Available online at www.sciencedirect.com
SCIENCE DIRECT®
Theoretical Computer Science 325 (2004) 3–24
Theoretical Computer Science
www.elsevier.com/locate/tcs

A multi-scale constraint programming model of alternative splicing regulation☆

Damien Eveillard[a,b,*], Delphine Ropers[b], Hidde de Jong[c], Christiane Branlant[b], Alexander Bockmayr[a]

[a] *LORIA, Université Henri Poincaré, BP 239, 54506 Vandoeuvre-lès-Nancy, France*
[b] *Laboratoire de Maturation des ARN et Enzymologie Moléculaire, UMR 7567 CNRS-UHP, BP 239, 54506 Vandoeuvre-lès-Nancy, France*
[c] *INRIA Rhône-Alpes, Helix Project, 655 avenue de l'Europe, Montbonnot, 38334 Saint Ismier, France*

Abstract

Alternative splicing is a key process in post-transcriptional regulation, by which different mature RNA can be obtained from the same premessenger RNA. The resulting combinatorial complexity contributes to biological diversity, especially in the case of the human immunodeficiency virus HIV-1. Using a constraint programming approach, we develop a model of the alternative splicing regulation in HIV-1. Our model integrates different scales (single site vs. multiple sites), and thus allows us to exploit several types of experimental data available to us.

Keywords: Computational biology; Constraint programming; Modeling; Hybrid system; Alternative splicing

1. Introduction

Molecular biology is concerned with the study of three types of biological macromolecules: DNA, RNA, and proteins. Each of these molecules can initially be viewed as a string on a finite alphabet: DNA resp. RNA are nucleic acids, made up of nucleotides A,C,G,T resp. A,C,G,U. Proteins are sequences of amino acids. There exist twenty amino acids, which may be represented by an alphabet of 20 letters. Molecular biology studies the information flow from DNA to RNA, and from RNA to

☆ Part of this work was done within the ARC INRIA "Process Calculi and Biology of Molecular Networks", http://contraintes.inria.fr/cpbio
* Corresponding author.
E-mail address: eveillar@loria.fr (D. Eveillard).

0304-3975/$ - see front matter
doi:10.1016/j.tcs.2004.03.062

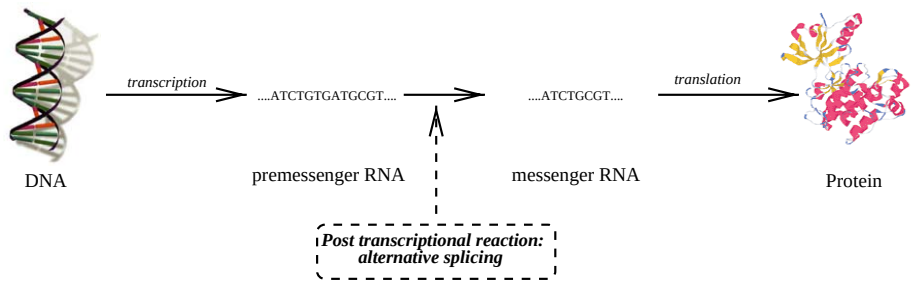

Fig. 1. Information flow in molecular biology.

proteins, see Fig. 1. In a first step, called *transcription*, a substring of DNA ("gene") is transformed into messenger RNA (mRNA). In the second step, called *translation*, the mRNA is translated into a protein, where each triplet of nucleotides encodes one amino acid ("genetic code").

In eukaryotes (i.e., organisms whose cells contain membrane-bound nuclei) and viruses, transcription is followed by another process, which is *alternative splicing* [15]. In a first step, the DNA molecule yields a *premessenger* RNA molecule, by constructing a single-stranded copy of the double-stranded DNA, and by replacing T's with U's. The premessenger RNA may be decomposed into a sequence of substrings called exons and introns. During *splicing*, introns are removed. The remaining exons are concatenated and yield the final *messenger* or *mature* RNA (mRNA). *Alternative* splicing means that through the elimination of selected introns and exons, different mature RNA may be obtained from the same premessenger RNA. In other words, through alternative splicing one and the same gene may code for a variety of proteins.

In our work, we are interested in the alternative splicing regulation of the human immunodeficiency virus (HIV-1). The different proteins that are obtained through this phenomenon play a crucial role in the virus life cycle. Our goal in developing a computational model of the alternative splicing regulation is to get a better understanding of the virus life cycle, in particular the transition from the early to the late phase.

Recent biological studies [8] show that the alternative splicing regulation in HIV-1 depends on a certain class of proteins, so-called *SR proteins* (SR stands for Serine ARginine rich). These proteins can be divided into two functional classes: they may activate or they may inhibit splicing. The knowledge currently available from experiments is limited. Each experiment focuses on one particular splicing site. In a first approach, we therefore model SR regulation in this restricted context. Using differential equations, we develop a continuous model for the regulation of the A3 splicing site in HIV-1. The qualitative behavior of the model depends on the values of the reaction kinetic parameters. Experimental results available to us validate this first approach in the equilibrium phase. In a second step, we integrate the continuous single-site model into a more global multi-site model that expresses the discrete switch from one splicing site to another. This model goes beyond currently available experimental data, and thus may indicate directions for further biological research. Our ultimate goal is to obtain a model that can be validated qualitatively both on the scale of a single splicing site and

on the scale of the whole HIV-1, and which represents the global effect of alternative splicing in the HIV-1 life cycle.

We build our models in a constraint programming framework [2,3]. Constraint programming seems well-suited for modeling biological systems because it allows one to handle partial or incomplete information. Each constraint gives one piece of information on the system that is studied. The overall knowledge is accumulated in the constraint store. The constraint engine available in constraint programming systems operates on the constraint store. It may add new information to the store or check whether some property is entailed by the information present in the store. While a constraint model may be refined whenever additional biological knowledge becomes available, it allows one to make useful inferences even from partial and incomplete information. Therefore, constraint programming seems to be a natural computational approach to face the current situation in systems biology as it is described by Palsson [18]: "Because biological information is incomplete, it is necessary to take into account the fact that cells are subject to certain constraints that limit their possible behaviors. By imposing these constraints in a model, one can then determine what is possible and what is not, and determine how a cell is likely to behave, but never predict its behavior precisely."

The organization of the paper is as follows: we start in Section 2 with a description of the biological process of alternative splicing regulation. Based on a number of biological hypotheses, we develop in Section 3 a continuous model of the regulation at one splicing site. This model includes competition and compensation of different proteins on two binding sites, ESE and ESS2. The single-site model is validated in a qualitative way by extracting from the model a splice efficiency function, which can be measured in experiments. In Section 4, we briefly present the hybrid concurrent constraint programming language `Hybrid cc` [9,10], and explain how it can be used for modeling dynamic biological systems. In Section 5, we first simulate the single-site continuous model in this language. Then we derive a more global model involving three generic splicing sites, which may be generalized to multiple sites. This means that we model at two different scales, using the splice efficiency function as a time-scale abstraction of the local model of one site in the more global context of different sites. The three-site model uses the constraint solving and default reasoning facilities of `Hybrid cc`. This allows us to make predictions on the global behavior even in the absence of detailed local information on some of the splicing sites.

