Lineage grammars: describing, simulating and analyzing population dynamics

Adam Spiro[1,3]
Email: adam.spiro@weizmann.ac.il

Luca Cardelli[2]
Email: luca@microsoft.com

Ehud Shapiro[1,3,*]
Email: ehud.shapiro@weizmann.ac.il

[1] Department of Computer Science and Applied Mathematics, Weizmann Institute of Science, Rehovot, Israel

[2] Microsoft Research, Cambridge, UK

[3] Department of Biological Chemistry, Weizmann Institute of Science, Rehovot, Israel

[*] Corresponding author. Department of Computer Science and Applied Mathematics, Weizmann Institute of Science, Rehovot, Israel

Abstract

Background

Precise description of the dynamics of biological processes would enable the mathematical analysis and computational simulation of complex biological phenomena. Languages such as Chemical Reaction Networks and Process Algebras cater for the detailed description of interactions among individuals and for the simulation and analysis of ensuing behaviors of populations. However, often knowledge of such interactions is lacking or not available. Yet complete oblivion to the environment would make the description of any biological process vacuous. Here we present a language for describing population dynamics that abstracts away detailed interaction among individuals, yet captures in broad terms the effect of the changing environment, based on environment-dependent Stochastic Tree Grammars (eSTG). It is comprised of a set of stochastic tree grammar transition rules, which are context-free and as such abstract away specific interactions among individuals. Transition rule probabilities and rates, however, can depend on global parameters such as population size, generation count, and elapsed time.

Results

We show that eSTGs conveniently describe population dynamics at multiple levels including cellular dynamics, tissue development and niches of organisms. Notably, we show the utilization of eSTG for cases in which the dynamics is regulated by environmental factors, which affect the fate and rate of decisions of the different species. eSTGs are lineage grammars, in the sense that execution of an eSTG program generates the corresponding lineage trees, which can be used to analyze the evolutionary and developmental history of the biological system under investigation. These lineage trees contain a representation of the entire events history of the system, including the dynamics that led to the existing as well as to the extinct individuals.

Conclusions

We conclude that our suggested formalism can be used to easily specify, simulate and analyze complex biological systems, and supports modular description of local biological dynamics that can be later used as “black boxes” in a larger scope, thus enabling a gradual and hierarchical definition and simulation of complex biological systems. The simple, yet robust formalism enables to target a broad class of stochastic dynamic behaviors, especially those that can be modeled using global environmental feedback regulation rather than direct interaction between individuals.

Background

In recent years there has been a great interest in modeling and simulating various aspects of population dynamics in biological and ecological systems [1-4]. The increasing computational resources along with a deeper understanding of biological and ecological phenomena have led to the development of many languages for describing, analyzing and simulating concurrent stochastic processes. Many such languages specify Markovian dynamics and differ by level of abstraction, ease and complexity of the description and execution efficiency [5]. Two widely used formalisms are based on Chemical Reaction Networks (CRN) [6] and stochastic Process Algebras (PA) [7].

CRNs were originally used to describe chemical systems. A CRN description consists of a finite set of reactions acting on a finite number of species. Each reaction specifies the identity and stoichiometry of the reactants and products along with a rate constant. Many processes can be described using CRNs, for example, Predator-Prey models [8], Cellular cascade pathways [9], Cancer progression [10], Epidemics dynamics [11], and many others [1]. Each of these processes consists of a continuous interaction between individual species (the reactants) that occurs at a certain rate and produces a group of other individuals (the products, which may be empty) that can be of the same (autocatalytic) or of different type. The description of dynamical systems using CRN is relatively simple and can be used both for analytical solving and simulations. However, this approach neglects biological aspects of the described systems by treating each object (reactant or product) as a simple entity, which ignores its environmental context and structure. For example, many molecular objects maintain their overall identity while changing in specific attributes, such as chemical modification or location. When using a CRN abstraction such molecules cannot retain identity while changing state.

PAs, on the other hand, are a family of mathematical formalisms that were originally developed to model concurrent computer systems. They enable the abstraction and specification of communication and synchronization between a collection of processes by passing messages between them. One of the most well studied PA is the π -calculus, which has been shown to be very useful in describing a range of biological systems [7,12]. The language consists of processes that are mapped to real-world objects, and channels, which are mapped to communications and interactions between the different objects. A unique feature of the π -calculus allows to dynamically communicate new channels between the processes (this is termed *mobility*), which enables the objects to keep their identity while changing their internal states or interactions with other objects. This feature is more compatible with real biological and ecological scenarios and fits well to the way we think and observe these processes. It also allows one to abstract and specify the dynamics in a more accurate fashion. It has also been shown that this abstraction can be treated as an executable computer program, allowing to stochastically simulate any specified model [13].

Many tools have been developed in order to allow and simplify the use of mathematical modeling for the life-science community, and each one has its strengths and weaknesses [14-16]. There is no single formalism that has all the required features and choosing the appropriate one depends on the specific goals and resources of the modeler. Our goal in this work is to develop and formulate a simpler and practical tool for modeling and simulating the behavior and interaction of populations. We do so by extending the notion of Stochastic Tree Grammar (STG) [17] by incorporating both rates and probabilities to the transition rules. These can be dynamically updated by defining them as functions of the system's state, which includes global values such as current population size, generation count or elapsed time. In addition, we extend the system by allowing each individual to hold its own internal states which can change through inheritance. We later discuss implementation of stochastic simulation and the relation to Ordinary Differential Equations (ODE).

A prominent feature of the language is that it enables to stochastically produce possible lineage trees corresponding to single executions. These lineage trees contain a representation of the entire events history of the process, including the dynamics that led to the existing as well as to the extinct individuals. As opposed to standard approaches that output only the population size dynamics, our implementation also outputs the corresponding lineage trees, which can be used to analyze the evolutionary and developmental history of the process.

Recently, Vaughan et al. [16] presented the usage of CRNs as lineage grammars and used them to simulate phylogenetic trees. Although they enable to sample possible genealogies based on the defined reaction rules, they do not allow the specification and analysis of more complex behaviors such as feedback onto the dynamic rates and general inherited properties.

Throughout the paper, we demonstrate the usability of the language by presenting a wide range of examples that can be modeled and simulated using this approach. The examples show that the language can provide simple descriptions of systems from various domains. Example parameter values were taken from the literature when available or chosen arbitrarily in order to simplify the presentation.

Results and discussion

eSTG programs

Following is an example of an eSTG program for stem-cell differentiation [18]:

$$SC \xrightarrow{0.1\frac{events}{day}} \{SC, SC\}_{0.5} \mid \{Diff, Diff\}_{0.5}$$

$$Diff \xrightarrow{1\frac{event}{day}} \{Diff, Diff\}_{0.49} \mid \{\phi\}_{0.51}$$

In this example, *SC* (stem cells) divide symmetrically 0.1 times per day, while self-renewing or differentiating with the same probability (50%), and *Diff* (differentiated cells) can once a day either proliferate (with probability 49%) or die (with probability 51%).

Alternatively, one can define an average time to event *t* instead of a rate, which can be translated interchangeably into a rate using $r = \frac{1}{t}$. The above rules are then written:

$$SC \xrightarrow{10\ days} \{SC, SC\}_{0.5} \mid \{Diff, Diff\}_{0.5}$$

$$Diff \xrightarrow{1\ day} \{Diff, Diff\}_{0.49} \mid \{\phi\}_{0.51}$$

An execution of an eSTG program proceeds through the stochastic application of its transition rules on its state. An example execution of the program, on an initial 10 *SC* and 5 *Diff*, can be summarized by a cell lineage tree and population size graphs shown in Figure 1B and Figure 1C. In addition to single executions, eSTG can also be used for obtaining overall population statistics, for example, to calculate the average population size over time (Figure 1D) and the distribution of clone sizes (Figure 1E).

Figure 1 An example of the stem cell differentiation program execution. The program was executed up to simulation time 100 days. **(A)** Schematic representation of the eSTG rules (without rates and probabilities). **(B)** Population size over time of a specific execution. **(C)** Cell lineage tree of a specific execution (only one cell lineage tree out of the originating ***SC***s and ***Diffs*** is shown). **(D)** Average population size over time (calculated from 1000 stochastic executions). **(E)** Clone size distribution, which is the final population size derived from each initiating individual (calculated from 1000 stochastic executions).

Following is another example of an eSTG program for the Luria–Delbrück Model [19]:

$$SC \xrightarrow{0.1\frac{events}{day}} \{SC, SC\}_{0.5} \mid \{Diff, Diff\}_{0.5}$$

$$Diff \xrightarrow{1\frac{event}{day}} \{Diff, Diff\}_{0.49} \mid \{\phi\}_{0.51}$$

In this model, wild-type bacteria (*WT*) are randomly mutated (in the absence of selection) to form a resistant bacteria (*MUT*), thus the population size of mutated bacteria varies dramatically and is dependent on the timing in which the mutation has happened. Figure 2B and Figure 2C show specific executions of typical and rare lineage trees. Averaging over many executions can yield average population size (Figure 2D) and clone size distribution (Figure 2E).

Figure 2 An example of the Luria–Delbrück program execution. The program was executed from 1 ***WT*** to 100 cells. **(A)** Schematic representation of the eSTG rules. **(B)** Typical lineage tree execution where mutations do not occur early. **(C)** Rare lineage tree execution where a mutation occurs early. **(D)** Average population size over time (calculated from 1000 stochastic executions). **(E)** Clone size distribution (calculated from 1000 stochastic executions). In the rare events where the mutation happens early in the lineage, the clone size of the mutated population is large.

