Available online at www.sciencedirect.com

SciVerse ScienceDirect

Electronic Notes in Theoretical Computer Science 293 (2013) 17–34

Electronic Notes in Theoretical Computer Science

www.elsevier.com/locate/entcs

Simulating Signalling Pathways With BioWayS

Davide Chiarugi [1] Moreno Falaschi[2] Diana Hermith[3]

Dipartimento di Ingegneria dell'Informazione e Scienze Matematiche
Università degli Studi di Siena. Italy.

Michell Guzman[4]

Escuela de Ingeniería de Sistemas y Computación.
Universidad del Valle Cali. Colombia.

Carlos Olarte[5]

Departamento de Electrónica y Ciencias de la Computación.
Universidad Javeriana Cali. Colombia.

Abstract

We report on a technique for modelling biological systems based on the ntcc calculus, a model of concurrency where systems are specified by means of constraints (i.e., formulae in logic). We show that the ability of ntcc to express partial information, concurrency, non-determinism and timed behaviour makes it well-suited model and simulate biochemical reactions networks. Based on this technique, we introduce BioWayS (BIOchemical pathWAY Simulator), a software tool for the quantitative modelling and analysis of biological systems. We show the applicability of BioWayS in the context of two well-studied biological systems: the glycogen breakdown pathway and the life cycle of the human immunodeficiency virus.

Keywords: Signaling pathways, Concurrent Constraint Programming, biological systems.

1 Introduction

Computational biology aims at using methods and techniques from computer science to integrate the existing knowledge concerning individual genes, proteins and

[1] Email: chiarugi3@unisi.it

[2] Email: moreno.falaschi@unisi.it

[3] Email: diana.hermith@unisi.it

[4] Email: michell.guzman@correounivalle.edu.co

[5] Email: carlosolarte@javerianacali.edu.co

doi:10.1016/j.entcs.2013.02.016

molecules and to investigate the behaviour and relationships among the various elements composing a biological system. A technique widely used in computational biology consists in the construction of executable models (EMs) [15] describing the studied systems as computer programs. EMs are typically specified through formal languages based either on process algebras (e.g. [10,12,11,25,3]) or other formalisms such as logic [9] or rewriting logic [14], constraint programming [21,13] or Petri nets [6]. These models can be used for analyzing either static, qualitative properties of biological systems [7] or their quantitative, dynamical behaviour. In the latter case, for taking into account dynamical aspects, the evolution of the model is driven by algorithms that, given the state of the system at one initial time t_0, allows to compute the state of the system at a subsequent time t. Depending on the chosen algorithm the model results to be stochastic or deterministic.

Recently there has been a significant interest in (executable) discrete stochastic (DS) models of biological systems, mainly because experimental data are providing evidences that stochasticity arising at the molecular level plays an important role in determining the overall behaviour of living organisms [30]. In DS models the evolution of the system is driven by a stochastic algorithm which computes the probability of state transitions according to given probability density functions (PDFs). Biochemical reactions are often modelled through DS approaches, typically by describing the reaction system in hand as a discrete-state continuous-time Markov process (DCMP) (see e.g. [4]). Gillespie's Stochastic Simulation Algorithm (SSA) [17], based on previous proposals (e.g. [4]), is the most widespread algorithm used for implementing DS simulations of biological systems. Gillespie's SSA requires that some hypotheses are satisfied, namely solutions are well stirred and in thermal equilibrium and, more importantly, it holds only for elementary chemical reactions i.e., those reactions occurring in one reactive event. Even though it has been shown that the SSA can work besides the prescribed scope of applicability as proved by the success of various stochastic models against experimental data, it is difficult to describe biochemical systems in terms of elementary reactions: often there is an incomplete knowledge of the full set of elementary reactions and mesoscopic or macroscopic transformations are the only observable ones. Most commonly this problem is circumvented abstracting away the unobservable elementary steps, lumping them in a single reaction event modelled as a single "Markov jump" with the waiting time τ sampled from a negative exponential distribution depending on an overall rate constant. However, abstractions usually introduce approximations in the behaviour of the models. The impact of these approximations is not easy to evaluate or estimate, as noticed by Gillespie in [31] for enzymatically catalyzed reactions. One crucial point in this abstraction approach concerns the modelling of the waiting time, i.e. the time needed for a reaction to occur: even though the elementary reactions underlying a given biochemical process can be modelled as a DCMP (and, thus, with waiting times distributed according to a negative exponential PDF) on a mesoscopic or macroscopic scale the process may exhibit different dynamics such as non-Markovian behaviours, as pointed out also in [26] and [10] and shown by various experimental evidence, e.g. [24]. These arguments suggest the need of proposing modelling approaches embedding a more general notion of transition time allowing to describe the observed time courses of biological phenomena without assuming a memoryless process. Various approaches have been proposed for addressing this issue. In [6] an extension for Petri Nets is proposed, while in [26] the Beta Workbench toolkit is enhanced allowing the sampling of transition times from non-exponential PDFs. BioPEPAd [10] allows to add deterministic delays to the duration of a reaction. The work mentioned above propose different approaches in modelling reaction's waiting times t. In [26] and [6], similarly to the Gillespie's SSA, t represents the time needed for the event *reaction* to be *enabled*, i.e. the time needed for the reactants to collide effectively and react. In these cases, the *effects* of the reactions (i.e. the formation of the products) are considered instantaneous. Conversely, in [10] the enabling and the completion of the reactions are rendered as two separated events and products formation is not instantaneous. As it will be detailed in Sections 2 and 3 we consider t as the interval between two reactive events, i.e. the time separating two subsequent events of product formation. In this way we abstract the difference between the enabling and the effects of a reaction without subsuming whether product formation is instantaneous or not. This choice allow us to model more faithfully experimental evidences. Indeed in (bio)chemistry and biology the kinetics of reactions are measured observing product formation during time, i.e. "counting" the amount of molecules produced in given time intervals. Usually these experiments do not provide information about which part of the time needed for a reaction to occur is spent for the enabling or completion phase. Thus, focusing on a more abstract notion of waiting time, we aim at preventing the insertion of artifacts in modelling reaction dynamics.

In this paper we propose an approach based on the `ntcc` calculus [28], a temporal extension of Concurrent Constraint Programming (CCP) [32], designed for specifying and verifying timed and reactive systems. In particular we report on BioWayS (BIOchemical pathWAY Simulator), a PHP based application designed for specifying and executing `ntcc` models of biological systems. Other authors (see e.g. [8]) used (extensions of) CCP for modelling biological systems. Anyway, up to our knowledge, only a few of them (see e.g., [2,20]) focus on timed extensions of CCP such as the `ntcc` calculus. In [2] the methods for describing living systems through `ntcc` are presented at the theoretical level. In [20] the `ntcc` approach is applied for specifying biological phenomena. This work integrates these proposals in three directions: (1) providing a systematic discussion of the features of the `ntcc`-based modelling approach for biology; (2) presenting a `ntcc`-based working software; (3) modelling and analyzing two complex biological scenarios, namely, the glucagon-induced glycogen breakdown and the life cycle of the human immunodeficiency virus (HIV).