2. Alternative splicing: a biological problem for formals methods

2.1. The biological problem of alternative splicing regulation

The regulation of the splicing process depends on different sites on the premessenger RNA. The first one is the *donor site SD*, located at the end of one exon, see Fig. 2. Its main characteristic is a `GU` nucleic acid sequence motif. The other site is the *acceptor site SA* located at the beginning of the next exon, which is characterized by an `AG` motif. Together, they define the intron to be excised from the premessenger RNA. They permit the binding of a huge ribonucleoproteic complex: the spliceosome.

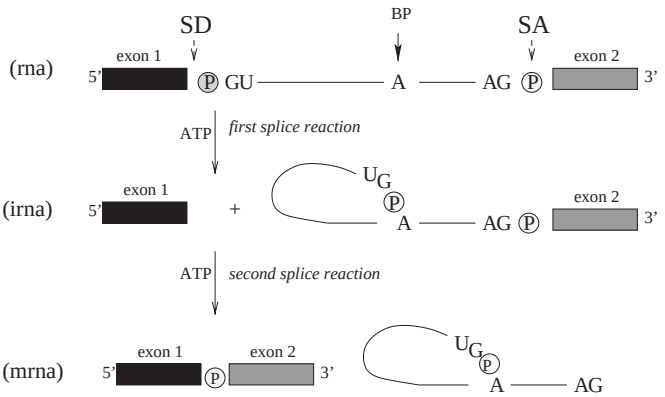

Fig. 2. The splicing process operates in the intronic region of a premessenger RNA that lies between two exons. The exons are delimited by the SD and SA binding sites. A first reaction cuts the RNA at the SD binding site. A second reaction cuts in the SA binding site. Each reaction requires ATP energy.

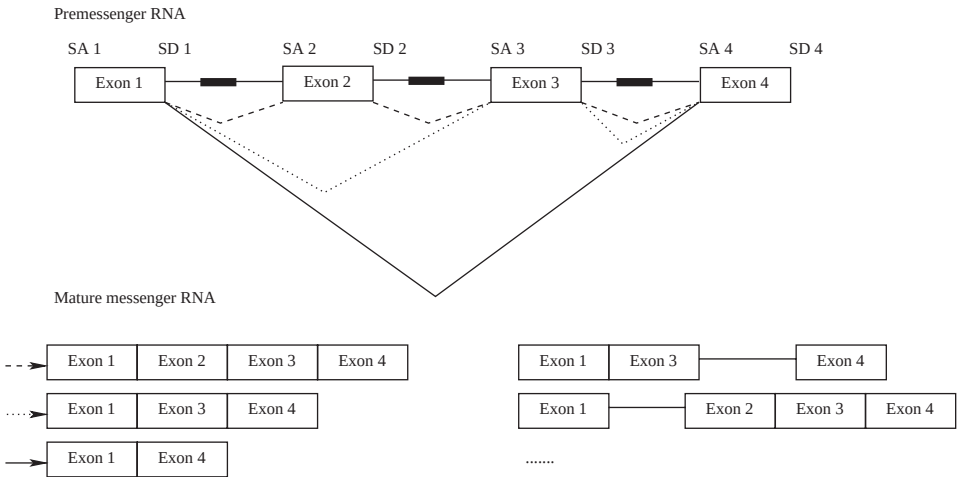

Fig. 3. Obtaining different mature RNA from the same premessenger RNA.

This complex is partially activated by another motif, the *branching point BP*. This is another binding site contained inside the intron. The three sites permit the regulation of the splicing process by activation of the spliceosome complex. The key to the regulation is the choice of one acceptor and one donor site. The splicing activity is determined by additional signals which activate or repress the splicing process.

Understanding the splicing process is a fundamental problem in molecular biology. As illustrated by Fig. 3, various messenger RNA can be obtained from a unique premessenger RNA through the elimination of different introns and exons, and the junction of the remaining exonic sequences. This process depends on the choice of the donor and the acceptor sites.

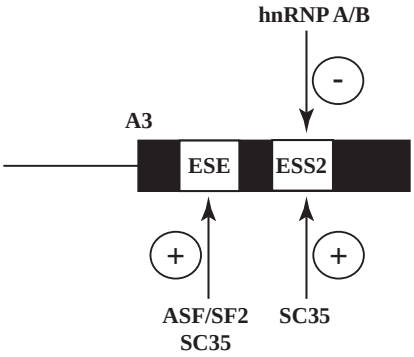

Fig. 4. Regulatory elements of the A3 splicing site. The exon delimited by the A3 acceptor site contains the ESE and ESS2 binding sites, which bind ASF/SF2, SC35 and hnRNP A/B proteins. These regulatory elements activate or repress the splicing reaction on the A3 site.

2.2. Alternative splicing in the context of HIV-1

In the life cycle of the human immunodeficiency virus HIV-1, splicing plays an important role. The viral RNA either remains unchanged to serve as genomic RNA for new virions, or it is spliced to allow for the production of virion proteins [25]. In the HIV-1 case, the alternative splicing regulation involves 4 donor sites (SD) and 8 acceptor sites (SA), which may yield 40 mature messenger RNAs [19]. This diversity is achieved by regulating the selection of the acceptor sites [17,19]. Protein factors such as hnRNP and SR proteins control the regulation via specific binding sites on the premessenger RNA. In general, SR proteins activate the splicing process by initializing the splicing machinery.

In our study, we focus on the acceptor site A3. Inside the A3 splicing site, we distinguish two protein binding sites, ESE and ESS2, see Fig. 4. Splicing can be repressed by hnRNP A/B proteins via the ESS2 binding site [4,7]. Splicing can be activated by the SR proteins SC35 and ASF/SF2 via the ESE binding site [20,21]. However, SC35 can also bind to the ESS2 site. The hypothesis underlying our model is that the ratio of hnRNP A/B and SR proteins determines the splice efficiency at the A3 site.

3. Modeling one splicing site

3.1. Biological hypotheses

We model the regulation by SR proteins in the restricted context of the A3 splicing site under the following hypotheses, see Fig. 4:

- We study only one splicing site. Thus, we consider regulation at the scale corresponding to our experimental results, which are measurements of the splice efficiency given as the ratio of mature RNA over premessenger RNA.

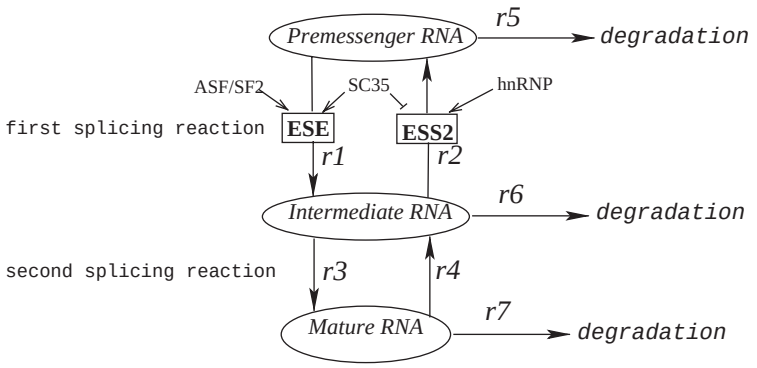

Fig. 5. Schematic representation of the splicing site regulation. The two splicing reactions are composed of 7 kinetic reactions.

