** in maintaining the system.

Other observations in Fig. 11 that are made on **flowerE** due to the reaction for its death indicate that most of the pollinations of **flowerE** takes place as result of pollinations by **beeB**, that is, 208 pollinations. This contrasts with the 25 pollinations of **flowerE** by **beeB** that result in the reproduction of **beeB**, and the other 63 pollinations by **beeA**. Similarly, for the case of **flowerF**, 216 of the death reactions occur after having participated in a pollination by **beeB**, which does not result in a pollination. This contrasts with the total of 49 reactions that occur after a pollination by **beeC**.

Figure 12 shows for different species the resources flowing between the reactions of the network that model the pollination of the flower species. An observation on these graphs indicates that **beeB** flows between various pollination reactions an order of magnitude more than both **beeA** and **beeC**. In particular, the cycling of **beeB** between reactions 6 and 7 of Fig. 9 contrasts with those for reactions 2 and 3 for **beeA** and for reactions 9 and 10 for **beeC**. This observation carries over to the behaviors of **flowerE** and **flowerF** with respect to their incoming resources at reactions 6 and 7 in contrast to other reactions. This points out that stability of the system is mainly due to pollination of **flowerE** and **flowerF** by **beeB**, however only a fraction of these pollinations reflect to the reproduction of **beeB**.

## 5 Discussion

The development of methods that can contribute to an understanding of the relationships between individual level interactions and actions of ecosystems' species and resulting behaviors at the ecosystem level is an area of research with implications on conservation ecology. In particular, when agricultural ecosystem domains are considered, methods that can help to predict the impacts of human interference can help to assist policy making where conservation ecology has also economic consequences. In this paper, with the aim of contributing to this discussion, we outline an integrated approach from a computational systems biology point of view.

We propose an integrated approach as this allows us to bridge methods from various fields and use them under a unifying setting to address different aspects of the models where they have specialized strengths. In this respect, the LIME language, that stems from a language-design perspective, makes it possible to easily construct the models and manage them [14]. On the other hand, the techniques that we borrow from continuous deterministic ODE analysis for parameter estimation provide the means to calibrate our model in order to factor for the context, that is not included in the model structure, of the modeled ecosystem [28]. Our flux analysis brings together ideas from concurrency theory that allows us to exploit the discrete structures of the stochastic simulations to analyze them as graphs that explain the flow of species. This is done by treating the species as resources that flow between ecosystem events [12, 16].

Besides the other methods, which are standard in systems biology, our stochastic flux analysis is prone to potential uses in ecology. With respect to the case analysis of the present paper, an example is the identification of the species that are critical to the robustness of the system. As our flux analysis showed, **beeB**, which pollinated both flower species, is more important in this regard than the other two species, which each pollinated only one species. This agrees with the observations of [20] stating that removal of generalist pollinators had the greatest effect on plant survival. Another potential application is on the movement of individuals between patches, which can be important to ecosystems [18]. Stochastic flux analysis can help to determine which movements have major consequences for the populations, as this makes it possible to monitor the movements at arbitrary time intervals as in our case study.

In the present paper, we have illustrated our approach on an hypothetical ecosystem model. Future work includes applying the methods here to study actual plant-pollinator systems by relying on field data, and also including other kinds of interactions between ecosystem species. In this respect questions of interest include the conditions that lead to stability or extinction, and identification of the keystone species. In our case study, also in order to factor for the non-modeled ecosystem context, we used a parameter estimation method in order to determine the rates of interactions. An alternative or complementary approach can be based on exploiting field data to assist the estimation of these rates. However, typically field data shows the rates at which the various pollinator species visit the plant species [7]. This poses other questions, as the available data does not reflect the quantitative effect of the visitations on their reproductive rates. Directions of future investigation include an investigation on stochastic simulations to be used to infer the reproductive rates from the field data.

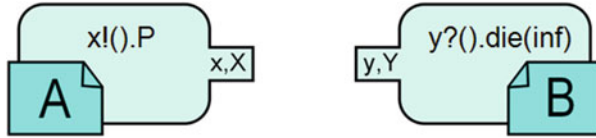
## Appendix A

BlenX shares features with the process algebra languages stochastic pi-calculus [25] and Beta-binders [4], it thus keeps a strong focus on the interactions of entities. Stochasticity is given by a continuous time Markov chain semantics, and it is realized by an efficient implementation of the Gillespie algorithm [8]. BlenX is a part of the software platform *COSBILAB*.

In BlenX, each individual of the modeled system is described as an abstract entity called *box*. Each box can interact with others via its connectivity interfaces called *binders*, and the result of the interactions and also other autonomous actions are determined by the user defined internal program of the box, which employs the stochastic semantics. As an example, consider the two boxes in Fig. 13, where each box has only one binder that is identified by its name, e.g., binder *x* and its type, e.g., *X*.