Internal states

We define internal states for each species as a vector of variables that can change, either deterministically or stochastically for each individual, with every execution of a rule. Internal states can be used to model inherited attributes, such as mutations or substance accumulation, or record historical events such as the number of generations, number of symmetrical/asymmetrical divisions, or time since historical events. We thus extend the basic rules defined above to include internal states which are functions of the predecessor's internal states. For example, extending the previous stem-cell differentiation scenario:

$$SC(\overrightarrow{MS} = \vec{x}_{MS}) \xrightarrow{10\ days} \left\{SC\left(\overrightarrow{MS} = f_{MS}(\vec{x}_{MS})\right), SC\left(\overrightarrow{MS} = f_{MS}(\vec{x}_{MS})\right)\right\}_{0.5} \mid$$
$$\left\{Diff(\overrightarrow{MS} = f_{MS}(\vec{x}_{MS})), Diff((\overrightarrow{MS} = f_{MS}(\vec{x}_{MS}))\right\}_{0.5}$$
$$Diff(\overrightarrow{MS} = \vec{x}_{MS}) \xrightarrow{1\ day} \left\{Diff(\overrightarrow{MS} = f_{MS}(\vec{x}_{MS})), Diff(\overrightarrow{MS} = f_{MS}(\vec{x}_{MS}))\right\}_{0.49} | \{\phi\}_{0.51}$$

In this example, we define a vector of *n* variables $\overrightarrow{MS} = (MS_1, \dots MS_n)$, which correspond to the number of repeats in *n* Microsatellite (MS) loci in the DNA [20]. In every cell division, the number of MS repeats for each locus changes according to the stochastic function f_{MS}, which can cause either a decrease or an increase of one repeat with probability *p* [21]:

$$f_{MS}(x) = \begin{cases} x+1 \, with\ probability\ \frac{p}{2} \\ x-1 \, with\ probability\ \frac{p}{2} \\ x \, otherwise \end{cases}$$

This simulated data can be used for example to evaluate the relationship between *n*, the number of MS, and the accuracy of phylogenetic reconstruction based on MS lengths of the tree (see [22,23] for details).

Another example for the use of internal states is the following program, which counts the number of generations since each differentiation event:

$$SC \xrightarrow{10\ days} \{SC, SC\}_{0.5} \mid \{Diff\,(Gen=1), Diff\,(Gen=1)\}_{0.5}$$
$$Diff\,(Gen = x) \xrightarrow{1\ day} \{Diff\,(Gen = x+1), Diff\,(Gen = x+1)\}_{0.49} \mid \{\phi\}_{0.51}$$

Figure 3 shows various distribution statistics of the internal state *Gen* over the population at different time points.

Figure 3 An example of generation counter internal state. Each species of the type holds an internal state called ***Gen*** which holds the number of cell divisions since the differentiation event. The histogram of the values over the entire population can be calculated at different time points (e.g. after 10, 50 and 100 days, shown in **(A)**, **(B)** and **(C)** respectively).

Other examples of internal states can be the counting of historical events (such as how many symmetric vs. asymmetric divisions a cell went through) or measuring the time since a certain event.

Probabilities and rates as functions

Population dynamics can change based on various conditions such as population size, internal or external changes, and elapsed time. A common phenomenon in population dynamics is the reaching of a homeostasis, meaning that at a certain point, the population size reaches a steady state.

A simple example is the growth of a species until reaching a target size. Consider the following parametric rule:

$$A \xrightarrow{r} \{A, A\}_p \mid \{\phi\}_{1-p}$$

Without feedback regulation on the population size, a setting of $p = \frac{1}{2}$ results in an extinction with probability 1 [24]. A simple regulation scheme is the logistic model [25]:

$$\frac{dN}{dt} = rN\left(1 - \frac{N}{K}\right)$$

where N is the population size, r is the growth rate and K is the target size (also termed carrying capacity). We can use the above parametric eSTG rule to model a logistic population growth by solving:

$\frac{dA}{dt} = \frac{dN}{dt}$ (we use A as the population size of the species A)

$$\frac{dA}{dt} = Arp - Ar(1-p) = rN(1 - \frac{N}{K})$$

For simplicity, r in the eSTG rule is the same as the r in the logistic model.

We then get:

$$2p - 1 = 1 - \frac{N}{K}$$

$$p = 1 - \frac{N}{2K}$$

Figure 4B and Figure 4C show the resulting dynamics (population size and lineage tree) starting from a single A of the following program (setting K = 100):

Figure 4 An example of dynamic population growth. An example of a simple proliferation with fate probabilities and rates that are functions of the population size. **(A)** Schematic representation of the eSTG rules. **(B)** Population size over time of a logistic growth starting from a single instance. **(C)** The corresponding lineage representation of the specific execution. **(D)** Population size over time of a production-removal growth starting from a single instance. **(E)** The corresponding lineage representation of the specific execution.

$$A \xrightarrow{r} \{A, A\}_p \mid \{\phi\}_{1-p}$$
$$p = 1 - \frac{A}{200}$$

In a different scenario, the growth is also regulated by the rate but is leading to the same steady state. Using the following production-removal equation [26]:

$$\frac{dN}{dt} = \beta - \alpha N$$

we can model the dynamics using the parametric eSTG by solving:

$$\frac{dA}{dt} = Arp - Ar\left(1 - p\right) = \beta - \alpha N$$
$$r = \frac{\beta - \alpha A}{\left(2p - 1\right)A}$$

The steady state of this system is $\frac{\beta}{\alpha}$ and for simplicity we limit p to be either 0 or 1, and set α = 1, β = 100. We thus define the following eSTG program:

$$A \xrightarrow{r} \{A, A\}_p \mid \{\phi\}_{1-p}$$
$$r = \frac{100 - A}{(2p - 1)A}$$
$$p = \begin{cases} 1 \ if \ A \leq 100 \\ 0 \ if \ A > 100 \end{cases}$$

Here, the rate is inversely dependent on the population size and the population is growing until reaching the steady state that is maintained by a feedback on p, which causes either a proliferation (p = 1) or death (p = 0). Figure 4D and Figure 4E show the resulting dynamics starting from a single A.

Another interesting scenario is described in [27], where an optimal development of the intestinal crypts is analysed. In the first stage, stem-cells are quickly amplified using self-replicating symmetric divisions, and after reaching the target size they differentiate asymmetrically into stem-cells and differentiated cells. We can describe this scenario using the following rules:

$$SC \xrightarrow{r_1} \{SC, SC\}_{p_1} \mid \{SC, Diff\}_{ow}$$

$$Diff \xrightarrow{r_2} \{Diff, Diff\}_{p_2} \mid \{\phi\}_{ow}$$

$$p_1 = 1 \cdot \text{until} \cdot |SC|_{Time=t} = |SC|_{Target}$$

$$p_1 = 0 \cdot \text{until} \cdot |Diff|_{Target=t} = |Diff|_{Target}$$

where $|X|_{Time\,=\,t}$ is the population size of species X at time t and $|X|_{Target}$ is the target population size of X. Although not described in [27], we continue the scenario with homeostasis by solving:

$$SC + \frac{dSC}{dt} = SC + r_1 p_1 SC = |SC|_{Target}$$

$$\begin{aligned} Diff + \frac{dDiff}{dt} &= Diff + r_1(1-p_1)SC + r_2 p_2 Diff - r_2(1-p_2)Diff \\ &= |Diff|_{Target} \end{aligned}$$

We thus extend the program with the following:

$$p_1 = \frac{|SC|_{Target} - |SC|}{|SC| \cdot r_1} after\ |Diff| = |Diff|_{Target}$$

$$p_2 = \frac{|Diff|_{Target} - |Diff| - r_1|SC|(1-p_1) + r_2|Diff|}{2r_2 \cdot |Diff|} after\ |Diff| = |Diff|_{Target}$$

Figure 5 shows simulation results of a specific execution.

Figure 5 Rules for optimal development of the crypt. Simulation results of the rules for optimal development of the crypt (see main text). The rules are executed with $\boldsymbol{r}_1 = 1.07$, $\boldsymbol{r}_2 = 1$, $|\boldsymbol{SC}|_{\boldsymbol{Time}\,=\,0} = 1$, $|\boldsymbol{Diff}|_{\boldsymbol{Time}\,=\,0} = 0$, $|\boldsymbol{SC}|_{\boldsymbol{Target}} = 10$, $|\boldsymbol{Diff}|_{\boldsymbol{Target}} = 50$ (values are taken from [27]). Shown are execution results for two time windows starting with one *SC*. **(A)** Schematic representation of the eSTG rules. **(B)** Population size for simulation time of 10 days. The beginning of the process is shown where the switch between and $\boldsymbol{p}_1 = 0$ occurs at around time . **(C)** The corresponding lineage representation of the specific execution. **(D)** Population size for simulation time of 50 days. Shown is the homeostatic phase that occurs after reaches $|\boldsymbol{Diff}|_{\boldsymbol{Target}}$ at around time $\boldsymbol{t} = 6$. **(E)** The corresponding lineage representation of the specific execution. It is interesting to observe the 10 clones that are maintained by the 10 *SCs*.