As we shall show, `ntcc` offers several advantages in the modelling of biochemical reaction systems: (1) the timed nature of the calculus allow us to faithfully model *temporal information about interactions*, *information about the temporal occurrence of an event* (e.g., when a binding occurs), and *information about the relative velocities of reactions* (e.g., the duration of an interaction) thus allowing to take into account non-markovian dynamics. (2) Constraints in `ntcc` provide a compact representation of the state of the system, (e.g., the concentration of the components along the time). (3) `ntcc` models can be seen as executable: `ntcc` processes can be straightforwardly executed and the evolution of the system can be observed. Finally, (4) the `ntcc` calculus is equipped with an underlying temporal logic that allows to formally specify and verify properties of the model. Notice that, with respect to the process algebra-based approaches, CCP allows an explicit notion of states based on constraints as well as an explicit notion of time. Summing up, the contribution of this paper is twofold: on the one hand we present both a systematic discussion of the main features that makes `ntcc` suitable for modelling biological systems and; the software tool designed for simulating biological phenomena enjoying the features of the `ntcc` based approach. On the other hand we apply our technique for modelling and analyzing two complex biological systems.

The rest of the paper is structured as follows: In Section 2 we describe the `ntcc` calculus. In Section 3 we present our software tool (BioWayS) on two examples (the glycogen breakdown pathway and the HIV life cycle). Section 4 concludes the paper.

2 Timed Concurrent Constraint Programming

Process calculi such as CCS and the π-calculus among several others have arisen as mathematical formalisms to model and reason about concurrent systems. They treat concurrent processes much like the λ-calculus treats computable functions. They then provide a language in which the structure of terms represents the structure of processes together with an operational semantics to represent computational steps.

In this paper we shall use as modelling language Concurrent Constraint Programming (CCP) [32], a model for concurrency that combines the traditional operational view of process calculi with a declarative view based on logic. This allows CCP to benefit from the large set of reasoning techniques of both process calculi and logic [32,28].

Agents in CCP *interact* with each other by *telling* and *asking* information represented as *constraints* to a global store. Constraints (e.g., $x > 42$) can be thought of as formulae in a first-order language and they represent (partial) information about the variables of the system. Partial must be understood here as the fact that constraints do not necessarily determine completely the values of the variables.

The basic constructs in CCP are the *tell* agent $\mathbf{tell}(c)$ that adds the constraint c (via logical conjunction) to the store, thus making it available to the other processes; and the *ask* process $\mathbf{when}\ c\ \mathbf{do}\ P$ that queries if the current store d can entail (deduce) the guard c, written $d \models c$; if so, it behaves like P. Otherwise it remains blocked until more information is added. Hence, ask processes define a synchronisation mechanism based on entailment of constraints. CCP features also constructs for declaring local variables as in $(\mathbf{local}\, x)\, P$ and for executing processes in parallel as in $P \parallel Q$.

The `ntcc` calculus [28] extends CCP with the notion of discrete time-units to model timed and reactive systems. Roughly speaking, a CCP-like process is executed in a time-unit. When the resting point is reached, i.e., no further evolution is possible, the store is output and a new time-unit is created to later execute the continuation of the process. In order to specify this kind of behaviour, `ntcc` extends CCP with operators such as $\textbf{next}\, P$ that delays the execution of P one time-unit; the replication $!\,P$ that executes P in all the time-units; and $\textbf{unless}\ c\ \textbf{next}\, P$ that executes P in the next time-unit if c cannot be deduced from the store. Furthermore, `ntcc` introduces non-deterministic choices of the form $\sum_{i\in I} \textbf{when}\ c_i\ \textbf{do}\ P_i$ where one P_i is chosen for execution if the guard c_i can be entailed from the store. When this happens, the other alternatives are precluded from execution. Asynchronous behaviour is introduced by adding to the syntax a finite delay operator: the process $\star P$ represents a finite but unbounded delay for the activation of P.

The notion of constraint and the language of processes in `ntcc` are expressive enough to specify the biological behaviour we are interested in modelling:

- Quantitative information can be naturally expressed by means of constraints. For instance $x > y$ states that the concentration of x is greater than that of y.
- Constraints provide also an elegant mechanism to represent partial information. For instance, $x > 42$ gives some information regarding the concentration of x but it does not give a specific value for it. This can be helpful when some components of the system are not well known or we do not have enough quantitative information about them.
- Synchronisation of ask processes via constraint entailment allow us to trigger actions when some information can be derived from the system. For instance, it is natural to express in the language that a given reaction occurs only when certain component is present in the system.
- The ability of CCP to compose models (i.e., components) by parallel composition leads to a robust modelling strategy: we can study separately components of a system and then, observe the behaviour of the whole system.
- Timed operators as $\textbf{next}\, P$ allow us to describe reaction (or interactions) that can take several time-units to be completed.
- Furthermore, since `ntcc` is a model of concurrency, we can use several techniques to reason about the models we build. For instance, operational and denotational semantics, model checking techniques and logical interpretation of processes (see e.g., [32,28]).

2.1 BioWayS: a `ntcc` model of biochemical reactions

The tool we propose here models biological systems by means of a set of reaction rules of the form

$$a_1X_1 + ...a_nX_n \dashrightarrow b_1Y_1... + b_mY_m \tag{1}$$

The constants $a_1, ..., a_n$ and $b_1, ..., b_m$ are the stoichiometric coefficients. Therefore, $a_1X_1, a_2X_2, ..., a_nX_n$ are reactants that interact (and then consumed) yielding to the products $b_1Y_1, b_2Y_2, ..., b_mY_m$.

In order to represent the reaction above, we model in `ntcc` each type of molecule as a variable (e.g., X_i) and Equation (1) as the process

$$\begin{aligned}\texttt{eq-proc} = {} & \textbf{when } X_1 \geq a_1 \wedge ... \wedge X_n \geq a_n \textbf{ do} \\ & \quad \textbf{next}^{(t)} \ (\textbf{tell}(Y_1 = Y_1' + b_1 \wedge ... \wedge Y_m = Y_m' + b_m)) \parallel \\ & \quad \textbf{next}\,\textbf{tell}(X_1 = X_1' - a_1 \wedge ... \wedge X_n = X_n' - a_n)\end{aligned} \tag{2}$$

We use primed variables to represent the value of the variables in the previous time unit as in $X_1 = X_1'$. The process above can be read as “when the reactants are available, they are consumed and the right-hand components are produced t time units later”. Hence, the kinetic parameter t allow us to represent the speed of reactions.