The internal program of a box describes the effect of the interactions and the autonomous actions that the box can undertake. Every time when such an action takes place with respect to the underlying stochastic semantics, the effect

of the action is reflected to the interfaces, and this way the new state of the box is computed. This is performed by the simulation engine by picking an action of the model with respect to the Gillespie algorithm by taking the rates of the available actions of all the boxes at that state. This results in a model behavior in the form of a sequence of model actions that can be read as a time series of the model individuals, i.e., boxes.



**Fig. 13.** Two BlenX boxes representing two interacting species A and B.

A BlenX model is written as two separate files, where the first file is the description of all the boxes of the model and their binders. The second file of the model contains a list of pairs of binder types together with their binding, unbinding, and interaction rates, which can all be 0. With respect to the compatibilities given by this list, binders can bind or unbind with binders of other boxes, or interact to exchange information communications with other boxes. As an example for this, consider the model given in Fig. 13, where the expression  $(X, Y, 0, 0, 1)$  indicates that the binders with types  $X$  and  $Y$  can interact with a rate 1. The third and fourth parameters of this expression state the binding and unbinding rates are 0.

The interaction of a predator  $A$  and its prey  $B$  can be described in a BlenX model with the boxes depicted in Fig. 13. The interaction rate, specified in the BlenX code, determines the rate of the predation being modeled. The internal program, which can be `nil`, describes this interaction and its consequences in terms of the *actions* the box can undertake. The `nil` action does nothing. Other stochastic *actions* that a BlenX box can perform are summarized as follows: a box can

- (i.) communicate with another box (or with itself) by performing an input action, e.g., `x?(message)` that is complementary to the output action, e.g., `x!(message)`, of the other box, or vice versa, and this way send or receive a message;
- (ii.) perform a stochastic `delay` action;
- (iii.) change (`ch`) the type of one of its interfaces;
- (iv.) eliminate itself by performing a `die` action;
- (v.) `expose` a new binder;
- (vi.) `hide` one of its binders;
- (vii.) `unhide` a binder which is hidden.

In addition to these actions, there are also other programming constructs available such as if-then statements and state-checks. For example, let us consider



the box  $A$  in Fig. 13. We can define the program  $P$  such that it changes the type  $X$  to  $Z$  if this box is bound to another species via its interface  $x$ :

$$\text{if } (x, X) \text{ and } (x, \text{bound}) \text{ then ch}(x, Z) \text{ endif}$$

As BlenX is a process algebra based language, internal programs can be written as compositions of actions by using algebraic composition operators in order to define increasingly complex behaviors. We can sequentially compose actions by using the prefix-operator, written as an infix dot. For example,  $\text{ch}(x, Z) \cdot \text{hide}(x) \cdot \text{nil}$  denotes a program that first performs change action and then hides the changed binder. Programs can be composed in parallel. Parallel composition (denoted by the infix operator  $|$ , for instance  $P|Q$ ) allows the description of programs, which may run independently in parallel and also synchronize on *complementary actions* (i.e., *input* and *output* over the same channel). Programs can also be composed by *stochastic choice*, denoted with the summation operator “+”. The sum of processes  $P$  and  $Q$ ,  $P + Q$  behaves either as  $P$  or as  $Q$ , determined by their stochastic rates, and selection of one discards the other forever.

## References

1. Cardelli, L., Caron, E., Gardner, P., Kahramanoğulları, O., Phillips, A.: A process model of Rho GTP-binding proteins. *Theoret. Comput. Sci.* **410**(33–34), 3166–3185 (2009)
2. Cardelli, L., Zavattaro, G.: Turing universality of the biochemical ground form. *Math. Struct. Comput. Sci.* **20**(1), 45–73 (2010)
3. DeAngelis, D.L., Gross, L.J.: *Individual-based Models and Approaches in Ecology*. Chapman and Hall, New York (1992)
4. Degano, P., Prandi, D., Priami, C., Quaglia, P.: Beta binders for biological quantitative experiments. *ENTCS* **164**(3), 101–117 (2005)
5. Dematté, L., Larcher, R., Palmisano, A., Priami, C., Romanel, A.: Programming biology in BlenX. *Syst. Biol. Signal. Netw.* **1**, 777–821 (2010)
6. Dematté, L., Priami, C., Romanel, A.: The BlenX language: a tutorial. In: Bernardo, M., Degano, P., Zavattaro, G. (eds.) *SFM 2008. LNCS*, vol. 5016, pp. 313–365. Springer, Heidelberg (2008)
7. Forup, M.L., Henson, K.S.E., Craze, P.G., Memmott, J.: The restoration of ecological interactions: plant-pollinator networks on ancient and restored heathlands. *J. Appl. Ecol.* **45**, 742–752 (2008)
8. Gillespie, D.T.: Exact stochastic simulation of coupled chemical reactions. *J. Phys. Chem.* **81**, 2340–2361 (1977)
9. Grimm, V.: Ten years of individual-based modelling in ecology: what have we learned and what could we learn in the future? *Ecol. Model.* **115**, 129–148 (1999)
10. Grimm, V., Berger, U., Bastiansen, F., Eliassen, S., Ginot, V., Giske, J., Goss-Custard, J., Grand, T., Heinz, S.K., Huse, G., Huth, A., Jepsen, J.U., Jørgensen, C., Mooij, W.M., Müller, B., Peer, G., Piou, C., Railsback, S.F., Robbins, A.M., Robbins, M.M., Rossmanith, E., Rügen, N., Strand, E., Souissim, S., Stillman, R.A., Vabø, R., Visser, U., DeAngelis, D.L.: A standard protocol for describing individual-based and agent-based models. *Ecol. Model.* **198**(1–2), 115–126 (2006)