An example from a different regime is the predator/prey model of Lotka-Volterra [8]. It describes the interaction dynamics between two species using two ODEs:

$$\frac{dPrey}{dt} = Prey(c_1 - c_2 Predator)$$

$$\frac{dPredator}{dt} = -Predator(c_3 - c_2 Prey)$$

where c_i are parameters. These equations are usually translated into the following mass action kinetic reactions:

$$Prey \xrightarrow{c_1} 2Prey,$$

$$Predator + Prey \xrightarrow{c_2} 2Predator$$

$$Predator \xrightarrow{c_3} \phi$$

Since eSTG has only context-free transitions, we convert the second reaction into two unimolecular reactions while preserving the 2nd order rate (see Methods for a general method to convert CRNs to unimolecular reactions while preserving the same underlined ODEs). The new reactions and their rates are described in Table 1. We note that although these new reactions are not identical to the original ones, they are still in agreement with the ODEs described above. The model can be described using the following parameterized eSTG program:

Table 1 Lotka-Volterra unimolecular representation

Reaction	Global Rate
$Prey \to 2Prey$	$c_1 \cdot Prey$
$Prey \to \phi$	$c_2 \cdot Prey \cdot Predator$
$Predator \to 2Predator$	$c_2 \cdot Prey \cdot Predator$
$Predator \to \phi$	$c_3 \cdot Predator$

$$Prey \xrightarrow{r_1} \{Prey, Prey\}_{p_1} | \{\phi\}_{ow}$$
$$Predator \xrightarrow{r_2} \{Predator, Predator\}_{p_2} | \{\phi\}_{ow}$$
$$r_1 = c_1 + c_2 \cdot |Predator|$$
$$r_2 = c_2 \cdot |Prey| + c_3$$
$$p_1 = \frac{c_1}{r_1}$$
$$p_2 = \frac{c_2 \cdot |Prey|}{r_2}$$

Figure 6 shows an example execution of the program.

Figure 6 An example execution of the Lotka-Volterra scheme. An output example of the executed program using $\boldsymbol{c}_1 = 2$, $\boldsymbol{c}_2 = 0.01$, $\boldsymbol{c}_3 = 5$, $|\boldsymbol{Prey}|_{\boldsymbol{Time}=0} = |\boldsymbol{Predator}|_{\boldsymbol{Time}=0} = 900$. **(A)** Population size as a function of time. **(B)** A lineage tree of one of the 900 originating preys. **(C)** A lineage tree of one of the 900 originating predators. Both (B) and (C) exhibit the characteristic bottleneck phenomenon, where most lineages get extinct.

The role of different feedback strategies on the control of organ and tissue growth can be investigated through the rates and probabilities of cellular decisions. Lander et al. [28] suggest two types of feedback strategies for the Olfactory Epithelium, one on the rate of division and the other on the probability of self-renewal (while keeping a constant division rate). They show that a feedback control onto the probability is a much more effective strategy for steady-state robustness and rapid regeneration.

The two strategies can be described using the following eSTG program (SC – stem cells, INP - Immediate Neuronal Precursor, ORN - olfactory receptor neuron):

$$SC \xrightarrow{v_0} \{SC, SC\}_{p_0} \mid \{INP, INP\}_{1-p_0}$$

$$INP \xrightarrow{v_1} \{INP, INP\}_{p_1} \mid \{ORN, ORN\}_{1-p_1}$$

$$ORN \xrightarrow{d} \{0\}_1$$

and the two feedback strategies are implemented by updating the parameters.

Strategy 1: Feedback onto the probability

$p_1 = \frac{p_1}{1+g\cdot|ORN|}$ where is a constant.

Strategy 2: Feedback onto the rate:

$v_1 = \frac{v_1}{1+h\cdot|ORN|}$ where h is a constant

Figure 7 shows possible executions generated using the two suggested strategies.

Figure 7 Scenarios for feedback regulation. (A) Schematic representation of the eSTG rules. Left plots - Feedback regulation onto the probability, where population size of an example execution, average population size over 1000 executions and an example of a lineage tree starting from a single are shown (**B**,**C**, and **D** respectively). Execution started with 10 ***SC****s*, $r_0 = 0.506, p_0 = 0.5, r_1 = 1, p_{1_{t=0}} = 0.942, d = 0.0138, g = 0.0449$, simulation time: 10. Right plots - Feedback regulation onto the rate, where population size of an example execution, average population size over 1000 executions and an example of a lineage tree starting from 10 ***SC****s* are shown (E,F, and G respectively). Execution started with 10 ***SC****s* and 200 INP, $r_0 = 0.128, p_0 = 0.5, r_1 = 1, p_{1_{t=0}} = 0.495, d = 0.0372, h = 0.0734$, simulation time: 20 (values are taken from [28]).

Possible extensions

Compartments

In many cases the population moves stochastically between different compartments, where each compartment corresponds to a different environment and different resources. Extending the language to include compartments allows one to define the same transition rules for species from the same type but different rates and probabilities, depending on the physical location of the individual. The system's state is then extended to include the population size in each compartment. In addition to the regular transition rules, one also needs to define rules for the migration of each species between each two compartments.

Individual’s probabilities and rates as functions

Defining probabilities and rates for each individual separately is not recommended due to heavy computational requirements when implementing such a scenario, however, an extension of the language can support such a definition. In this case we can allow the probabilities and rates of each individual to be also dependent on its internal states. This allows each individual to have a distinct stochastic value of its probabilities and transition rates. For example, we can define a more sophisticated predator/prey model where the probability of reproduction is dependent on the individual’s age (or weight) which is represented as an internal state, or define the proliferation dynamics of a cell based on its mutations (represented as internal states).

Conclusions

Stochastic simulation is a powerful tool to execute a complicated modeling system for which a closed form analytical solution is not possible. In addition, a simulation can generate a sample of representative scenarios that can be used for further analysis or as inputs to other programs. The complexity of natural phenomena requires a formal description framework which on one hand should be rich enough to capture the complexity and dynamics of the system and on the other hand will be compact and simple so it can be widely used by a broad community and could be implemented efficiently. There are many systems that are purely generative and derive their core results by ignoring interactions (e.g. L-Systems [29] and branching processes [24]). Although the assumption of independence enables certain analytical techniques, it precludes the ability to model processes and lineages that evolve through complex interactions between individuals and their environment. In order to allow both generativity and interaction, systems such as PA and CRN are more suitable. As described in [3], the trend towards individual-based stochastic models carries many advantages; they are easier to construct, more intuitive and can predict richer phenomena than population level models. In addition, it is possible to deduce population level conclusions (such as the underlined ODEs, see Methods) from the stochastic model. The presented formalism does not offer a new modeling approach in the sense that eSTG programs can be translated interchangeably into other languages (see Methods). Instead, the suggested eSTG language formalism allows a simpler description and specification of complex stochastic dynamics of individual entities. As demonstrated by the host of examples provided, these may include population level feedback from the current system’s state (either population size, internal or external factors) onto the rates and probabilities of the different species. In addition, eSTG, as a lineage grammar also enables the representation and analysis of historical events including those of extinct sub-lineages and transitional time points. Derivation trees produced by simulations can be examined for consistency with specific biological hypotheses [22,30], so that eSTG models can be validated or falsified on the basis of the trees that they generate.

The language can also be used as a basis for inference and learning of the system’s governing rules, described in the eSTG formalism as the transition rules and the underlying rates and probabilities as functions of the system’s state. The question of parameter inference from biological data is an active area of study [31-34]. In our context, biological knowledge inferred from experimentally-obtained trees [22,30,35-43] could be used in order to infer the corresponding lineage grammars [17,44,45]. This will allow the use of computers and computing resources in order to gain new biological insights. This is a great challenge, especially given noise and hidden variables, and is a subject of our future work. We hope that the development of theoretical models and tools, such as the one presented here, will facilitate research in this important direction.

Methods

Stochastic simulation

eSTG programs can be naturally simulated by the well-known Gillespie stochastic simulation algorithm [6]. Gillespie's implementation uses the rates of all possible reactions and chooses stochastically the next reaction by assuming that the time to the next reaction is exponentially distributed with rate parameters corresponding to the reaction rates.

A rule of the form:

$$A \xrightarrow{r} \{S_1\}_{p_1} \middle| \{S_2\}_{p_2} \middle| \dots \middle| \{S_{n-1}\}_{p_{n-1}} \middle| \{S_n\}_{p_n}$$

can be converted into *n* separated reaction rules:

$$A \xrightarrow{r \cdot p_i} S_i, \forall i = 1..n$$

and thus existing implementations of the Gillespie algorithm can be used to determine the next reaction and the time interval. Applying these rules to build the lineage tree is described in the Operational semantics section.

The code that was used to generate the examples in this paper will be made available as an open source tool and is currently under preparation for publication.

Equivalence and conversion to other languages

In this section we compare the expressiveness of eSTG to 4 other families:

1. maODE: Ordinary Differential Equations arising from mass-action kinetics.
2. maCRN: Chemical Reaction Networks with mass-action kinetics.
3. gCRN: Chemical Reaction Networks with general rate kinetics.
4. U-gCRN: Unimolecular Chemical Reaction Networks with general rate kinetics.

An maCRN is a chemical reaction network where each reaction has an associated rate constant, and where the instantaneous rate of a reaction is determined by the product of the rate constant with the instantaneous concentrations of the reagents. It is known that an maCRN under that mass-action law produces a system of ODEs with a special structure, here called an maODE system. In an maODE system each right-hand-side of each differential equation for species has the form of a polynomial over the set of species, where each monomial with a negative sign has *s* as a factor (raised to some non-zero power). Conversely each maODE determines a canonical maCRN that has that maODE as its kinetics. Therefore there are canonical translations back and forth between maODEs and maCRNs [46].

A gCRN is instead a Chemical Reaction Network where each reaction has an associated rate function from current or past system states to changes of concentrations. The instantaneous rate of a reaction is then given immediately by its rate function without further considerations; the class of ODEs that a gCRN may generate depends on the class of rate functions that are available.