Assume now a set of n stoichiometric equations. We need a process that chooses one of the reaction to occur at a given time-unit. This can be done by composing each process $\texttt{eq-proc}_i$ in a non-deterministic choice of the form:

$$\sum_{i \in 1..n} \texttt{eq-proc}_i$$

Furthermore, if the propensity of each reaction to occur is available, it is possible to consider probabilistic extensions of CCP as the one studied in [19]. Then, instead of performing a non-deterministic choice, we can consider a probabilistic process of the form:

$$\texttt{new}(p, F) \ \texttt{in} \sum_{i \in 1..n} \textbf{when } p = i \textbf{ do } \texttt{eq-proc}_i$$

where the value of p is chosen according to the probability mass function F for $\{1, \ldots, n\}$. For instance, for a system of two reactions, the process

$$\texttt{new}(p, F : F(1) = 0.2, F(2) = 0.8) \ \texttt{in} \sum_{i \in 1..2} \textbf{when } p = i \textbf{ do } \texttt{eq-proc}_i$$

will choose with a probability of 0.2 the first reaction and with a probability of 0.8 the second reaction.

Summing up, in BioWayS it is possible to express two important features of biochemical reactions: the propensity (i.e. the probability of occurring) and the duration, i.e. the time steps needed for the products to appear in the system. This is an abstraction to represent the speed of a chemical reaction (i.e. the reaction rate constant).

The model as a runnable specification

Processes in `ntcc` can be seen as runnable specifications of a system: the model can be directly simulated by using the operational semantics (SOS) of the calculus.

The SOS dictates how processes evolve along time units. For instance, a process **tell**(c) evolves into **skip** (the inactive process) by adding c to the current store d:

$$\text{R}_{\text{TELL}}\ \frac{}{\langle \mathbf{tell}(c), d\rangle \longrightarrow \langle \mathbf{skip}, d \wedge c\rangle}$$

Similarly, for the rest of the processes. The reader may refer [28] for a complete description of SOS rules of `ntcc`.

Following the operational rules of the calculus, we built an interpreter of `ntcc` on top of the Mozart programming language (`http://www.mozart-oz.org/`). Central to this implementation is the Mozart abstraction of a *computation space* (CS). A CS is a constraint store where multiple threads can access (concurrently) the shared variables and impose constraints on them. In Mozart, different constraint systems are available. Here we used the Finite Domain Constraint System (FD). In FD variables are assumed to range over finite domains and, in addition to equality, we may have predicates that restrict the possible values of a variable to some finite set as in $x > y$. We thus model `ntcc` processes as threads that post and query constraint in the CS until a resting point is reached. When this happens, we output the final store that contains the information about the variables of the model. This gives a simple way to "execute" the `ntcc` model and observe the behaviour of the modelled system in each time-unit.

In order to make the tool available on Internet, we embedded the `ntcc` interpreter into BioWayS, a PHP based application freely available at `http://avispa.puj.edu.co`. Users can build models of biological systems in BioWayS through a wizard that comprises the following steps:

(i) Variables: Define the reacting species and their initial concentration.

(ii) System's reactions: Define the type of reaction that describes how molecules interact.

(iii) Propensity of reactions: Define the probability of each reaction to occur.

(iv) Duration of reactions: Define the duration of each type of interaction.

(v) Number of time-units: Time-window for the simulation.

3 Modelling

In this section we outline the use of our toolkit through the specification and simulation of two well-studied biological scenarios: (*i*) the signalling pathway leading to the *glycogen* breakdown into *glucose 1-phosphate*, and (*ii*) the life cycle of the HIV. During the discussion we also highlight the features of our approach in these specific cases.

3.1 The glycogen breakdown pathway: A model of intracellular processes

In higher organisms such as mammals glycogen is stored in the liver as a reservoir of glucose. When the concentration of glucose in the blood is low the α cells of the pancreas secrete glucagon, a polypeptidic hormone which triggers the process of glycogen breakdown (glycogenolysis). This process is started by the interaction of glucagon with its receptor expressed by liver cells [1]. Once the glucagon receptor embedded in the cell membrane binds its ligand, it activates a signal transduction pathway inside the cell leading to a glycogenolysis. More precisely, the signal transduction system for the glycogen degradation pathway is modular and is made of three type of proteins: (*i*) a receptor, (*ii*) a transducer, and (*iii*) an effector. Glucagon recognises and binds to its receptor causing an allosteric change [16]. Responding to this structural modification, the transducer (a G-protein *–guanine nucleotide-binding protein–* located in the inner side of the cell membrane) interacts with the hormone-receptor complex stimulating a reaction in which a *GDP* (*guanosine diphosphate*) molecule bound to the G-protein is replaced by *GTP* (*guanosine-5′-triphosphate*). This reaction activates the G-protein, which then interacts with the effector, the enzyme adenylyl cyclase. This protein catalyses the conversion of *ATP* (*adenosine-5′-triphosphate*) to *cAMP* (*cyclic adenosine monophosphate*), an intracellular second messenger. Thus, the binding of glucagon at the cell-surface stimulates the synthesis of a second messenger inside the cell, which in turn stimulates a metabolic response (see Figure 1). The first interaction triggers a cascade of biochemical reactions in a signal transduction pathway through the activation of G-proteins [1].

In order to model the system described above we apply a compositional approach by adding iteratively biochemical interactions. This is particularly straightforward in our framework: new information can be added to the system by posting constraints and the subsystems can be easily composed by sharing variables. Compositionality allows to build complex biological models combining partial information coming from different sources. This result is certainly more difficult to achieve, for example, in models based on ordinary differential equations (ODE) because the large number of parameters needed and, in general, ODE are not compositional. Moreover, the idea of partial information represented as constraints makes CCP appropriate for this aim.

Note that our technique allows the description of the biological scenario at different levels of abstraction. For instance, in [21] we considered the interactions between the transmembrane receptor and G-proteins in three different environments (extracellular, transmembrane, and intracellular). Now, we expand this perspective by zooming into the intracellular domain with the aim to analyse the system's behaviour at this level. This should allow us to gain a better understanding of the system dynamics in response to the presence or absence of signalling molecules.