11. Guerriero, M.L., Dudka, A., Underhill-Day, N., Heath, J.K., Priami, C.: Narrative-based computational modelling of the gp130/jak/stat signalling pathway. *BMC Syst. Biol.* **3**(1), 40 (2009)
12. Kahramanođulları, O.: Flux analysis in process models via causality. In: Proceedings of the 3rd Workshop from Biology to Concurrency and Back, vol. 19, pp. 20–39. EPTCS (2010)
13. Kahramanođulları, O., Cardelli, L.: An intuitive modelling interface for systems biology. *Int. J. Softw. Inform.* (2011) (in press)
14. Kahramanođulları, O., Lynch, J., Jordán, F.: CoSBIlab LIME: a language interface for stochastic dynamical modelling in ecology. *Environ. Model Softw.* **26**(5), 685–687 (2011)
15. Kahramanođulları, O., Jordán, F., Priami, C.: Composability: perspectives in ecological modeling. In: Horimoto, K., Nakatsui, M., Popov, N. (eds.) ANB 2010. LNCS, vol. 6479, pp. 136–148. Springer, Heidelberg (2012)
16. Kahramanođulları, O., Lynch, J.: Stochastic flux analysis of chemical reaction networks. *BMC Syst. Biol* **7**(133) (2013)
17. Kahramanođulları, O., Phillips, A., Vaggi, F.: Process modeling and rendering of biochemical structures: actin. In: Biomechanics of cells and tissues: experiments, models and simulations. Lecture Notes in Computational Vision and Biomechanics. vol. 9, Springer (2013)
18. Knight, T.M., McCoy, M.W., Chase, J.M., McCoy, K.A., Holt, R.D.: Trophic cascades across ecosystems. *Nature* **437**, 880–883 (2005)
19. Levin, S.A.: Ecosystems and the biosphere as complex adaptive systems. *Ecosystems* **1**(5), 431–436 (1998)
20. Memmott, J., Waser, N.M., Price, M.V.: Tolerance of pollination networks to species extinctions. *Proc. R. Soc. Lond., Ser. B, Biol. Sci.* **271**, 2605–2611 (2004)
21. Olf, H., Alonso, D., Berg, M.P., Eriksson, B.K., Loreau, M., Piersma, T., Rooney, N.: Parallel ecological networks in ecosystems. *Philos. Trans. R. Soc. B* **364**(1524), 1755–1779 (2009)
22. Phillips, A., Cardelli, L.: Efficient, correct simulation of biological processes in the stochastic pi-calculus. In: Calder, M., Gilmore, S. (eds.) CMSB 2007. LNCS(LNBI), vol. 4695, pp. 184–199. Springer, Heidelberg (2007)
23. Pimm, S.L.: *The Balance of Nature?*. The University of Chicago Press, Ecological Issues in the Conservation of Species and Communities (1991)
24. Powell, C.R., Boland, R.P.: The effects of stochastic population dynamics on food web structure. *J. Theor. Biol.* **257**(1), 170–180 (2009)
25. Priami, C.: Stochastic  $\pi$ -calculus. *Comput. J.* **38**(6), 578–589 (1995)
26. Priami, C.: Algorithmic systems biology. *Commun. ACM* **52**(5), 80–89 (2009)
27. Regev, A., Shapiro, E.: Cellular abstractions: cells as computation. *Nature* **419**, 343 (2002)
28. Shaffer, C.A., Zwolak, J.W., Randhawa, R., Tyson, J.J.: Modeling molecular regulatory networks with jigcell and pet. *Methods Mol. Biol.* **500**, 1–31 (2009)