A U-gCRN is a special case of a gCRN where all the reactions are unimolecular. For sufficiently powerful rate functions it is possible to have a (nominally) unimolecular reaction depend on the concentrations of other species, so that U-gCRN is in fact as expressive as gCRN. For example, an maCRN reaction $A\xrightarrow{r}B$ can be translated to the gCRN reaction $A\xrightarrow{r\cdot[A]}B$ where $[A]$ is the instantaneous concentration of A, and an maCRN reaction $A+B\xrightarrow{r}C$ can be translated to a gCRN reaction $A+B\xrightarrow{r\cdot[A]\cdot[B]}C$ or to two U-gCRN reactions $A\xrightarrow{r\cdot[A]\cdot[B]}C$ and $B\xrightarrow{r\cdot[A]\cdot[B]}0$.

The family of population dynamics specifications that can be described using basic eSTG is equivalent to U-gCRN. A U-gCRN reaction $A\xrightarrow{r}B_1+\cdots+B_n$ can be translated into an eSTG reaction $A\xrightarrow{r}\{B_1,\dots,B_n\}_{1.0}$, and conversely an eSTG reaction $A\xrightarrow{r}\{B_{1,1},\dots\}_{\mathrm{p}_1}|\dots|\{B_{n,1},\dots\}_{\mathrm{p}_n}$ can be translated into a set of U-gCRN reactions $A\xrightarrow{r\cdot\mathrm{p}_1}B_{1,1}+\cdots,\dots,A\xrightarrow{r\cdot\mathrm{p}_n}B_{n,1}+\cdots$.

The U-gCRN form of eSTGs implies that one must make choices in modelling: the main species that are the focus of a model, and occur in the left-hand side of productions, will be reflected in the generated lineage trees, but auxiliary species that appear only in the rate laws will not, even when those would be considered as equal in a model based on bimolecular interactions.

Operational semantics

We will start with basic definitions for the semantics of Lineage Trees. Paths in a tree are represented as finite sequences of natural numbers $\pi = n_1, \dots, n_m \in N^*$ (star means finite sequence, with *nil* as the empty sequence, and as sequence concatenation). Each number *n* in a path represents the child of a node, starting from the root. Nodes in a tree are labeled by an alphabet $S_0 = S \cup \{0\}$ consisting of species in *S* and a distinguished symbol (the "dead" leaf).

Definition: a *tree* is a partial function in $\mathbb{N}^* \to S_0$, from paths in $\mathbb{N}^*$ to label nodes in , whose domain is non-empty and prefix-closed (that is, $L(\pi_1, \pi_2)$ defined ⇒ $L(\pi_1)$ defined).

Definition: A *leaf* in a tree is a maximal path - one such that $L(\pi)$ is defined and there is no $\pi' \neq nil$ where is defined. We also say that π, B is a (-labeled) leaf in if π is a leaf in and $L(\pi) = B$.

Definition: A *lineage tree* is a tree where each path π such that $L(\pi) = 0$ is a leaf. is the set of such trees.

By these definitions, a tree is a non-empty set of paths and each node has a "unique label" which is the path π that leads to it. A root-only tree is a function from *nil* to some species .

Next, we use the -calculus notation for the definitions of lineage tree operators (if then we write $f = \lambda x.\ b$). We use the element for partially defined functions:

1. The lineage tree with just one dead leaf:

$$0 = \lambda x. if\ x = nil\ then\ 0\ else\ undef$$

2. The lineage tree with root $A \in S$ and children L_i, for $A \neq 0$ and $n >= 0$:

$$A(L_1,...,L_n) = \lambda x.\ if\ \ x = nil\ \ then\ A\ else\ if\ \ x = 1,\pi\ then\ L_1(\pi) ...\ else\ if\ \ x = n,\pi\ then\ L_n(\pi)\ else\ undef$$

Where for $n = 0$, $A = A()$ is a "live" leaf.

3. The *leaf-extension* operator $L, \pi, A \lhd (B_1, \ldots B_n)$, which is defined if π is an -labeled live leaf in $(L(L(\pi) = A \neq 0))$, and , and $B_1 \ldots B_n \in S_0$:

$$L,\pi, A \lhd (B1,...Bn) = \lambda x.\ if\ \ x = \pi,1\ then\ B_1, ..., else\ if\ \ x = \pi, n\ then\ B_n,\ else\ L(x)$$

For example, by the above definitions a tree with root and with children $B_1, \ldots, B_n$ which are all leaves can be written as the expression , representing a function that given the sequence *nil* returns the label C, given the sequence returns the label B_i, and is otherwise undefined. Similarly, the expression $C()$, *nil*, $C \lhd (B_1, \ldots, B_n)$ represents the tree $C()$ where the leaf is extended into a node with children ; this is then the same as the tree $C(B_1(), \ldots, B_n())$.

A collection of eSTG reactions describes a way of generating and transforming lineage trees. We now describe how each eSTG reaction transforms a lineage tree into new lineage trees. More precisely, since eSTG reactions are stochastic/probabilistic, how each reaction produces a *measure* of new lineage trees, where each new lineage tree is associated with its rate of occurrence.

Definition: A *measure* is a function from finite tuples of lineage trees to non-negative reals, with operators:

$$d\left(r,(L_1,\ldots,L_n)\right) = \lambda x. if\ x = (L_1,\ldots,L_n) then\, r\, else\, 0$$

the singleton measure, which measures as r and everything else as 0;

$$M_1 + \ldots + M_m = \lambda x. M_1(x) + \ldots + M_m(x)$$

the sum measure, with ;

$$L,\pi,B \lhd M = \ \lambda x. if\ x = L,\pi,B \lhd \left(L_1,\ldots,L_n\right) then\, M\left(L_1,\ldots,L_n\right) else\, 0$$

the leaf-extension measure, where π, B is a leaf in L; this is a function in . This is the measure such that any extended tree of the form for some $L_1, \ldots, L_n$ receives the measure $M(L_1, \ldots, L_n)$.

For example, $C(), nil, C \lhd (d(r,(D(), E())) + d(s, F())) = d(r, C(D(), E())) + d(s, C(F()))$ because $C(D(), E())$ has the shape $C(), nil, C \lhd (D(), E())$ and so it receives measure , and $C(F())$ has shape $C(), nil, C \lhd (F())$ and so it receives measure s.

We are now ready to define the effect of a set of eSTG reactions $\boldsymbol{S}$ on lineage trees. This is given as a *reduction* relation $\boldsymbol{R}$ between lineage trees and measures. We write (L *reduces to* M) for $(L, M) \in \boldsymbol{R}$, where $\boldsymbol{R}$ is defined as the smallest relation satisfying the following rule:

$$\begin{gathered} \textit{if } L \textit{ is a lineage tree and } \pi, B \textit{ is a leaf in } L \\ \textit{and } B \xrightarrow{r} \{M_1\}_{p_1} \mid \ldots \mid \{M_m\}_{p_m} \textit{ is a reaction in } \boldsymbol{S} \ (\textit{the only one for } B) \\ \textit{then } L \to L, \pi, B \lhd (d(r \cdot p_1, M_1) + \ldots + d(r \cdot p_m, M_m)) \end{gathered}$$

This rule prescribes, for example, how to carry out a simulation of a set of eSTG reactions given an initial lineage tree: at each step apply the rule above to all applicable reactions and tree leaves, sum all the measures so obtained, and sample a new lineage tree according to the resulting measure.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

AS and ES conceived the project. AS carried out all the simulations and wrote the manuscript. LC and ES provided project oversight, formal analyses, and contributed to drafting of the manuscript and final approval. All the authors read and approved the final manuscript.

Acknowledgements

This research was supported by The European Union FP7-ERC-AdG Foundation and by the Kenneth and Sally Leafman Appelbaum Discovery Fund. Ehud Shapiro is the Incumbent of The Harry Weinrebe Professorial Chair of Computer Science and Biology.

References

1. Wilkinson DJ: *Stochastic modelling for systems biology.* 2nd edition. London: Chapman & Hall/CRC Press; 2011.

2. Wilkinson DJ: **Stochastic modelling for quantitative description of heterogeneous biological systems.** *Nat Rev Genet* 2009, **10**(2)**:**122–133.

3. Black AJ, McKane AJ: **Stochastic formulation of ecological models and their applications.** *Trends Ecol Evol* 2012, **27**(6)**:**337–345.

4. Hucka M, Finney A, Sauro HM, Bolouri H, Doyle JC, Kitano H, Arkin AP, Bornstein BJ, Bray D, Cornish-Bowden A, Cuellar AA, Dronov S, Gilles ED, Ginkel M, Gor V, Goryanin II, Hedley WJ, Hodgman TC, Hofmeyr JH, Hunter PJ, Juty NS, Kasberger JL, Kremling A, Kummer U, Novère NL, Loew LM, Lucio D, Mendes P, Minch E, Mjolsness ED, Nelson MR, Nielsen PF, Sakurada T, Schaff JC, Shapiro BE, Shimizu TS, Spence HD, Stelling J, Takahashi K, Tomita M, Wagner J, Wang J, Forum SBML: **The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models.** *Bioinformatics* 2003, **19**(4)**:**524–531.

5. Henzinger T, Jobstmann B, Wolf V: **Formalisms for Specifying Markovian Population Models.** *Int J Found Comput Sci* 2011, **22**(04)**:**823–841.