We describe the glycogen breakdown pathway through a set of reaction rules resembling [13,21] and we considered the actions of binding, dissociation, complex formation, and transfer of molecule groups. The full specification of the model is reported in Appendix A. For each reaction we specified the initial concentration c of reactants , the probability p to occur and the time t (see parameter t in Equation (2)) between two subsequent events of product formation. These parameters were estimated accordingly to literature data [5,22,23,27,34]. In particular t is estimated

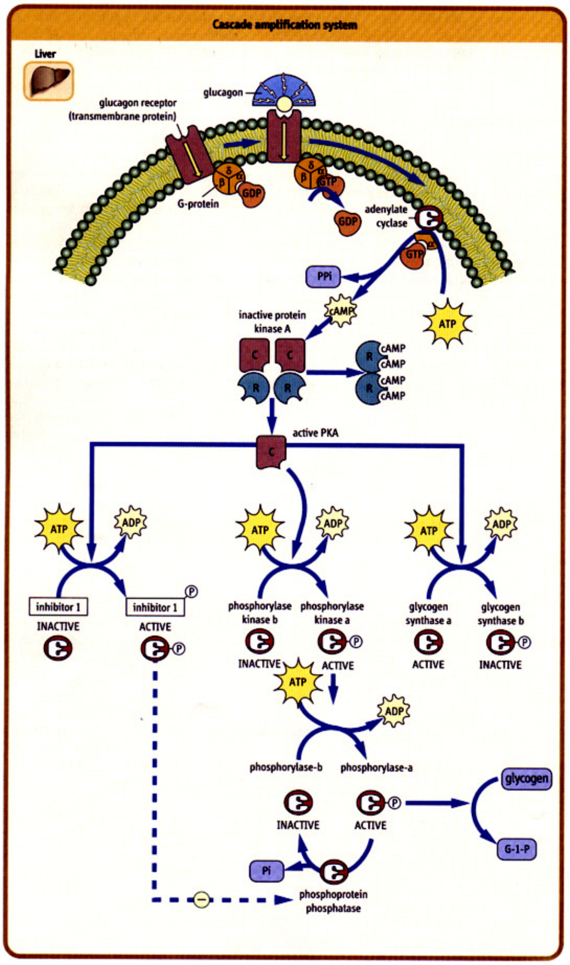


Fig. 1. A reaction pathway for the glycogen breakdown (taken from [5] and https://files.nyu.edu/gcl1/public/).

from reaction rates: the highest the reaction rate, the shortest t. We used c to set the experimental conditions. Indeed we simulated different scenarios (called *modes* in the Appendix) in which c can be *low*, *medium* or *high*. For the simulations performed here, we assumed that all reactions have the same propensity to occur. Thanks to the timed nature of our modelling language, we are able to emphasise the interactions between different species during time.

Our in silico experiments show that the behaviour of our model is consistent with that of the real counterpart. In particular, in liver cells, in response to the hormone *glucagon*, the transmembrane protein *adenylyl cyclase* is stimulated and catalyses the conversion of ATP into $cAMP$, an intracellular second messenger (see Tables A.1 and A.2). This step of the signal transduction pathway is called amplification of the signal, since in response to a small amount of glucagon, a large quantity of $cAMP$ is produced (see Figure 2). The amplification process is mediated by subsequent molecular interactions. Indeed, before glucagon unbinds from its

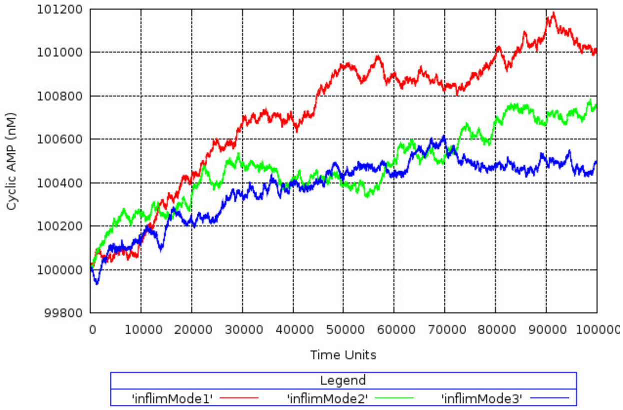
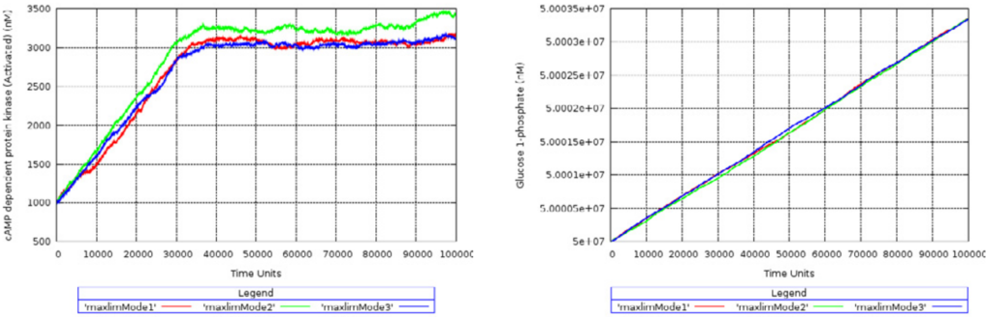


Fig. 2. Temporal trace of $cAMP$ involved in the glycogen breakdown cascade. It can be observed an increased formation of $cAMP$ in presence of lower values of concentrations and rate constants, particularly in terms of the extracellular *ligand* (see details in Appendix A.)



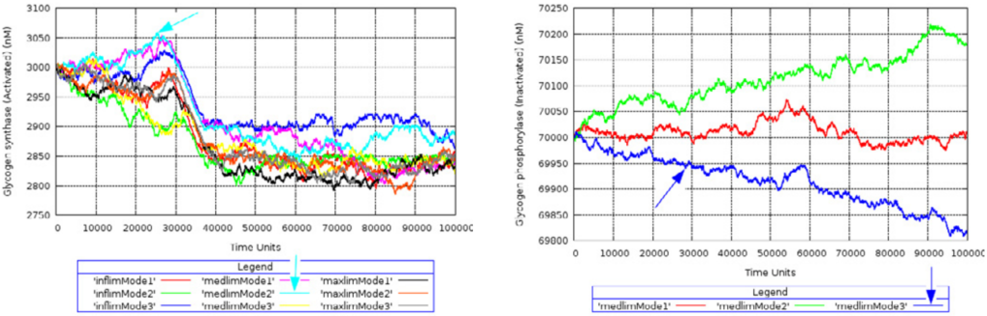
(a) (b)

Fig. 3. 3a Temporal trace for enzyme *protein kinase A* ($cAMP$ *dependent protein kinase* ($cAPK$)) and 3b *glucose 1-phosphate*.

receptor, several G-proteins can be activated. The activated proteins migrate along the inner side of the plasma membrane and may enhance, in turn, the catalytic capabilities of many adenylyl cyclases. Moreover, during the life-time of the G-protein-adenylyl cyclase complex, several ATP molecules are converted into $cAMP$, thus leading to further amplification.

Throughout the signal cascade, $cAMP$ is capable to activate the enzyme *protein kinase A* ($cAMP$ *dependent protein kinase* ($cAPK$)). Thus, high levels of $cAPK$ are observed in the system (see Figure 3a). This environment is a requirement for the activation of others intracellular signalling molecules (see Equations EQ8-EQ12.1 in Appendix A) and to promote the degradation of *glycogen* into molecules of *glucose* 1-*phosphate* (see Figure 3b). Therefore, the processing of information from the external environment to the intracellular medium, starts at the level of the cell membrane through the binding of the hormone *glucagon* to its respective receptor in a “physical interaction” followed by a set of biochemical reactions (transduction pathway) in which the initial stimulus is greatly amplified.