- We suppose that the splicing process involves two reactions, relating three functional classes of RNA, see Fig. 2: immature RNA (`rna`), intermediate RNA (`irna`), and mature RNA (`mrna`). Intermediate RNA corresponds to immature RNA activated by proteins. Mature RNA corresponds to mature RNA and introns in lariat.
- The protein concentration in experiments is saturated. Therefore, we assume that it is constant, despite the binding of proteins to the RNA during regulation.
- SR proteins regulate the splicing process by initialization of the splicing machinery.
- Regulation is controlled by the ESE and ESS2 binding sites, which are independent.
- The SR proteins ASF/SF2 and SC35 may activate the first splicing reaction by binding to the site ESE. We assume that these two proteins compensate each other.
- The hnRNP A/B proteins may inhibit the first splicing reaction by binding to the site ESS2. On the other hand, if the SC35 proteins bind to ESS2, this inhibits the hnRNP A/B effect. Therefore we have a competitive inhibition between hnRNP A/B and SC35.

These biological hypotheses are summarized in Fig. 5.

3.2. Mathematical model

Our biological hypotheses can be represented by a system of ordinary differential equations inspired from a model by Monod [14]. In this model, the rate increase depends on the external concentrations, which are limiting factors, and is controlled by a Michaelis–Menten-type kinetics. In our case, we assume that the regulatory protein concentrations are the limiting factors. Such an approach is generally used in ecological modeling, and is well-suited to describe systems that are only partially known.

Table 1
Symbols and units for the biological variables and parameters

Symbol	Variables and parameters	Unit
rna	Immature RNA	μM
irna	Intermediate RNA	μM
mrna	Mature RNA	μM
ASF	Protein ASF/SF2	μM
SC	Protein SC35	μM
R	Protein hnRNP A/B	μM
φ_{ESE}	Maximal affinity for the enhancer	s^{-1}
φ_R	Maximal affinity of hnRNP A/B	s^{-1}
k_{ESE}	Half saturation coefficient for the enhancer	μM
k_{SC}	Half saturation coefficient for SC35	μM
k_R	Half saturation coefficient for hnRNP A/B	μM
κ	Reaction rate	s^{-1}
κ'	Reaction rate	s^{-1}
λ	Degradation coefficient	s^{-1}

The single-site model that we obtain will later be integrated into a larger multi-site model, see Section 5. We will describe the splicing process by seven kinetic reactions. The symbols used are given in Table 1.

The reaction r_1 represents the transformation of premessenger RNA to intermediate RNA. It requires cooperation between ASF/SF2 and SC35 proteins for the regulation of ESE. Since we assume compensation, only the sum of the two activator proteins is important. We represent the reaction rate by a Michaelis–Menten function depending on the quantity of immature RNA, and controlled by the sum of the proteins ASF/SF2 and SC35. The generic form of the Michaelis–Menten function, see e.g. [16], is:

$$v = \frac{v_{\max} x}{k_m + x}.$$

The curve expressing the relationship between v and x is given in Fig. 6. Here, $v_{\max}$ is the maximum rate, and the Michaelis constant k_m is the value at which v is half maximal. With $x = ASF + SC$ and the notation from Table 1, we get:

$$r_1 = \frac{\varphi_{ESE}(ASF + SC)}{k_{ESE} + (ASF + SC)} rna.$$

The reaction r_2 represents the transformation of intermediate RNA to premessenger RNA. It captures the antagonistic function of hnRNP A/B and SC35 proteins on the site ESS2. We use a similar function as before. However, we now have a *competitive inhibition*, see e.g. [13], between two species x and y. The generic form becomes

$$v = \frac{v_{\max} x}{k_m(1 + \frac{y}{k_y}) + x}.$$

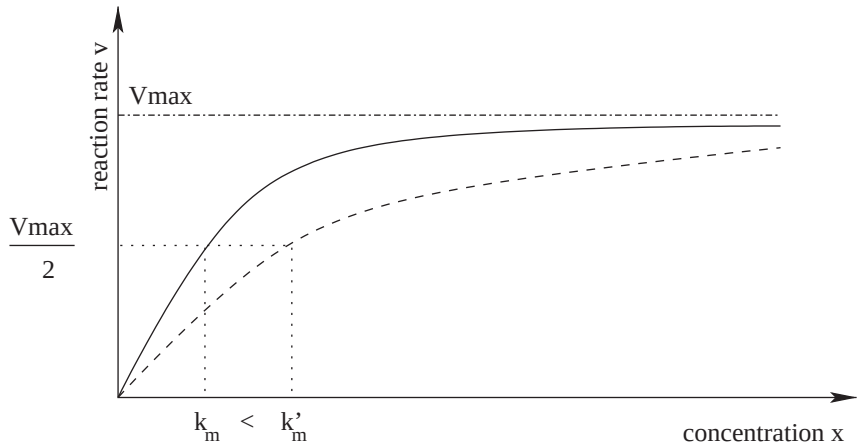

Fig. 6. Michaelis–Menten function.

The factor $(1 + y/k_y)$ increases the value of k_m, see Fig. 6 for illustration. With $x = R$ and $y = SC$ and the notation from Table 1, we get

$$r_2 = \frac{\varphi_R R}{k_R(1 + \frac{SC}{k_{SC}}) + R}\, irna.$$

The reaction r_3 represents the transformation of intermediate RNA to mature RNA (mrna). We assume for this reaction a simple first-order kinetics with a constant parameter κ. Similarly, r_4 represents the reaction which transforms mature RNA to intermediate RNA:

$$r_3 = \kappa irna, \quad r_4 = \kappa' mrna.$$

r_5, r_6 and r_7 respectively, represent the degradation reaction of immature RNA, intermediate RNA and mature RNA. Different RNAs decrease proportionally to the same degradation factor λ

$$r_5 = \lambda rna, \quad r_6 = \lambda irna, \quad r_7 = \lambda mrna.$$

We formalize the splicing process at site A3 by the system of differential equations, see again Fig. 5,

$$\frac{\mathrm{d}(rna)}{\mathrm{d}t} = r_2 - r_1 - r_5,$$
$$\frac{\mathrm{d}(irna)}{\mathrm{d}t} = r_1 + r_4 - r_2 - r_3 - r_6,$$
$$\frac{\mathrm{d}(mrna)}{\mathrm{d}t} = r_3 - r_4 - r_7,$$

which corresponds to

$$\frac{\mathrm{d}(rna)}{\mathrm{d}t} = \frac{\varphi_R R}{k_R(1+\frac{SC}{k_{SC}})+R}\, irna - \frac{\varphi_{ESE}(ASF+SC)}{k_{ESE}+(ASF+SC)}\, rna - \lambda rna,$$

$$\begin{aligned}\frac{\mathrm{d}(irna)}{\mathrm{d}t} &= \frac{\varphi_{ESE}(ASF+SC)}{k_{ESE}+(ASF+SC)}\, rna - \frac{\varphi_R R}{k_R(1+\frac{SC}{k_{SC}})+R}\, irna \\ &\quad - \kappa irna + \kappa' mrna - \lambda irna,\end{aligned}$$

$$\frac{\mathrm{d}(mrna)}{\mathrm{d}t} = \kappa irna - \kappa' mrna - \lambda mrna.$$

3.3. Validation of the regulatory system

The mathematical model of regulation at the acceptor site A3 can be directly simulated in the constraint programming language `Hybrid cc`, as will be shown in Section 4. However, it should first be validated with respect to existing biological knowledge [1].