6. Gillespie DT: **Stochastic Simulation of Chemical Kinetics.** *Annu Rev Phys Chem* 2007, **58**(1)**:**35–55.

7. Regev A, Silverman W, Shapiro E: **Representation and simulation of biochemical processes using the π-calculus process algebra.** *Pac Symp Biocomput* 2001, **6:**459–470.

8. Fujii T, Rondelez Y: **Predator–prey molecular ecosystems.** *ACS Nano* 2012, **7**(1)**:**27–34.

9. Kholodenko BN: **Negative feedback and ultrasensitivity can bring about oscillations in the mitogen-activated protein kinase cascades.** *Eur J Biochem* 2000, **267**(6)**:**1583–1588.

10. Boman BM, Wicha MS, Fields JZ, Runquist OA: **Symmetric division of cancer stem cells–a key mechanism in tumor growth that should be targeted in future therapeutic approaches.** *Clin Pharmacol Ther* 2007, **81**(6)**:**893–898.

11. Alonso D, McKane AJ, Pascual M: **Stochastic amplification in epidemics.** *J R Soc Interface* 2007, **4**(14)**:**575–582.

12. Phillips A, Cardelli L: **Efficient, Correct Simulation of Biological Processes in the Stochastic Pi-calculus**. In *Computational Methods in Systems Biology*, Volume 4695. Edited by Calder M, Gilmore S. Berlin Heidelberg: Springer; 2007:184–199.

13. Regev A, Shapiro E: *The π-calculus as an abstraction for biomolecular systems. Modelling in Molecular Biology.* Berlin-Heidelberg: Springer; 2004:219–266.

14. Ghosh S, Matsuoka Y, Asai Y, Hsin KY, Kitano H: **Software for systems biology: from tools to integrated platforms.** *Nat Rev Genet* 2011, **12**(12)**:**821–832.

15. Machado D, Costa RS, Rocha M, Ferreira EC, Tidor B, Rocha I: **Modeling formalisms in Systems Biology.** *AMB Express* 2011, **1:**45.

16. Vaughan TG, Drummond AJ: **A stochastic simulator of birth-death master equations with application to phylodynamics.** *Mol Biol Evol* 2013, **30**(6)**:**1480–1493.

17. Gonzalez RC, Thomason MG: *Syntactic pattern recognition: An introduction.* ; 1978.

18. Potten CS: *Stem cells.* London, San Diego: Academic Press; 1997.

19. Luria SE, Delbrück M: **Mutations of Bacteria from Virus Sensitivity to Virus Resistance.** *Genetics* 1943, **28**(6)**:**491–511.

20. Weber JL, Wong C: **Mutation of human short tandem repeats.** *Hum Mol Genet* 1993, **2**(8)**:**1123–1128.

21. Valdes AM, Slatkin M, Freimer N: **Allele frequencies at microsatellite loci: the stepwise mutation model revisited.** *Genetics* 1993, **133**(3)**:**737–749.

22. Hucka M, Finney A, Sauro HM, Bolouri H, Doyle JC, Kitano H, Arkin AP, Bornstein BJ, Bray D, Cornish-Bowden A, Cuellar AA, Dronov S, Gilles ED, Ginkel M, Gor V, Goryanin II, Hedley WJ, Hodgman TC, Hofmeyr JH, Hunter PJ, Juty NS, Kasberger JL, Kremling A, Kummer U, Novère NL, Loew LM, Lucio D, Mendes P, Minch E, Mjolsness ED, Nelson MR, Nielsen PF, Sakurada T, Schaff JC, Shapiro BE, Shimizu TS, Spence HD, Stelling J, Takahashi K, Tomita M, Wagner J, Wang J, Forum SBML: *Bioinformatics* 2003, **19**(4)**:**524–531.

23. Chapal-Ilani N, Maruvka YE, Spiro A, Reizel Y, Adar R, Shlush LI, Shapiro E: **Comparing algorithms that reconstruct cell lineage trees utilizing information on microsatellite mutations.** *PLoS Comput Biol* 2013, **9**(11)**:**e1003297.

24. Haccou P, Jagers P, Vatutin VA: *Branching Processes: Variation, Growth and Extinction of Populations.* Cambridge: Cambridge University Press; 2005.

25. Vandermeer J: **How populations grow: the exponential and logistic equations.** *Nat Educ Knowl* 2010, **1**(8)**:**1.

26. Hart Y, Antebi YE, Mayo AE, Friedman N, Alon U: **Design principles of cell circuits with paradoxical components.** *Proc Natl Acad Sci U S A* 2012, **109**(21)**:**8346–8351.

27. Itzkovitz S, Blat IC, Jacks T, Clevers H, van Oudenaarden A: **Optimality in the development of intestinal crypts.** *Cell* 2012, **148**(3)**:**608–619.

28. Lander AD, Gokoffski KK, Wan FY, Nie Q, Calof AL: **Cell lineages and the logic of proliferative control.** *PLoS Biol* 2009, **7**(1)**:**e15.

29. Lindenmayer A: **Mathematical models for cellular interactions in development. I. Filaments with one-sided inputs.** *J Theor Biol* 1968, **18**(3)**:**280–299.

30. Reizel Y, Chapal-Ilani N, Adar R, Itzkovitz S, Elbaz J, Maruvka YE, Segev E, Shlush LI, Dekel N, Shapiro E: **Colon stem cell and crypt dynamics exposed by cell lineage reconstruction.** *PLoS Genet* 2011, **7**(7)**:**e1002192.

31. Drummond AJ, Rambaut A, Shapiro B, Pybus OG: **Bayesian coalescent inference of past population dynamics from molecular sequences.** *Mol Biol Evol* 2005, **22**(5)**:**1185–1192.

32. Ellison AM: **Bayesian inference in ecology.** *Ecol Lett* 2004, **7**(6)**:**509–520.

33. Buckland ST, Newman KB, Fernández C, Thomas L, Harwood J: **Embedding population dynamics models in inference.** *Statistical Science* 2007, **22:**44–58.

34. Reinker S, Altman R, Timmer J: **Parameter estimation in stochastic biochemical reactions.** *IEE Proc-Syst Biol* 2006, **153**(4)**:**168–178.

35. Carlson CA, Kas A, Kirkwood R, Hays LE, Preston BD, Salipante SJ, Horwitz MS: **Decoding cell lineage from acquired mutations using arbitrary deep sequencing.** *Nat Methods* 2012, **9**(1)**:**78–80.

36. Reizel Y, Itzkovitz S, Adar R, Elbaz J, Jinich A, Chapal-Ilani N, Maruvka YE, Nevo N, Marx Z, Horovitz I, Wasserstrom A, Mayo A, Shur I, Benayahu D, Skorecki K, Segal E, Dekel N, Shapiro E: **Cell lineage analysis of the mammalian female germline.** *PLoS Genet* 2012, **8**(2)**:**e1002477.

37. Segev E, Shefer G, Adar R, Chapal-Ilani N, Itzkovitz S, Horovitz I, Reizel Y, Benayahu D, Shapiro E: **Muscle-bound primordial stem cells give rise to myofiber-associated myogenic and non-myogenic progenitors.** *PLoS One* 2011, **6**(10)**:**e25605.

38. Siegmund KD, Marjoram P, Woo YJ, Tavaré S, Shibata D: **Inferring clonal expansion and cancer stem cell dynamics from DNA methylation patterns in colorectal cancers.** *Proc Natl Acad Sci U S A* 2009, **106**(12)**:**4828–4833.

39. Frumkin D, Wasserstrom A, Itzkovitz S, Stern T, Harmelin A, Eilam R, Rechavi G, Shapiro E: **Cell lineage analysis of a mouse tumor.** *Cancer Res* 2008, **68**(14)**:**5924–5931.

40. Salipante SJ, Thompson JM, Horwitz MS: **Phylogenetic fate mapping: theoretical and experimental studies applied to the development of mouse fibroblasts.** *Genetics* 2008, **178**(2)**:**967–977.

41. Wasserstrom A, Adar R, Shefer G, Frumkin D, Itzkovitz S, Stern T, Shur I, Zangi L, Kaplan S, Harmelin A, Reisner Y, Benayahu D, Tzahor E, Segal E, Shapiro E: **Reconstruction of cell lineage trees in mice.** *PLoS One* 2008, **3**(4)**:**e1939.