Our representation of the interactions is in agreement with [33]: the following enzymes must be both present and available for activation: *glycogen phosphorylase*



(a) (b)

Fig. 4. 4a Temporal trace for proteins *glycogen synthase* (active form) and 4b *glycogen phosphorylase* (inactive form).

kinase and *glycogen phosphorylase*, as well as *protein kinase A* must be available for activation by *cAMP*. The enzyme *adenylyl cyclase* must be present according with [29]. In line with [33,29] in almost all species the activity of *glycogen synthase* increases rapidly while *glycogen phosphorylase* remains low, and the ratio of the active forms of *glycogen phosphorylase* and *glycogen synthase* might be of major importance in the regulation of metabolism. Our simulations (see Figure 4) reproduce this behaviour: enzyme *glycogen synthase* (active) increases in an overshoot and afterwards decreases and remains in a constant level especially when we consider the configuration “medlimMode2” (see the blue arrow in Figure 4a) to perform the simulations (see Tables A.1 and A.2). In such configurations the mid values of concentrations and rate constants of the molecules and reactions in the network are taken. When we consider the configuration “medlimMode3” (see Tables A.1 and A.2), the *glycogen phosphorylase* (inactive) decreases to lower levels of concentration (see the blue arrow in Figure 4b). Moreover, we can notice that the ratio and behaviour of the inactive/active forms of *glycogen phosphorylase* and *glycogen synthase* affects not only the degradation of *glycogen* to *glucose* 1-*phosphate* (the active form of the *glycogen phosphorylase* degrades *glycogen* –see Equations EQ11-11.1 in Appendix A) but also the *glycogen* synthesis (the active form of *glycogen synthase* catalyses the formation of *glycogen* polymers –see Equations EQ12-12.1 in Appendix A). The ability of our modeling strategy and tool to easily adjust parameters and compose subsystems was crucial here to identify the above mentioned behaviours and also to gain insights on the regulation of glycogenolysis.

3.2 The HIV life cycle: A representation based on biochemical interactions

Since the discovery of the human immunodeficiency virus (HIV), the etiologic agent of the acquired immune deficiency syndrome (AIDS), scientists have studied the dynamics and details of the HIV life cycle in order to develop efficient antiviral therapies. In the context of computational biology the dynamics of cell-virus interactions have traditionally been investigated through both ODEs and EMs. In the latter case the proposed modelling approaches mainly focus on membrane interactions (see e.g. in [11]). We shall use here BioWayS for building a model of the

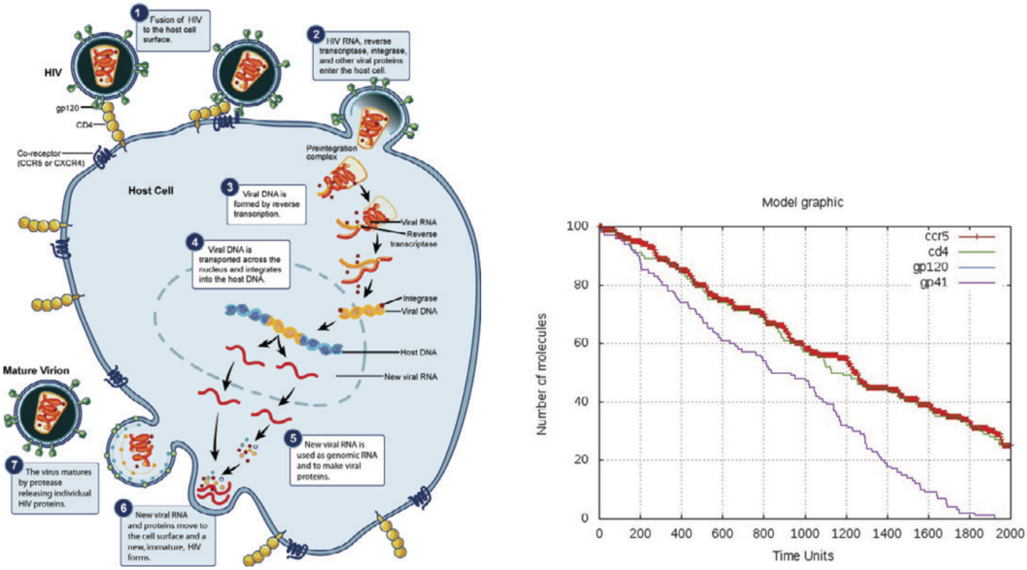
HIV infection cycle shifting the focus on quantitative issues. Our aim is to track the time course of a set of molecules during the various steps of the infection cycle. Our investigation is driven by the available observations describing the progression throughout the infection cycle as conditioned by the presence of certain molecules in each phase. In other words, each stage of the infection cycle is characterised by a precise set of molecules which are necessary for bootstrapping the following step. Our interest in modelling the timing of this process reflects one research strategy in drug discovery, aiming at blocking the infection cycle by interfering with the bootstrapping molecules. Our framework promises to be particularly suited for this purpose since time rates can be taken into account and `ntcc` processes can be progressively composed in order to build a model integrating information coming from different sources.

The various steps of the HIV infection process can be described as follows (see [18] for a detailed description and Figure 5):

- Binding and Fusion: HIV binds to a specific receptor CD4 (cluster of differentiation 4) and one of two co-receptors on the surface of a CD4+ T-lymphocyte and fuses with the host cell releasing its RNA genome.
- Reverse Transcription: reverse transcriptase converts the single-stranded HIV RNA to double-stranded HIV DNA.
- Integration: the HIV DNA enters the host cell's nucleus where it may remain inactive producing few or no new copies of HIV.
- Assembly: new virus particles are assembled in the host cell.
- Budding: the newly assembled virus pushes out ("buds") from the host cell.

We used BioWayS to describe the interactions amongst the molecules of both the virus and the host cell involved in the infection cycle. For this, we specified for each reaction its duration and the initial amount of reactants. The equations composing the model and the chosen parameters are reported in Appendix B. An interesting observation that can be abstracted from the simulation results, is the cyclical behaviour of the system based on the availability of the following molecules: ccr5 (C-C chemokine receptor type 5), cd4, gp41 (glycoprotein-membrane anchored fusion protein subunit) and gp120 (glycoprotein-receptor binding domain) (see Figure 5). These molecules set a sort of boundary condition for the reaction scheme: if they are not available to feed the system, the cycle is stopped.

BioWayS then allowed us to correctly identify a set of molecules whose presence is a necessary condition for proceeding throughout each step of the infection cycle. Consistently with available data, for instance, we found that in our model the presence of the complex $gp120/gp41$ is necessary for both the Binding and Fusion and the Budding phases. Without this complex, the virus is unable to infect the host cell. Interestingly, drugs called Fusion Inhibitors, such as Maraviroc, block the fusion phase interfering with the binding of the host-cell co-receptor $ccr5$ and the complex $gp120/gp41 : cd4$ thus avoiding the HIV life cycle to continue.