In our model, the RNA concentrations do not reach an equilibrium, i.e., a state in which no further net change is occurring, but continue to decrease until total degradation of RNA. However, we may assume that the splicing reactions quickly reach an equilibrium. In the equilibrium phase, we have $r_1 = r_2$, $r_3 = r_4$, which is equivalent to

$$\frac{\varphi_{ESE}(ASF+SC)}{k_{ESE}+(ASF+SC)}\, rna = \frac{\varphi_R R}{k_R(1+\frac{SC}{k_{SC}})+R}\, irna, \quad \kappa irna = \kappa' mrna.$$

If we define the *splice efficiency* by

$$efficiency(t) = \frac{mrna(t)}{rna(t)},$$

we obtain the following formula for the splice efficiency in the equilibrium phase:

$$efficiency_{\mathrm{eq}} = \frac{\kappa\varphi_{ESE}(ASF+SC)(k_R k_{SC} + k_R SC + R k_{SC})}{\kappa'(k_{ESE}+ASF+SC)\varphi_R R k_{SC}}.$$

According to our formula, the splice efficiency is
- an increasing function of the activators SC and ASF.
- a decreasing function of the inhibitor R.

Experimental results show that
- $(mrna/rna)_{\mathrm{eq}}$ increases with an increase of activator proteins.
- $(mrna/rna)_{\mathrm{eq}}$ decreases with an increase of inhibitor proteins.

Thus, the results of our model correlate with available experimental data. Therefore, we may consider the model to be qualitatively validated under the hypotheses described in Section 3.1. We next consider simulation in the concurrent constraint language `Hybrid cc`.

4. Hybrid concurrent constraint programming

To model alternative splicing regulation, we will use hybrid concurrent constraint programming, Hybrid cc [9,10]. The general idea of *constraint programming* for system modeling is that the user specifies constraints on the behavior of the system that is being studied. Each constraint expresses some partial information on the system state. The constraint solver may check constraints for consistency or infer new constraints from the given ones. In *concurrent constraint programming* (cc), different computational processes may run concurrently. Interaction is possible via the *constraint store*. The store contains all the constraints currently known about the system. A process may *tell* the store a new constraint, or *ask* the store whether some constraint is entailed by the information currently available, in which case further action is taken [22]. One major difficulty in the original cc framework is that cc programs can detect only the presence of information, not its absence. To overcome this problem, Saraswat et al. [23] proposed to add to the cc paradigm a sequence of phases of execution. At each phase, a cc program is executed. At the end, absence of information is detected, and used in the next phase. This results in a synchronous reactive programming language, Timed cc. But, the question remains how to detect negative information instantaneously. Default cc extends cc by a negative ask combinator if c else A, which imposes the constraints of A unless the rest of the system imposes the constraint c. Logically, this can be seen as a default. Introducing phases as in Timed cc leads to Timed Default cc [24]. Only one additional construct is needed: hence A, which starts a copy of A in each phase after the current one.

Hybrid cc [9,10], is an extension of Default cc over continuous time. First continuous constraint systems are allowed, i.e., constraints may involve differential equations that express initial value problems. Second, the hence operator is interpreted over continuous time. It imposes the constraints of A at every real time instant after the current one. The evolution of a system in Hybrid cc is piecewise continuous, with a sequence of alternating point and interval phases. All discrete changes take place in a point phase, where a simple Default cc program is executed. In a continuous phase, computation proceeds only through the evolution of time. The interval phase, whose duration is determined in the previous point phase, is exited as soon as the status of a conditional changes [10]. Table 2 summarizes the basic combinators of Hybrid cc.

It has been argued in [2,3] that Hybrid cc is well-suited for modeling dynamic biological systems. In addition to the general discussion in [3], we illustrate here by a number of small examples, how the basic combinators of Hybrid cc can be applied naturally to the study of biological systems.

4.1. Interval constraints and continuous dynamics

The Hybrid cc language that we are using is based on interval constraints [5]. This means that variables are defined over an interval of real numbers, and computations are done in interval arithmetic. This is very useful in biology, where typically parameters and values are not exactly known.

Table 2
Combinators of `Hybrid cc`

Agents	Propositions
c	*c* holds now
`if` *c* `then` *A*	if *c* holds now, then *A* holds now
`if` *c* `else` *A*	if *c* does not hold now, then *A* holds now
`new` *X* `in` *A*	the variable *X* is local to *A* (hiding)
(*A*, *B*)	both *A* and *B* hold now
`hence` *A*	*A* holds at every instant after now
`always` *A*	same as (*A*, `hence` *A*)
`unless`(*c*) *A* `else` *B*	same as (`if` *c* `then` *B*, `if` *c* `else` *A*)

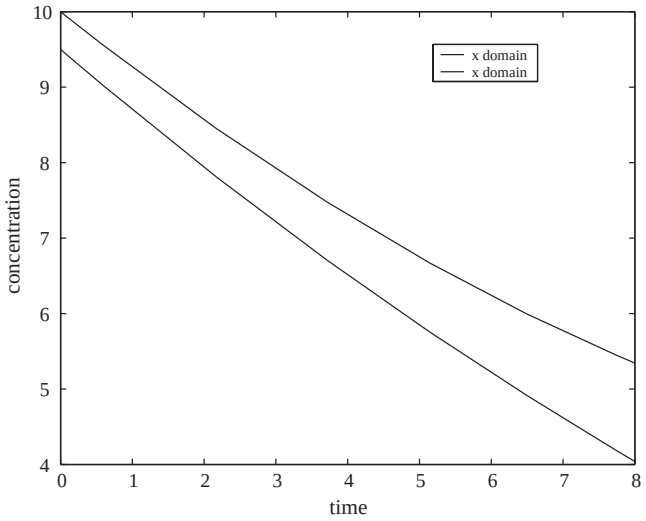

Fig. 7. Enclosure for the dynamics of a molecular species with linear kinetics.

We illustrate this by a very simple example in `Hybrid cc` involving a single constraint on an interval variable *x*, see Fig. 7. Since we are reasoning about dynamical systems, we use the `always` *A* combinator, expressing that *A* holds at every time instant.

```
interval x;
x = [9.5,10];
always { x' = -(2*x)/(15+x);
       }
sample(x);
```

4.2. Parallel composition

Hybrid cc allows for parallel composition of constraints. (A, B) imposes the constraints of both A and B. Operationally, the program (A, B) behaves like the simultaneous execution of both A and B. A and B may share common variables, and thus communicate via the constraint store.