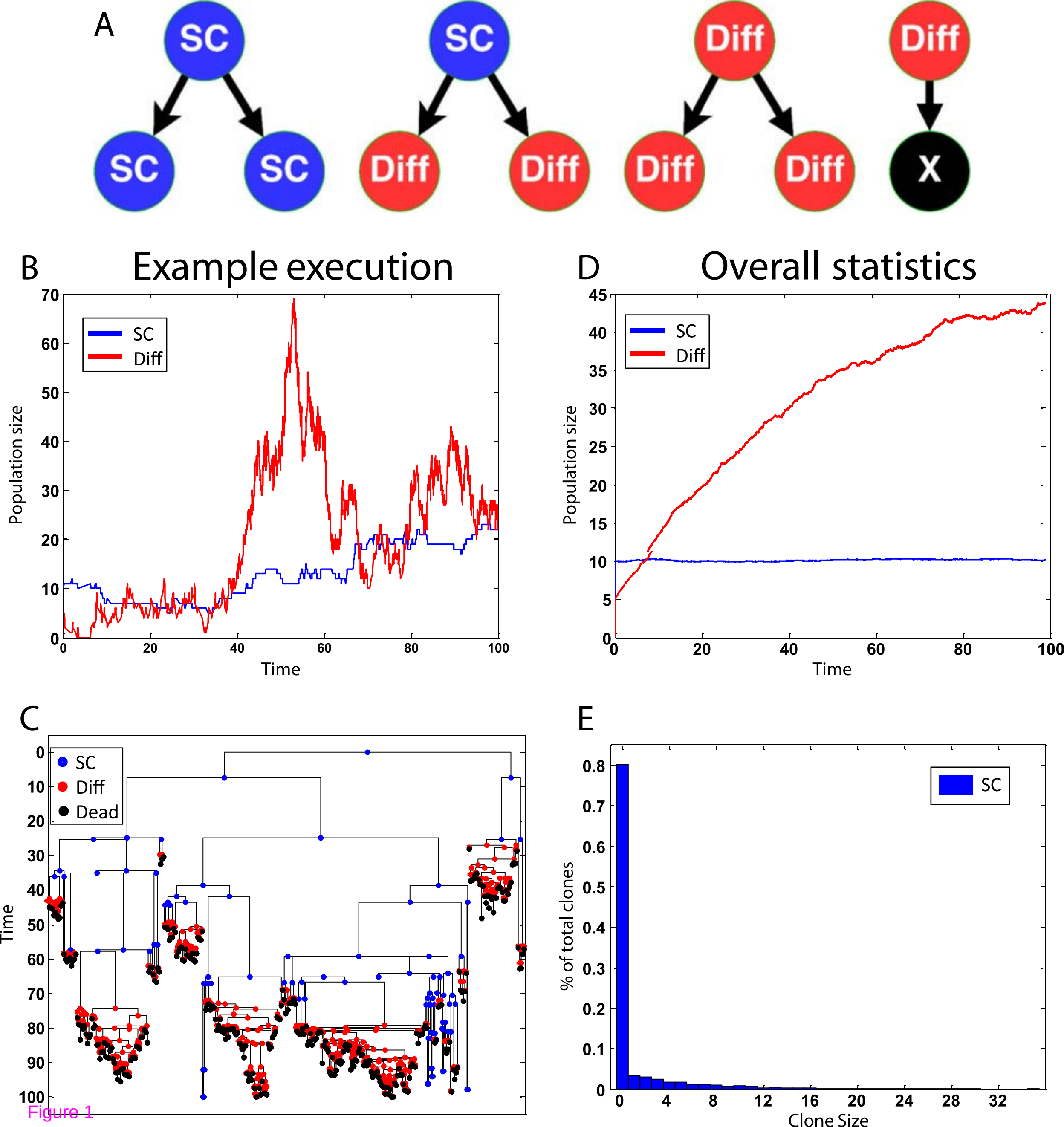
42. Salipante SJ, Horwitz MS: **Phylogenetic fate mapping.** *Proc Natl Acad Sci U S A* 2006, **103**(14)**:**5448–5453.

43. Navin N, Kendall J, Troge J, Andrews P, Rodgers L, McIndoo J, Cook K, Stepansky A, Levy D, Esposito D, Muthuswamy L, Krasnitz A, McCombie WR, Hicks J, Wigler M: **Tumour evolution inferred by single-cell sequencing.** *Nature* 2011, **472**(7341)**:**90–94.

44. Carrasco RC, Oncina J, Calera-Rubio J: **Stochastic inference of regular tree languages.** *Mach Learning* 2001, **44**(1–2)**:**185–197.

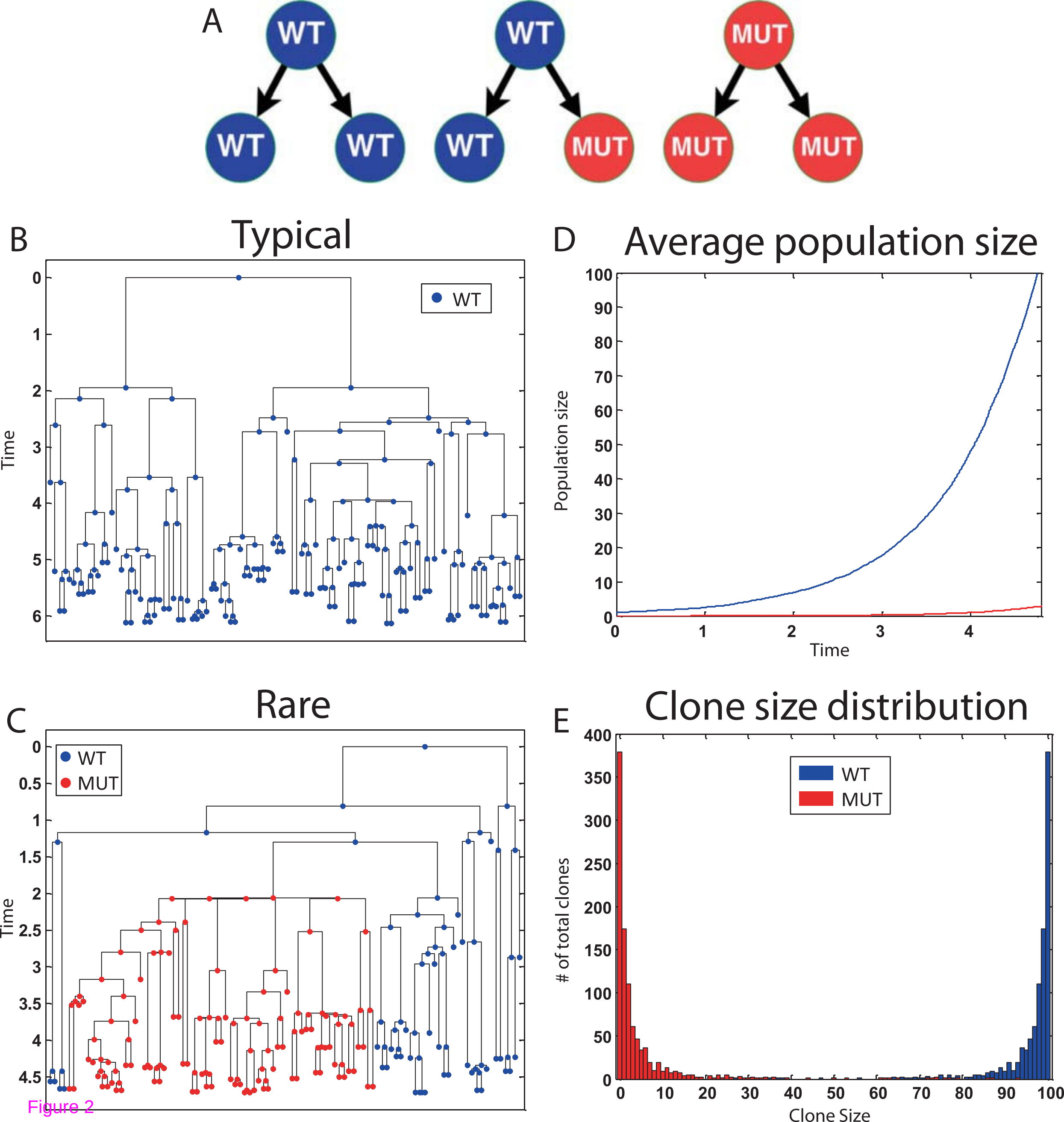
45. Sakakibara Y: **Grammatical inference in bioinformatics.** *IEEE Trans Pattern Anal Mach Intell* 2005, **27**(7)**:**1051–1062.

46. Hárs V, Tóth J: **On the inverse problem of reaction kinetics**. In *Colloquia Mathematica Societatis János Bolyai,(Szeged, Hungary, 1979) Qualitative Theory of Differential Equations.* Edited by Farkas M. 1981:363–379.



A
SC
SC
SC
SC
Diff
Diff
Diff
Diff
Diff
Diff
X
B
Example execution
SC
Diff
Population size
Time
D
Overall statistics
SC
Diff
Population size
Time
C
SC
Diff
Dead
Time
E
SC
% of total clones
Clone Size

Figure 1


A
WT
WT
WT
WT
WT
MUT
MUT
MUT
MUT
B
Typical
WT
Time
C
Rare
WT
MUT
Time
D
Average population size
Population size
Time
E
Clone size distribution
WT
MUT
# of total clones
Clone Size
Figure 2