(a) (b)

Fig. 5. Steps of the replication of the virus (taken from http://www.niaid.nih.gov) (left) and the time course of $gp120/gp41 : cd4 : ccr5$ necessary to initiate membrane fusion in the HIV life cycle (right).

4 Conclusions and future work

We have defined a technique based on a temporal extension of Concurrent Constraint Programming (CCP) for modelling biological systems that allows us to represent straightforwardly transition times and partial information. In this paper we have presented also a software tool (BioWayS) for modelling and analysing biochemical interaction networks. Through two working examples we have illustrated how our method can be used for gaining insights on the dynamics of biological phenomena.

We are currently developing an extension of our toolkit allowing us to consider also stochastic waiting times distributed according to non-exponential probability distribution functions. This will lay the ground to describe a larger set of biological scenarios.

References

[1] John L. Tymoczko adn Jeremy M. Berg and Lubert Stryer. *Biochemistry.* W. H. Freeman, 5th edition, 2002.

[2] Alejandro Arbelaez, Julian Gutierrez, and Jorge A. Pérez. Timed concurrent constraint programming in systems biology. *Newsletter of the ALP*, 19(4), 2006.

[3] Roberto Barbuti, Giulio Caravagna, Andrea Maggiolo-Schettini, Paolo Milazzo, and Giovanni Pardini. The calculus of looping sequences. In Marco Bernardo, Pierpaolo Degano, and Gianluigi Zavattaro, editors, *SFM*, volume 5016 of *Lecture Notes in Computer Science*, pages 387–423. Springer, 2008.

[4] A. F. Bartholomay. Molecular set theory: A mathematical representation for chemical reaction mechanisms. *Bulletin of Mathematical Biology*, 22(3):285–307, 1960.

[5] John W. Baynes and Marek H. Dominiczak. *Medical Biochemistry.* Mosby, 2004.

[6] Andrea Bobbio and Miklós Telek. Computational restrictions for SPN with generally distributed transition times. In Klaus Echtle, Dieter K. Hammer, and David Powell, editors, *EDCC*, volume 852 of *Lecture Notes in Computer Science*, pages 131–148. Springer, 1994.

[7] Chiara Bodei, Andrea Bracciali, Davide Chiarugi, and Roberta Gori. A taxonomy of causality-based biological properties. In Emanuela Merelli and Paola Quaglia, editors, *FBTC*, volume 19 of *EPTCS*, pages 116–133, 2010.

[8] Luca Bortolussi and Alberto Policriti. Modeling biological systems in stochastic concurrent constraint programming. *Constraints*, 13(1-2):66–90, 2008.

[9] Laurence Calzone, François Fages, and Sylvain Soliman. Biocham: an environment for modeling biological systems and formalizing experimental knowledge. *Bioinformatics*, 22(14):1805–1807, 2006.

[10] Giulio Caravagna and Jane Hillston. Bio-PEPAd: A non-markovian extension of Bio-PEPA. *Theor. Comput. Sci.*, 419:26–49, 2012.

[11] Luca Cardelli. Brane calculi. In Vincent Danos and Vincent Schächter, editors, *CMSB*, volume 3082 of *Lecture Notes in Computer Science*, pages 257–278. Springer, 2004.

[12] Davide Chiarugi, Pierpaolo Degano, and Roberto Marangoni. A computational approach to the functional screening of genomes. *PLoS Computational Biology*, 3(9), 2007.

[13] Davide Chiarugi, Moreno Falaschi, Carlos Olarte, and Catuscia Palamidessi. Compositional modelling of signalling pathways in timed concurrent constraint programming. In Aidong Zhang, Mark Borodovsky, Gultekin Özsoyoglu, and Armin R. Mikler, editors, *BCB*, pages 414–417. ACM, 2010.

[14] Steven Eker, Merrill Knapp, Keith Laderoute, Patrick Lincoln, and Carolyn L. Talcott. Pathway logic: Executable models of biological networks. *Electr. Notes Theor. Comput. Sci.*, 71:144–161, 2002.

[15] Jasmin Fisher and Thomas A. Henzinger. Executable cell biology. *Nature Biotechnology*, 25(11):1239–1249, 2007.

[16] Keith Frayn. *Metabolic Regulation: A Human Perspective*. Wiley-Blackwell, 2nd edition, 2003.

[17] Daniel T. Gillespie. Stochastic simulation of chemical kinetics. *Annu Rev Phys Chem.*, 58:35–55, 2007.

[18] T. Goto, M. Nakay, and K. Ikut. The life-cycle of human immunodeficiency virus type 1. *Micron.*, 29(2-3):123–138, 1998.

[19] Vineet Gupta, Radha Jagadeesan, and Vijay A. Saraswat. Probabilistic concurrent constraint programming. In Antoni W. Mazurkiewicz and Józef Winkowski, editors, *CONCUR*, volume 1243 of *Lecture Notes in Computer Science*, pages 243–257. Springer, 1997.

[20] Julián Gutiérrez, Jorge A. Pérez, Camilo Rueda, and Frank D. Valencia. Timed concurrent constraint programming for analysing biological systems. *Electron. Notes Theor. Comput. Sci.*, 171(2), 2007.

[21] Diana Hermith, Carlos Olarte, Camilo Rueda, and Frank D. Valencia. Modeling cellular signaling systems: An abstraction-refinement approach. In Miguel P. Rocha, Juan M. Corchado Rodríguez, Florentino Fdez-Riverola, and Alfonso Valencia, editors, *PACBB*, volume 93 of *Advances in Intelligent and Soft Computing*, pages 321–328. Springer, 2011.

[22] Vladimir L. Katanaev and Matey Chornomorets. Kinetic diversity in G-protein-coupled receptor signalling. *Biochem J.*, 401(2), 2007.

[23] Yumiko Kawai and Ifeanyi J. Arinze. Activation of glycogenolysis in neonatal liver. *The Journal of Biological Chemistry*, 256(2):853–858, 1981.

[24] S. C. Kou, Binny J. Cherayil, Wei Min, Brian P. English, and X. Sunney Xie. Single-molecule michaelismenten equations. *The Journal of Physical Chemistry B*, 109(41):19068–19081, 2005.

[25] Roberto Larcher, Corrado Priami, and Alessandro Romanel. Modelling self-assembly in BlenX. *T. Comp. Sys. Biology*, 12:163–198, 2010.

[26] I. Mura, Davide Prandi, Corrado Priami, and Alessandro Romanel. Exploiting non-markovian bio-processes. *Electr. Notes Theor. Comput. Sci.*, 253(3):83–98, 2009.

[27] Vivek K. Mutalik and K.V. Venkatesh. Quantification of the glycogen cascade system: the ultrasensitive responses of liver glycogen synthase and muscle phosphorylase are due to distinctive regulatory designs. *Theoretical Biology and Medical Modelling*, 2(1):19, 2005.