We illustrate parallel composition by a small Hybrid cc program specifying a Michaelis–Menten kinetics. Consider two molecular species X and Y with concentrations x and y, and suppose X is transformed into Y. The initial concentration of X lies in the interval $[14, 14.5]$. The production rate of Y depends on the concentration of X according to the formula $y' = (v_{\max} * x)/(k_m + x)$, for some constants $v_{\max}$ and k_m. The concentration of X is reduced at the same rate. We add constraints $x, y \geqslant 0$ to say that concentrations are non-negative, and constraints $s = x + y$, $s' = 0$ to express conservation of matter. The constraint solver computes enclosures for x and y, see Fig. 8. In particular, we can observe that at the end of the experiment, the concentration of y will be greater than the concentration of x. Interval constraints are particularly useful in sensibility studies, where we can easily test the importance of one variable compared to the others.

```
#define km 1.5
#define vmax  2
interval x,y,s;
x = [14,14.5];              /* Initialization */
y = 0;
always {
   x' = -(vmax*x)/(km+x); /* Michaelis-Menten kinetics */
   y' =  (vmax*x)/(km+x);
   x >= 0;                /* Non-negative concentrations */
   y >= 0;
   s = x + y;             /* Conservation of matter */
   s' = 0;
}
sample(x, y);
```

4.3. Conditionals and discrete change

In general, the dynamics of a system will depend on conditions. In Hybrid cc, we may use the combinator if c then A expressing that if c holds now, then A holds now. This allows one to make discrete changes to switch from one dynamics to another. The next program models the situation that the transformation of X to Y gets activated if a certain protein P reaches a threshold, see Fig. 9 (top) for illustration.

```
interval x, y, p;
x=[14,14.5];
y=0;
```

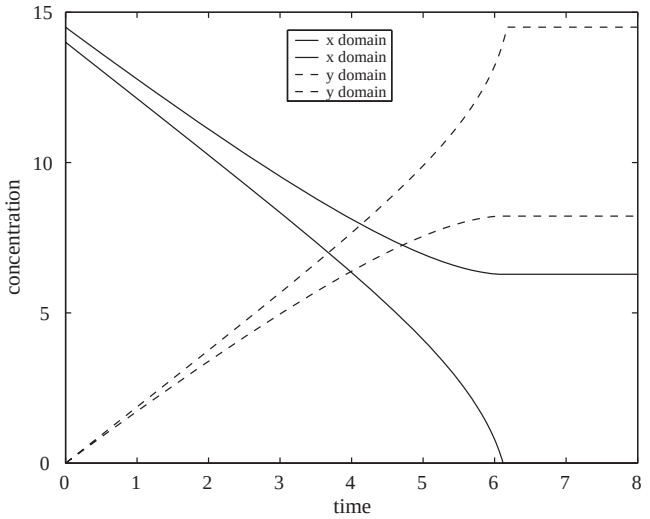

Fig. 8. Enclosures for the dynamics of two molecular species with Michaelis–Menten kinetics. Due to the constraints $x, y \geqslant 0, (x + y)' = 0$, the domain bounds get constant when the lower bound of x reaches 0.

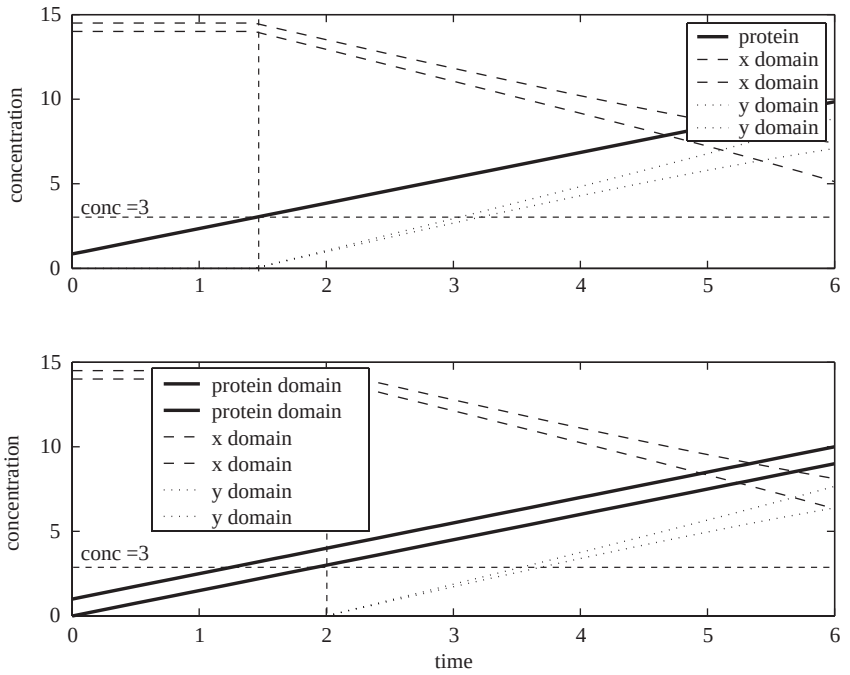

Fig. 9. Switching behavior for conditional (top) and default combinator (bottom).

```
p=0.75;
always {
   p' = 1.5;
   if (p >= 3)
      { x' = -(vmax*x)/(km+x);
        y' =  (vmax*x)/(km+x);
   }
   if (p < 3)
      { x' = 0;
        y' = 0;
   }
}
sample(p, x ,y);
```

Here, we have assumed that there is no uncertainty on the initial value of p. Without this hypothesis, the constraint solver cannot decide between the two alternatives $p \geqslant 3$ and $p < 3$. In order to handle conditions in the presence of uncertainty, we use default reasoning that we describe next.

4.4. Default behavior

The default combinator `if` c `else` A (or `unless`(c) A) expresses that A holds now, if c will not hold now. Operationally, this means that the current store on quiescence does *not* entail c. Note that `unless`(c) A is not equivalent to `if` $\neg c$ `then` A. If A is executed, this may have two reasons:

- The current store entails $\neg c$ (in this case `unless`(c) A behaves like `if` $\neg c$ `then` A), or
- the current store neither entails c nor $\neg c$, i.e., it is not known whether or not c holds. In this case, A is executed *by default*.

A is not executed, if the current store entails c.

We use the same example as before. The only difference is that the variable p representing the protein concentration is initialized with the interval $[0, 1]$. As we can see in Fig. 9 (bottom), the reaction gets activated when the *lower* bound for p reaches the threshold.

```
interval x, y, p;
x=[14,14.5];
y=0;
p=[0,1];
always {
   p' = 1.5;
   if ( p >= 3 )
      { x' = -(vmax*x)/(km+x);
        y' =  (vmax*x)/(km+x);
   }
```

```
   unless (p >= 3)
      { x' = 0;
        y' = 0;
   }
}
sample(p, x ,y);
```

The default combinator is a convenient way of handling incomplete knowledge in biology. In particular, we will use it in our multi-site model of alternative splicing regulation in Section 5.2.