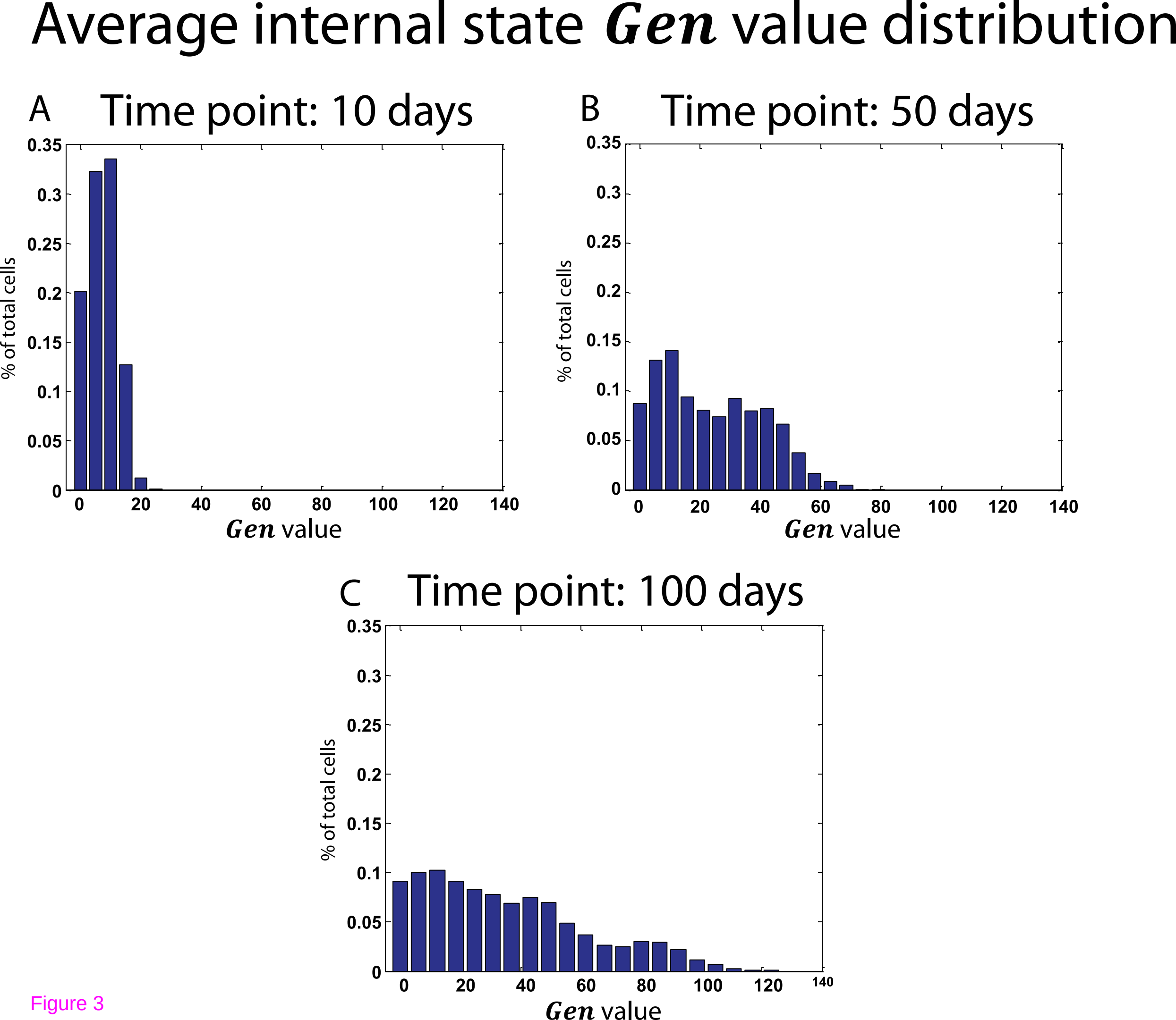
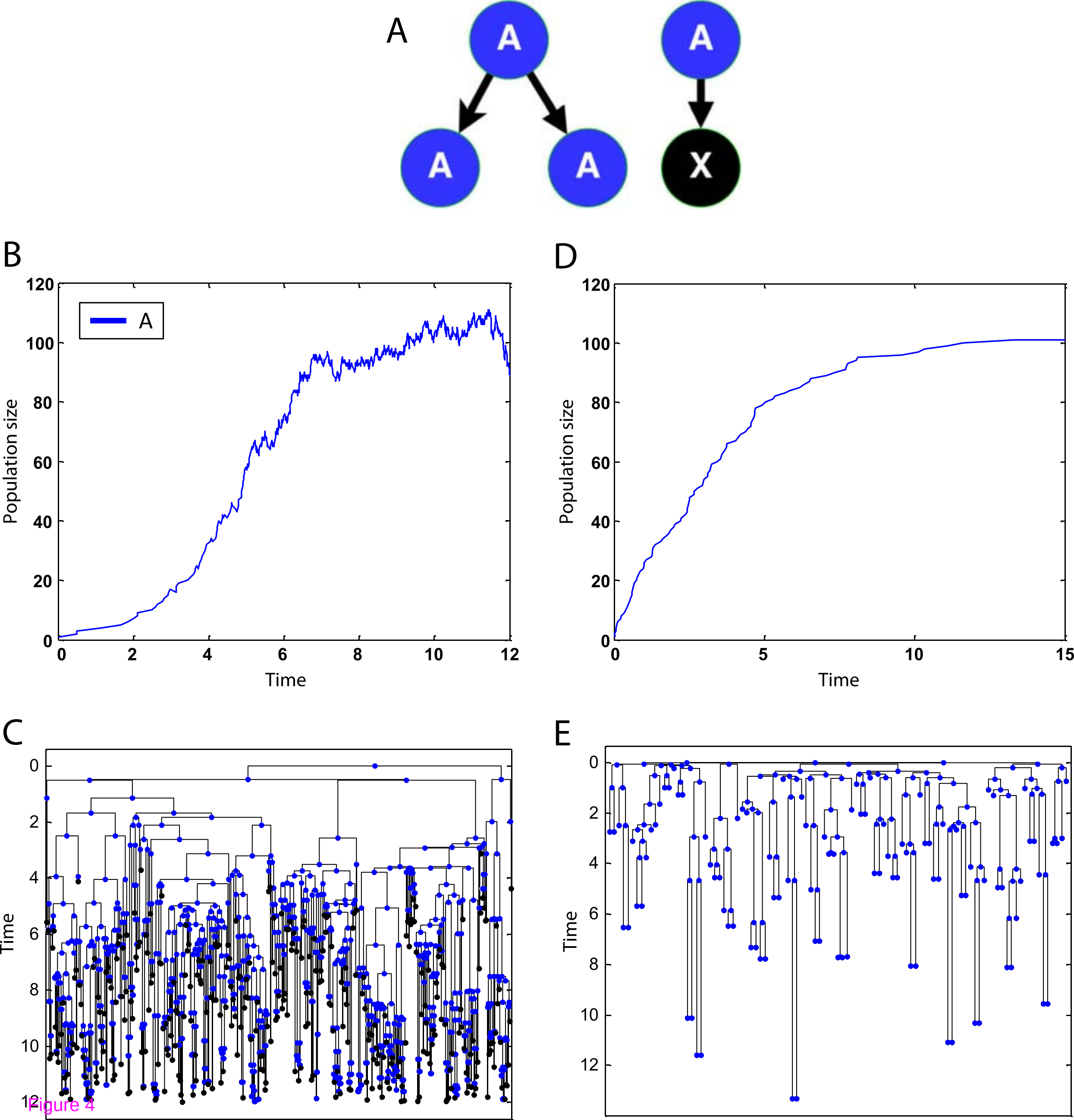
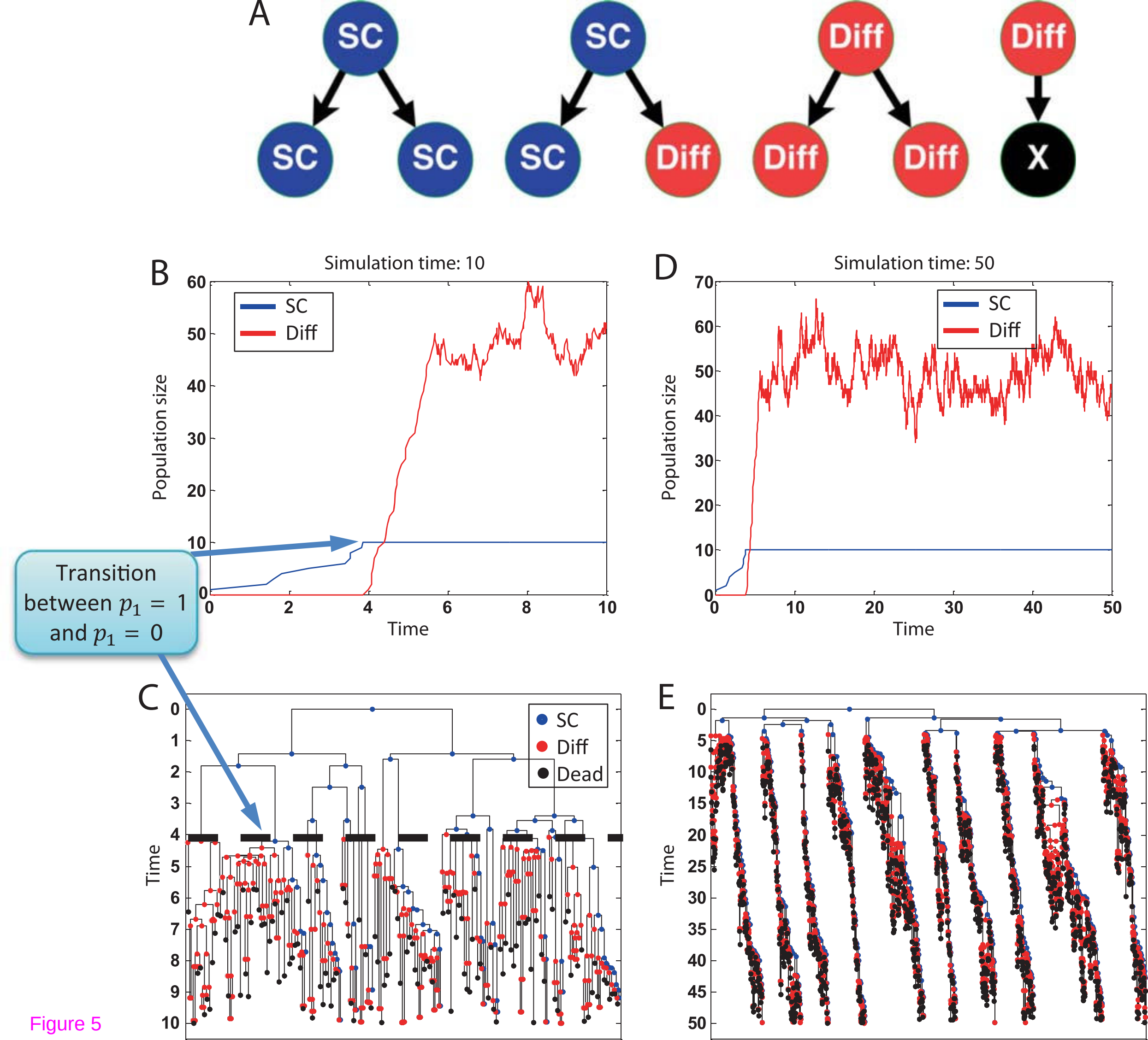