[28] Mogens Nielsen, Catuscia. Palamidessi, and Frank D. Valencia. Temporal Concurrent Constraint Programming: Denotation, logic and applications. *Nordic Journal of Computing*, 9(1), 2002.

[29] E. Novak, G.I. Drummond, J. Skala, and P. Hahn. Developmental changes in cyclic AMP, protein kinase, phosphorylase kinase, and phosphorylase in liver, heart, and skeletal muscle of the rat. *Arch. Biochem. Biophys.*, 150(2):511–518, 1972.

[30] Arjun Raj, Scott A. Rifkin, Erik Andersen, and Erik van Oudenaarden. Variability in gene expression underlies incomplete penetrance. *Nature*, 463(7283):913–918, 2010.

[31] K.R. Sanft, D. Gillespie, and L.R. Petzold. Legitimacy of the stochastic Michaelis-Menten approximation. *IET Systems Biology*, 5(1):58–69, 2011.

[32] Vijay A. Saraswat. *Concurrent Constraint Programming.* MIT Press, 1993.

[33] A. L. Schwartz and T. W. Rall. Hormonal regulation of glycogen metabolism in neonatal rat liver. *Biochem J.*, 134(4):85–993, 1973.

[34] M. Vardanega-Peicher, R. Galletto, and S. Pagliarini. Comparative effect of glucagon and isoproterenol on hepatic glycogenolysis and glycolysis in isolated perfused liver. *Brazilian Archives of Biology and Technology*, 46:563–568, 2003.

A Glycogen breakdown pathway

Below we list the reactions we considered for the simulation of the signaling pathway of glycogen breakdown. In Table A.1 we describe the experimental conditions, i.e., the rate constants for each reaction. Table A.2 describes the component of the system along with the molar concentration we considered in each simulation.

$EQ1$ $R_{glcgn}[G_\alpha GDP] + L_{glcgn} \rightleftharpoons R_{glcgn}[G_\alpha GTP]L_{glcgn}$

$EQ2$ $R_{glcgn}[G_\alpha GTP]L_{glcgn} \rightleftharpoons G_\alpha GTP + R_{glcgn}L_{glcgn}$

$EQ3$ $G_\alpha GTP + GAP \rightleftharpoons GAP \longrightarrow G_\alpha GDP + GAP + P_i$

$EQ4$ $G_\alpha GDP + R_{glcgn} \rightleftharpoons R_{glcgn}[G_\alpha GDP]$

$EQ5$ $R_{glcgn}L_{glcgn} \rightleftharpoons R_{glcgn} + L_{glcgn}$

$EQ6$ $G_\alpha GTP + AC \rightleftharpoons AC[G_\alpha GTP]$

$EQ7$ $AC[G_\alpha GTP] + ATP \rightleftharpoons [AC[G_\alpha GTP]]ATP \longrightarrow cAMP + 2P_i + AC[G_\alpha GTP]$

$EQ8$ $4cAMP + 2cAPK_{inact} \rightleftharpoons 2cAPK_{act}$

$EQ9$ $cAPK_{act} + GPK_{inact} \rightleftharpoons [cAPK_{act}GPK_{inact}]$

$EQ9.1$ $[cAPK_{act}GPK_{inact}] + ATP \longrightarrow (GPK - P)_{act} + ADP + cAPK_{act}$

$EQ10$ $(GPK - P)_{act} + GP_{inact} \rightleftharpoons [(GPK - P)_{act}GP_{inact}]$

$EQ10.1$ $[(GPK - P)_{act}GP_{inact}] + ATP \longrightarrow (GP - P)_{act} + ADP + (GPK - P)_{act}$

$EQ11$ $(GP - P)_{act} + \quad Glycogen \rightleftharpoons (GP - P)_{act}Glycogen$

$EQ11.1$ $(GP - P)_{act}Glycogen + P_i \longrightarrow Glucose1 - P + (GP - P)_{act}$

$EQ12$ $(GPK - P)_{act} + cAPK_{act} + GS_{act} \rightleftharpoons [cAPK_{act}GS_{act}(GPK - P)_{act}]$

$EQ12.1$ $[cAPK_{act}GS_{act}(GPK - P)_{act}] + ATP \longrightarrow (GS - P)_{inact} + ADP + cAPK_{act} + (GPK - P)_{act}$

	Duration of a reaction		
Equations	**Mode 1**	**Mode 2**	**Mode 3**
EQ1: Receptor and G-protein activation	$k_f = 1$, $k_b = 1$	$k_f = 1$, $k_b = 1$	$k_f = 1$, $k_b = 1$
EQ2: G-protein and ligand dissociation	$k_f = 1$, $k_b = 1$	$k_f = 1$, $k_b = 1$	$k_f = 1$, $k_b = 1$
EQ3: GAP-driven GTPase process	$k_f = 1$, $k_b = 1$, $k_{hydr} = 1$	$k_f = 1$, $k_b = 1$, $k_{hydr} = 1$	$k_f = 1$, $k_b = 1$, $k_{hydr} = 1$
EQ4: G-protein-receptor inactive complex	$k_f = 1$, $k_b = 1$	$k_f = 1$, $k_b = 1$	$k_f = 1$, $k_b = 1$
EQ5: Receptor and ligand dissociation	$k_f = 1$, $k_b = 1$	$k_f = 1$, $k_b = 1$	$k_f = 1$, $k_b = 1$
EQ6: G-protein (active) and adenylate cyclase association	$k_f = 1$, $k_b = 1$	$k_f = 1$, $k_b = 1$	$k_f = 1$, $k_b = 1$
EQ7: cyclic AMP formation	$k_f = 1$, $k_b = 1$, $k_{form} = 1$	$k_f = 1$, $k_b = 1$, $k_{form} = 1$	$k_f = 1$, $k_b = 1$, $k_{form} = 1$
EQ8: $cAPK$ activation	$k_f = 1$, $k_b = 1$	$k_f = 1$, $k_b = 1$	$k_f = 1$, $k_b = 1$
EQ9;EQ9.1: GPK phosphorylation	$k_f = 1$, $k_b = 1$, $k_3 = 2$	$k_f = 1$, $k_b = 1$, $k_3 = 20$	$k_f = 1$, $k_b = 1$, $k_3 = 200$
EQ10;EQ10.1: GP phosphorylation	$k_f = 1$, $k_b = 1$, $k_5 = 2$	$k_f = 1$, $k_b = 1$, $k_5 = 20$	$k_f = 1$, $k_b = 1$, $k_5 = 200$
EQ11;EQ11.1: Degradation of glycogen to glucose 1-phosphate	$k_f = 1$, $k_b = 1$, $k_{degr} = 1$	$k_f = 1$, $k_b = 1$, $k_{degr} = 1$	$k_f = 1$, $k_b = 1$, $k_{degr} = 1$
EQ12;EQ12.1: GS phosphorylation	$k_f = 1$, $k_b = 1$, $k_7 = 2$	$k_f = 1$, $k_b = 1$, $k_7 = 20$	$k_f = 1$, $k_b = 1$, $k_7 = 200$

Table A.1
Reaction scheme in the signaling pathway of glycogen breakdown. The total time of the simulation was 1.000.000 time-units. The corresponding rate constant of a reaction is represented as its duration in discrete time units. Adapted from [5,22,23,27,34].