5. Modeling the alternative splicing regulation with Hybrid cc

5.1. Single-site model: local modeling

The single-site model from Section 3.2 with experimental values can be expressed directly in `Hybrid cc`.

```
# define Pese 0.01              # define kr 0.01
# define Psc 0.2                # define k 0.19
# define Pr 0.4                 # define kk 0.01
# define kese 0.35              # define SC 2
# define ksc 2                  # define ASF 1.75
# define R 0.35

interval t, rna, irna, mrna;
t=0; rna = 0.06; irna = 0; mrna = 0;
always{
   rna'  =  (Pr*R*irna)/(kr*(1+(SC/ksc))+R)
           -(Pese*(ASF+SC)*rna)/(kese+ASF+SC)
           -delta*rna;
   irna' =  (Pese*(ASF+SC)*rna)/(kese+ASF+SC)
           -(Pr*R*irna)/(kr*(1+(SC/ksc))+R)
            -k*irna+kk*mrna-delta*irna;
   mrna' = k*irna-kk*mrna-delta*mrna;
}
sample(rna, irna, mrna);
```

During the simulation, we obtain for the splice efficiency *mrna*/*rna* the equilibrium predicted in Section 3.3, see Fig. 10. Under our hypotheses, which include protein competition and compensation, the model correctly simulates the alternative splicing activity at site A3. This supports the hypotheses made in the model such as the role of the ESE and ESS2 binding sites.

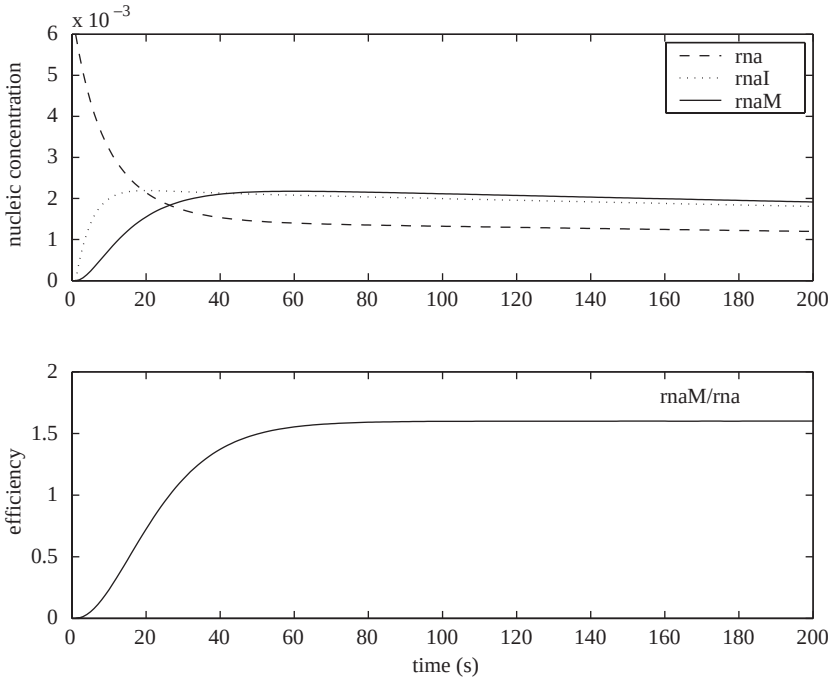

Fig. 10. Variation of RNA pool and splice efficiency in the splicing reaction.

5.2. Three-site model: global modeling

A realistic model of alternative splicing has to reflect the combinatorial complexity discussed in Section 2.2. Assuming that regulation is modular [12], the single-site model may be seen as one module inside a larger framework. The qualitative validation given in Section 3.3 justifies the introduction of the single-site model into a larger-scale model involving several splicing sites. To illustrate this, we consider the generic example of three acceptor sites (A3, A4 and A7) associated with one donor site (SD), see Fig. 11.

Using time-scale abstraction, the behavior at one splicing site is captured by a single function, the splice efficiency, which depends on the protein concentrations. This function is used in a larger-scale global model that describes the choice between three acceptor sites A3, A4 and A7. In the HIV-1 case, the A4 site is the default splicing site. Only if the efficiency of A3 (`effA3`) or A7 (`effA7`) gets larger than the efficiency of A4 (`effA4`), regulation switches to the other state. The sites A3, A4, and A7 exhibit three generic behaviors, see also Fig. 12:

- A3 is a regulated site with known behavior.
- A7 is a regulated site with unknown behavior.
- A4 is an unregulated site, i.e., the behavior does not depend on protein concentrations.

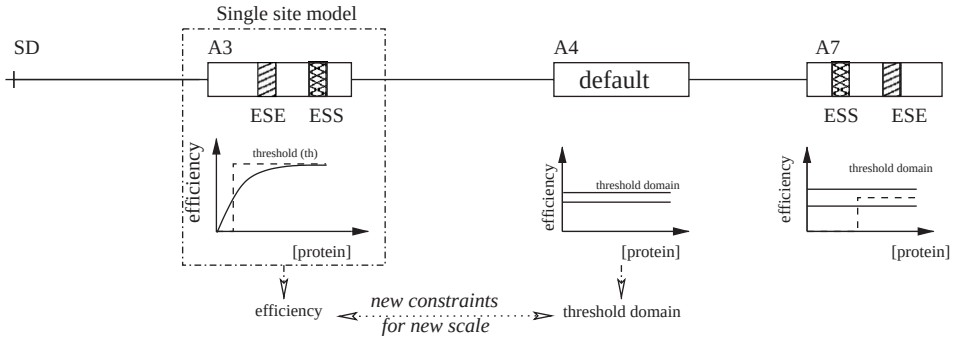

Fig. 11. Single-site model inside a more general multi-site regulation model.

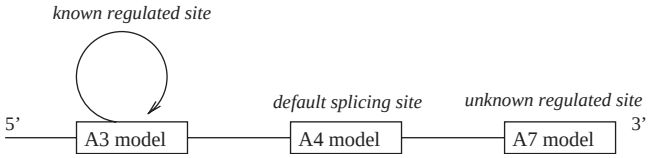

Fig. 12. Biological information on three acceptor sites A3, A4, A7.

Current biological experiments give information on the local behavior at one site. However, modeling the local behavior is not enough. In order to understand the global splicing process, we must integrate several types of knowledge. On the one hand, we have information on the local behavior at individual acceptor sites. On the other hand, we have some information on the global behavior, like the default role of A4 or the competition between different acceptor sites. Constraint programming allows us to integrate this information, and to produce a global model.

Recent work [6] shows the linearity of the splicing kinetics. Thus, on the larger scale, we may consider splicing as a linear process described by three systems of ordinary differential equations. For each acceptor site Ai, $i \in \{3,4,7\}$, we introduce one system with four differential equations:

- r_{i1} represents the consumption of immature RNA if Ai is dominating.
- r_{i3} represents the production of mature RNA at A3.
- r_{i4} represents the production of mature RNA at A4.
- r_{i7} represents the production of mature RNA at A7.

k_{ij} is the kinetic constant for reaction r_{ij}.