Average internal state Gen value distribution
A
Time point: 10 days
B
Time point: 50 days
C
Time point: 100 days
% of total cells
Gen value
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
20
40
60
80
100
120
140

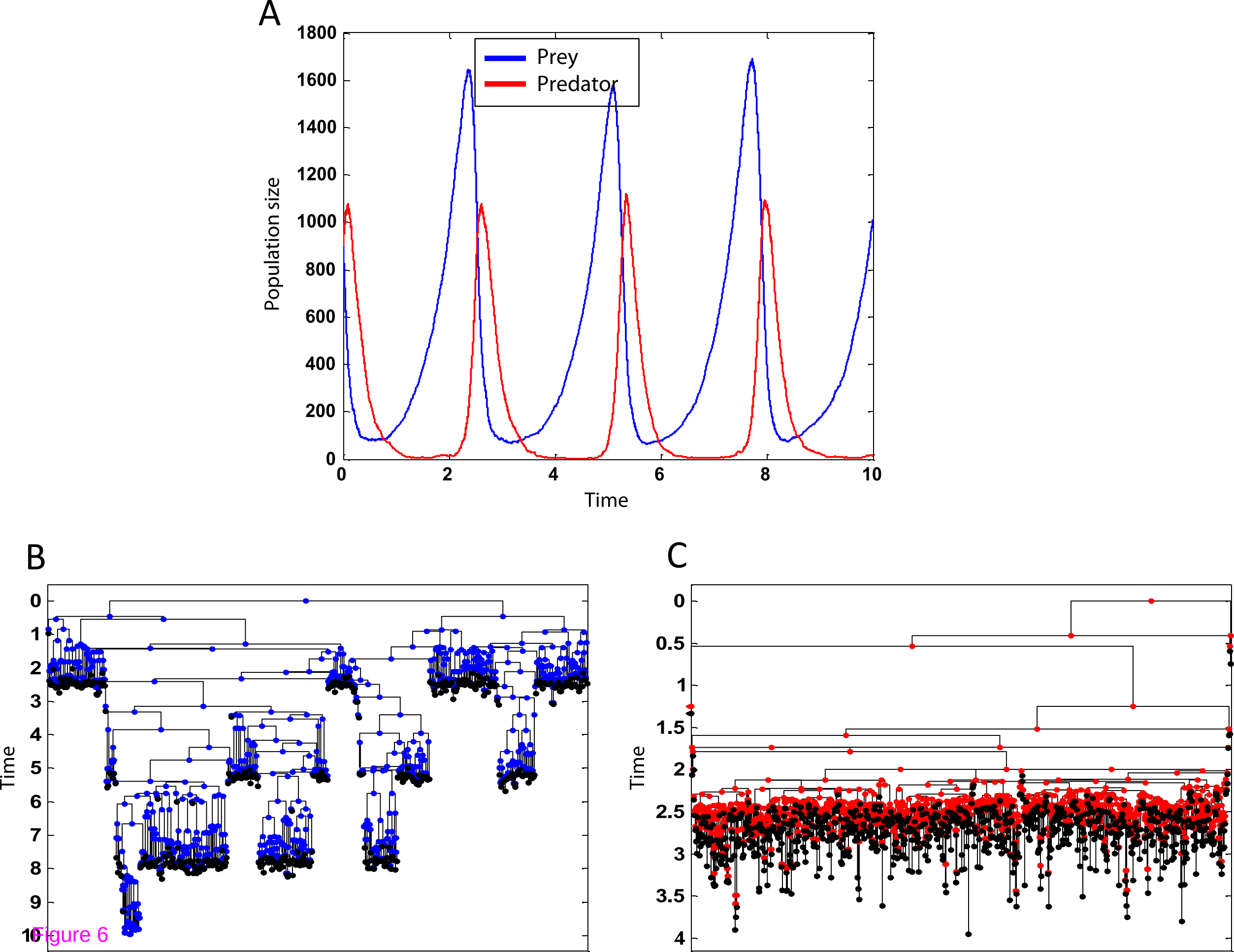
Figure 3


A
A
A
A
A
X
B
D
C
E
A
Population size
Time
Population size
Time
Time
Time
Figure 4



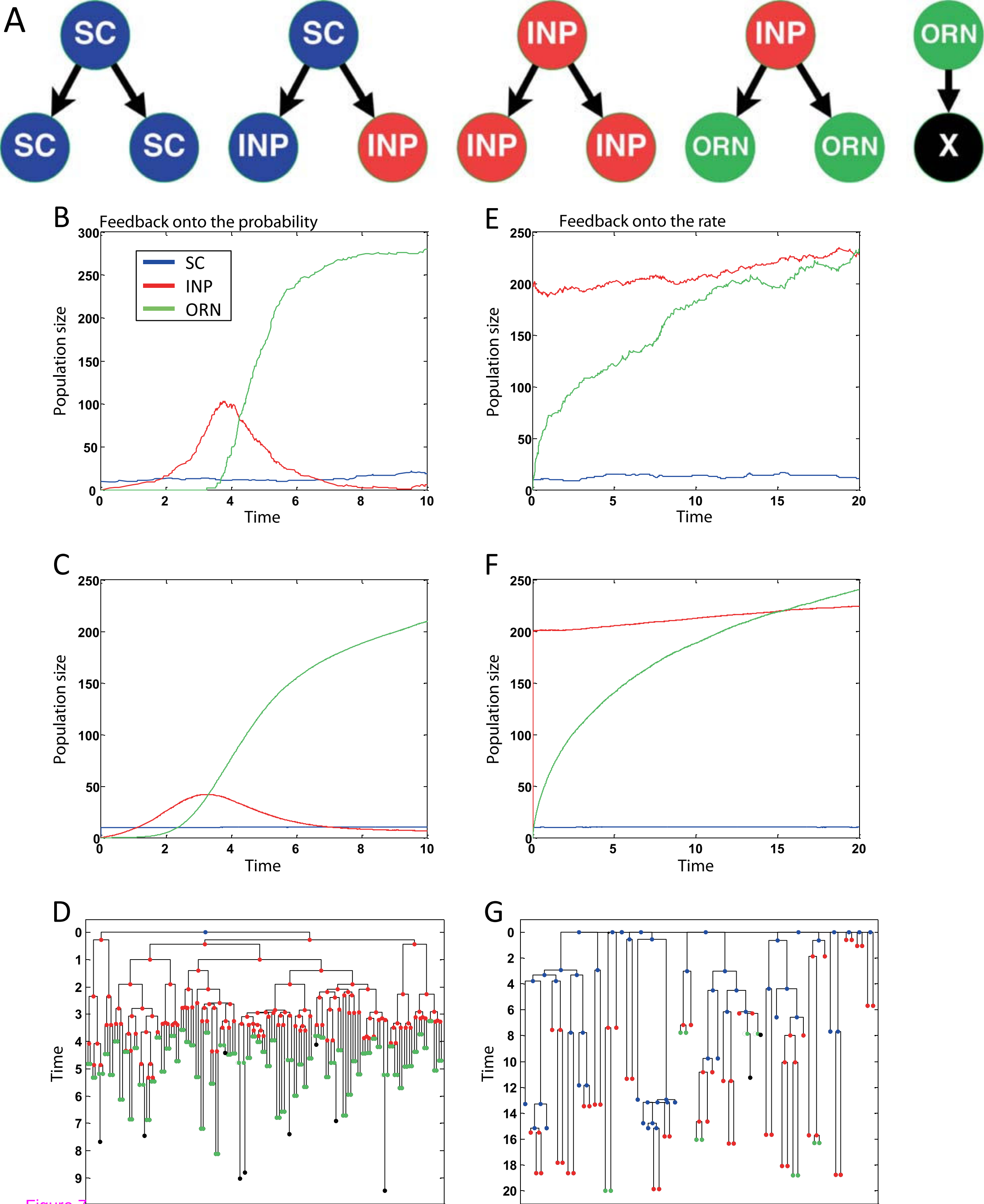
A
SC
SC
SC
SC
SC
Diff
Diff
Diff
Diff
Diff
X
B
Simulation time: 10
Population size
Time
SC
Diff
D
Simulation time: 50
Population size
Time
SC
Diff
Transition between $p_1 = 1$ and $p_1 = 0$
C
Time
SC
Diff
Dead
E
Time

Figure 5



A
Prey
Predator
Population size
Time
0
200
400
600
800
1000
1200
1400
1600
1800
0
2
4
6
8
10
B
Time
0
1
2
3
4
5
6
7
8
9
10
C
Time
0
0.5
1
1.5
2
2.5
3
3.5
4

Figure 6


A
SC
SC
SC
SC
INP
INP
INP
INP
INP
INP
ORN
ORN
ORN
X
B
Feedback onto the probability
SC
INP
ORN
Population size
Time
C
Population size
Time
D
Time
E
Feedback onto the rate
Population size
Time
F
Population size
Time
G
Time
Figure 7