Variables	**Definition**	**Concentration (nM)**
$R_{glcgn}[G_{\alpha}GDP]$	GPCR-transm. glucagon receptor-G_{α} subunit	200;350;500
L_{glcgn}	Ligand (glucagon)	1;500;1000
$R_{glcgn}[G_{\alpha}GTP]L_{glcgn}$	GPCR-G Protein-Ligand Complex	200;350;500
$G_{\alpha}GTP$	G-protein GTP-bound alpha subunit (active)	200;1400;3000
$R_{glcgn}L_{glcgn}$	Ligand-receptor complex	1;250;500
GAP	GTP-hydrolysis enzyme	10;145;300
$[G_{\alpha}GTP]GAP$	GTP-hydrolysis complex	200;250;300
$G_{\alpha}GDP$	G-protein GTP-bound alpha subunit (inactive)	200;1400;3000
R_{glcgn}	GPCR- transmem. glucagon recep.	1;250;500
AC	Adenylate cyclase enzyme	1;250;500
$AC[G\alpha GTP]$	AC enzyme and G-protein complex	200;350;500
ATP	Adenosine triphosphate	4000;4000;4000
$[AC[G\alpha GTP]]ATP$	AC enzyme, G-protein and ATP complex	200;350;500
$cAMP$	Cyclic AMP	$1 \times 10^5; 4.5 \times 10^5; 1 \times 10^6$
P_i	Inorganic phosphorus	4000;4000;4000
$cAPK_{inact}$	cAMP dependent protein kinase (inactive)	1000;1000;1000
$cAPK_{act}$	cAMP dependent protein kinase (active)	1000;1000;1000
GPK_{inact}	Glycogen phosphorylase kinase (inactive)	3000;3000;3000
$[cAPK_{act}GPK_{inact}]$	$cAPK_{act}$GPK$_{inact}$ complex	2000;2000;2000
$(GPK - P)_{act}$	Phosphorylate glycogen phosphorylase kinase	3000;3000;3000
ADP	Adenosine diphosphate	4000;4000;4000
GP_{inact}	Glycogen phosphorylase (inactive)	$7 \times 10^4; 7 \times 10^4; 7 \times 10^4$
$[(GPK - P)_{act}GP_{inact}]$	glycogen phosphorylase kinase-glycogen phosphorylase complex	3000;3000;3000
$(GP - P)_{act}$	Glycogen phosphorylase (active)	$7 \times 10^4; 7 \times 10^4; 7 \times 10^4$
$Glycogen$	Glycogen	5×10^7;5×10^7;5×10^7
$(GP-P)_{act}Glycogen$	Glycogen phosphorylase-glycogen complex	$7 \times 10^4; 7 \times 10^4; 7 \times 10^4$
$Glucose1 - P$	Glucose 1-phosphate	5×10^7;5×10^7;5×10^7
GS_{act}	Glycogen synthase (active)	3000;3000;3000
$[cAPK_{act}GS_{act}(GPK-P)_{act}]$	Protein kinase glycogen synthase complex	1000;1000;1000
$(GS - P)_{inact}$	Glycogen synthase (inactive)	3000;3000;3000

Table A.2
Reacting species, their initial amount, and a model of biochemical reaction rules for the signaling pathway of glycogen breakdown. For each mode of simulation, we use the lower, medium and upper limit of the molar concentration of the variables: *InfLim*, *MedLim*, *MaxLim*, respectively. Adapted from [5,22,23,27,34].

B HIV infection cycle

Table B.1 describes the components of the system and the reactions considered for the simulation of the HIV cycle.

Variables	Definitions	Biochemical Reaction Rules
$gp120, gp41$	Glycoprotein 120 - On the surface of the HIV, glycoprotein 41 - Subunit of the envelope protein complex	EQ1: $gp120 + gp41 \rightarrow [gp120]gp41$
$cd4$	Cluster of differentiation 4 - Cell surface glycoprotein	EQ2: $[gp120]gp41 + cd4 \rightarrow [[gp120]gp41]cd4$
$ccr5$	C-C Chemokine receptor type 5 - Protein receptor for Chemokines	EQ3: $[[gp120]gp41]cd4 + ccr5 \rightarrow [[[gp120]gp41]cd4]ccr5$
$ectgp41$	Glycoprotein 41 ectodomain - Released by a sequential binding of gp120 to cd4 and coreceptor	EQ4: $[[[gp120]gp41]cd4]ccr5 \rightarrow ectgp41$
cpd	Capsid - A shell containing the virus information	EQ5: $ectgp41 \rightarrow 2cpd$
rT, int, prt	Reverse transcriptase, integrase, protease - Enzymes released into the cell by the virus	EQ6: $cpd \rightarrow rT + int + prt$
$sRNA$	Single-stranded Ribonucleic acid - genetic information of the virus	EQ7: $cpd \rightarrow 2sRNA$
$dDNA$	Double-stranded Deoxyribonucleic acid	EQ8: $2sRNA + rT \rightarrow dDNA$
		EQ9: $dDNA + int \rightarrow [dDNA]int$
gnm	Genome - genetic information of the cell	EQ10: $[dDNA]int + gnm \rightarrow [[dDNA]int]gnm$
$vDNA$	Viral Deoxyribonucleic acid - viral information produced by the genome	EQ11: $[[dDNA]int]gnm \rightarrow vDNA$
$mRNAHIV$	Viral messenger ribonucleic acid - viral information used in order to produce the viral proteins	EQ12: $vDNA \rightarrow mRNAHIV$
$pHIV$	HIV viral protein-protein to be divided into small pieces by a protease	EQ13: $prt + pHIV \rightarrow [prt]pHIV$
		EQ14: $[prt]pHIV \rightarrow 4prtv$
$prtv$	Viral proteins used to form new virus	EQ15: $prtv \rightarrow rTv$
		EQ16: $prtv \rightarrow prtvr$
		EQ17: $prtv \rightarrow intv$
		EQ18: $prtv \rightarrow 2sARNv$
$cpdv$	Viral capside - ontains the information needed in order to form a new virus	EQ19: $rTv + prtvr + intv \rightarrow cpdv$
		EQ20: $2sARNv \rightarrow cpdv$
$nVirus$	New virus - contains the information of a new virus	EQ21: $2cpdv \rightarrow nVirus$

Table B.1
Reacting species, their initial amount, and a model of intracellular processes for the HIV life cycle. Our simulation considers the parameter $t = 1$ (see Equation 2) and all the reactions have the same probability to occur. Initial concentrations were set up to 100 copies for *ccr*5, *cd*4, *gp*120, *gp*41, and 0 for complexes. Adapted from [18].