A4 is the default splicing site. It is dominating unless the splice efficiency of A3 or A7 gets larger than the splice efficiency of A4. If this happens, A7 becomes the default splicing site unless the efficiency of A3 gets larger than the efficiency of A7, see Fig. 13. The local behavior at A3 has been described by the single-site model given in Section 5.1. This model predicts the splice efficiency of A3 depending on the protein concentrations.

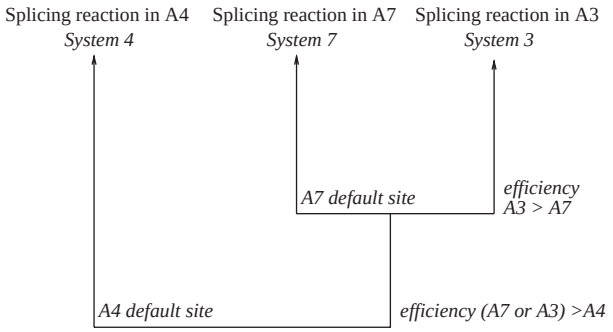

Fig. 13. Choice of the acceptor site A3, A4 or A7 depending on the splice efficiency.

In the Hybrid cc program given below, the concentration of SC35 is increased linearly. Depending on the corresponding variation of the splice efficiency at A3, the three-site model exhibits different behaviors, characterized by the choice of one of the three differential equation systems. The default behavior discussed before can be expressed naturally in Hybrid cc using the combinator unless(c) A.

```
#define c1 2
#define c2 0.5
#define c3 0.8
#define c4 1
#define c5 0.9
#define c6 0.1
#define R 3.5

interval t, prot, effA3, effA4, effA7, rna, mrnaA3,
    mrnaA4, mrnaA7;
t = 0;
rna = 10;
mrnaA3 = 0; /*known regulated acceptor site */
mrnaA4 = 0; /*unregulated acceptor site*/
mrnaA7 = 0; /*unknown regulated acceptor site*/

always { t' = 10;
prot = 0.1*t;                 /* protein variation */

/* A3 efficiency depends on the protein concentration
   A3 efficiency represents the local behavior of A3
   (observer of A3) */
effA3 = c1*(prot+c2)*(c3*prot+c4)/(c5*(prot+c6));

6 <= effA4; effA4 <= 8;  /* effA4 : efficiency domain of A4*/
7 <= effA7; effA7 <= 9;  /* effA7 : efficiency domain of A7*/
}
```

```
/* The behavior depends on the efficiency of
   the 3 acceptor sites*/

always {
/* if A3 or A7 dominant */
if (effA3 >= effA4 || effA7 >= effA4) {
      if (effA7 <= effA3)  {        /* splicing on A3 */
          rna' = -0.51 * rna - 0.01*rna;
       mrnaA3' =  0.4  * rna - 0.1*mrnaA3;  /* A3 kinetics */
       mrnaA4' =  0.01 * rna - 0.1*mrnaA4;  /* A4 kinetics */
       mrnaA7' =  0.1  * rna - 0.1*mrnaA7;  /* A7 kinetics */
      };
      unless ((effA7 <= effA3)) { /*default splicing on A7*/
          rna' = -0.51 * rna - 0.01*rna;
       mrnaA3' =   0.1 * rna - 0.1*mrnaA3; /* A3 kinetics */
       mrnaA4' =  0.01 * rna - 0.1*mrnaA4; /* A4 kinetics */
       mrnaA7' =   0.4 * rna - 0.1*mrnaA7; /* A7 kinetics */
      };
   };
/* default splicing on A4 */
unless (effA3 >= effA4 || effA7 >= effA4) {
     rna' = -0.32 * rna - 0.01*rna;
  mrnaA3' = -0.01 * rna - 0.1*mrnaA3; /* A3 kinetics */
  mrnaA4' =  0.3  * rna - 0.1*mrnaA4; /* A4 kinetics */
  mrnaA7' = -0.01 * rna - 0.1*mrnaA7; /* A7 kinetics */
   };
};
sample(prot, effA3, rna, mrnaA3, mrnaA4, mrnaA7);
```

According to the semantics of the default combinator, the A4 site will be chosen if the solver cannot deduce that (effA3 ⩾ effA4) or (effA7 ⩾ effA4). This may have *two* reasons:

- (effA3 ⩾ effA4) or (effA7 ⩾ effA4) is false, i.e., (effA3 < effA4) and (effA7 < effA4), or
- it is not known whether (effA3 ⩾ effA4) or (effA7 ⩾ effA4) holds (default behavior).

Thus A4 is the default site if the splice efficiency of A3 and A7 is not sufficiently high. If A3 or A7 dominate A4, then A7 is the default splicing site, unless A3 dominates A7.

Simulation in Hybrid cc yields the behavior shown in Fig. 14. First mrnA4 is produced, i.e., the default site A4 is active. When effA3 passes the upper threshold for effA4, site A7 gets activated, and mrnA7 is produced. Finally, when effA3 further increases and passes the upper threshold for effA7, site A3 gets activated and we observe production of mrnA3, while the concentrations of mrnA4 and mrnA7 become stationary.

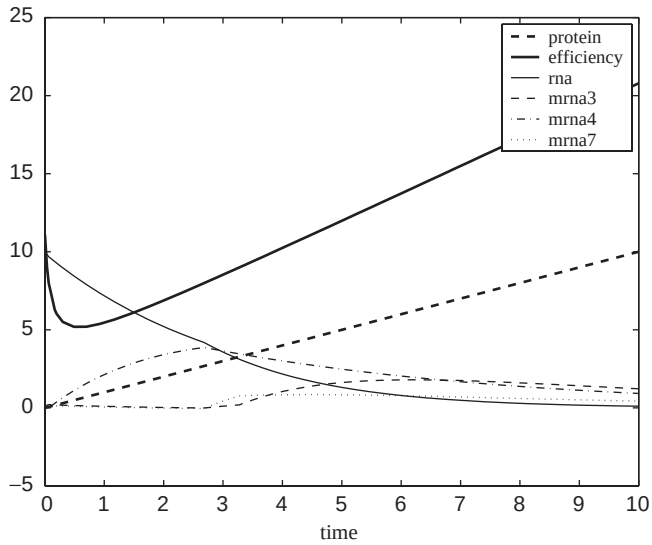

Fig. 14. Variation of mRNA production depending on a variation of SR proteins.

Basically, the model gives to the biologist three qualitative states: first splicing at the A4 site, second splicing at the A7 site, and finally splicing at the A3 site. The constraint programming system can compute enclosures for the three biological states, despite the variation in the concentrations of SR proteins. The enclosure is an important qualitative information to extend the single-site to a multi-site model. `Hybrid cc` permits a qualitative validation of the model, although the currently available information on the alternative splicing regulation in HIV-1 is incomplete.

6. Conclusion and further research

Our approach combines mathematical and computational methods. Mathematical analysis allows us to validate the single-site model in a qualitative way, based on the experimental data obtained in our group. The validation shows the consistency of our biological hypotheses. In a second step, we can extract the splice efficiency as a time-scale abstraction of the local behavior at one site inside a more global model involving different sites. For the experimental biologist, the single-site model may serve as a computational tool to evaluate his knowledge on a fine-grained biological process.

On the computational side, the constraint solving and default reasoning capabilities of `Hybrid cc` allow us to exploit as much as possible the incomplete knowledge of our system. Default behavior may compensate the lack of experimental data. Using constraint programming, we can delimit with our model the possible splicing behavior. This provides a powerful tool for qualitative validation.

Combining mathematical analysis and computational methods is the key to extending the single-site model to a multi-site model as described in this paper. It leads to the

qualitative validation represented by the extraction of the splice efficiency function. The splice efficiency characterizes the modularity of the regulation. Thus, the one-site behavior is represented in the three-site model, based on the single-site splice efficiency. The extraction of a suitable criterion on the smaller scale is crucial to understanding an experimental process from a systems biology perspective. Furthermore, constraints can be used to handle the problem of missing data in time-scale abstraction of a single-site model in a more global multi-site model. Different scales usually correspond to biological experiments yielding different types of results. Despite the variety of possible experiments, these must be integrated into a global model in order to better understand the biological process.

Modeling alternative splicing requires a close interaction between biological and computational approaches. In the context of alternative splicing regulation, we are currently working on new experimental data for the quantitative validation of our models. On the computational side, we have integrated our model into a general model of the HIV-1 life cycle [11]. Preliminary results show that the modification of a splice constant may induce different behaviors in the HIV-1 life cycle model. Using the extended model, we may validate several biological hypotheses on the global effect of alternative splicing in the full HIV-1 life cycle.

Acknowledgements

The authors would like to thank Arnaud Courtois for his comments on a draft of this paper.

References

[1] O. Bernard, J.-L. Gouzé, Nonlinear qualitative signal processing for biological systems: application to the algal growth in bioreactors, Math. Biosci. 157 (1999) 357–372.
[2] A. Bockmayr, A. Courtois, Modeling biological systems in hybrid concurrent constraint programming (abstract), in: The Second Internat. Conf. Systems Biology, ICSB'01, Pasadena, CA, 2001, p. 106.
[3] A. Bockmayr, A. Courtois, Using hybrid concurrent constraint programming to model dynamic biological systems, in: 18th Internat. Conf. on Logic Programming, ICLP'02, Copenhagen, Lecture Notes in Computer Science, Vol. 2401, Springer, Berlin, 2002, pp. 85–99.
[4] M. Caputi, M. Mayeda, A. Krainer, A. Zahler, hnRNP A/B proteins are required for inhibition of HIV-1 pre-mRNA splicing, EMBO 18 (14) (1999) 4060–4067.
[5] B. Carlson, V. Gupta, Hybrid cc and interval constraints, in: Hybrid Systems: Computation and Control, HSCC'98, Lecture Notes in Computer Science, Vol. 1386, Springer, Berlin, 1998, pp. 80–95.
[6] A. Audibert, D. Weil, F. Dautry, In vivo kinetics of mRNA splicing and transport in mammalian cells, Molecular Cell Biol. 22 (2002) 6706–6718.
[7] F. Del Gatto-Konczak, M. Olive, M. Gesnel, R. Breathnach, hnRNP A1 recruited to an exon in vivo can function as an exon splicing silencer, Molecular Cell. Biol. 19 (1) (1999) 251–260.
[8] B. Graveley, Sorting out the complexity of SR protein functions, RNA 6 (2000) 1197–1211.
[9] V. Gupta, R. Jagadeesan, V. Saraswat, Computing with continuous change, Sci. Comput. Programming 30 (1–2) (1998) 3–49.
[10] V. Gupta, R. Jagadeesan, V. Saraswat, D.G. Bobrow, Programming in hybrid constraint languages, in: Hybrid Systems II, Lecture Notes in Computer Science, Vol. 999, Springer, Berlin, 1995, pp. 226–251.

[11] B. Hammond, Quantitative study of the control of HIV-1 gene expression, J. Theoret. Biol. 163 (1993) 199–221.
[12] L. Hartwell, J. Hopfield, S. Leibler, A. Murray, From molecular to modular cell biology, Nature 402 (1999) C47–C52.
[13] R. Heinrich, S. Schuster, The Regulation of Cellular Systems, Thomson Publishing, New York, 1996.
[14] J. Monod, La technique des cultures continues. Théorie et applications, Ann. Inst. Pasteur 79 (1950) 390–410.
[15] M. Moore, C. Query, P. Sharp, Splicing of precursors to mRNA by the spliceosome, in: R. Gesteland, J. Atkins (Eds.), The RNA World, Cold Spring Harbor Laboratory Press, New York, 1993, pp. 303–357.
[16] J.D. Murray, Mathematical Biology I, An Introduction, 3rd Edition, Springer, Berlin, 2002.
[17] M. O'Reilly, M. McNally, K. Beemon, Two strong 5' splice sites and competing, suboptimal 3' splice sites involved in alternative splicing of human immunodeficiency virus type 1 RNA, Virology 213 (2) (1995) 373–385.
[18] B. Palsson, The challenges of in silico biology, Natur. Biotechnol. 18 (2000) 1147–1150.
[19] D. Purcell, M. Martin, Alternative splicing of human immunodeficiency virus type 1 mRNA modulates viral protein expression, replication, and infectivity, J. Virol. 67 (11) (1993) 6365–6378.
[20] D. Ropers, Etude expérimentale du rôle des protéines SR dans la régulation de l'épissage de l'ARN du virus HIV-1, responsable de l'immunodéficience humaine, et modélisation mathématique de ces régulations, Ph.D. Thesis, University Henri Poincaré, Nancy, France.
[21] D. Ropers, L. Ayadi, S. Jacquenet, A. Méreau, D. Thomas, A. Mougin, P. Bilodeau, M. Stolzfus, R. Gattoni, J. Stévenin, C. Branlant, Differential effects of the SR proteins 9G8, SC35, ASF/SF2 and SRp40 on the utilization of the A1 to A5 splicing sites of HIV-1 RNA, J. Biol. Chem., in press.
[22] V.A. Saraswat, Concurrent Constraint Programming, MIT Press, Cambridge, MA, 1993.
[23] V.A. Saraswat, R. Jagadeesan, V. Gupta, Foundations of timed concurrent constraint programming, in: The Ninth Symp. Logic in Computer Science, LICS'94, Paris, IEEE, New York, 1994, pp. 71–80.
[24] V.A. Saraswat, R. Jagadeesan, V. Gupta, Timed default concurrent constraint programming, J. Symbol. Comput. 22 (5/6) (1996) 475–520.
[25] H. Tang, K. Kuhen, F. Wong-Staal, Lentivirus replication and regulation, Annual Rev. Genet. 33 (1999) 133–170.