Ralph-Johan Back | Ion Petre (Eds.)

Proceedings of COMPMOD 2008

Workshop on Computational Models for Cell Processes

Turku Centre *for* Computer Science

TUCS General Publication

No 47, May 2008

Proceedings of COMPMOD 2008

Workshop on Computational Models for Cell Processes

May 27, 2008, Turku, Finland

Editors:

Ralph-Johan Back
Ion Petre

TUCS General Publication

No 47, May 2008

Preface

This is the proceedings of the Workshop on "Computational Models for Cell Processes", organized in Turku, Finland, on May 27, 2008. The workshop is a satellite event of the "Formal Methods 2008" conference organized in Turku. The goal of the workshop is to bring together researchers in computer science (especially in formal methods) and mathematics (both discrete and continuous mathematics), interested in the opportunities and the challenges of systems biology. The program consists of three invited lectures by Professor Monika Heiner (Brandenburg University of Technology), Professor Jane Hillston (University of Edinburgh), and Dr. Russ Harmer (University Paris-Diderot), as well as five contributed papers. The scientific program of the workshop spans an interesting mix of approaches to systems biology, ranging from quantitative to qualitative techniques, from continuous to discrete mathematics, and from deterministic to stochastic methods. The contributed papers were peer-reviewed by a program committee consisting of Ralph-Johan Back (Åbo Akademi), Igor Goryanin (University of Edinburgh), Ion Petre (Åbo Akademi), Gordon Plotkin (University of Edinburgh), Corrado Priami (Microsoft Research - University of Trento, Centre for Computational and Systems Biology), and Grzegorz Rozenberg (University of Leiden). We thank them all for helping selecting such an interesting scientific program. We also thank Turku Centre for Computer Science for publishing these proceedings in their general publication series.

Turku, May 7, 2008 Ralph-Johan Back and Ion Petre

Contents

Part I

Invited talks

Petri Nets for Systems and Synthetic Biology

Monika Heiner

Brandeburg University of Technology

monika.heiner@gmx.de

Abstract

This talk describes a Petri net-based framework for modelling and analysing biochemical pathways, which unifies the qualitative, stochastic and continuous paradigms. Each perspective adds its contribution to the understanding of the system, thus the three approaches do not compete, but complement each other. A signal transduction pathway is used as running example. Consequently the focus is on transient behaviour analysis, and specifically on model checking by discussing related properties in the qualitative, stochastic and continuous paradigms. Although the framework is based on Petri nets, it can be applied more widely to other formalisms which are used to model and analyse biochemical networks.

This is joined work with David Gilbert and Robin Donaldson.

Bio-PEPA: A Formal Method for Integrated Systems Biology Modelling

Jane Hillston

University of Edinburgh

jeh@inf.ed.ac.uk

Abstract

PEPA is a stochastic process algebra which was introduced in the early 1990s for modelling computer and communication systems. More recently there has been some interest in applying PEPA, and other stochastic process algebras, to modelling intracellular networks. However there are some fundamental differences between biochemical pathways and computer systems. These have been the main motivators for Bio-PEPA, a new language tailored to modelling biochemical reaction pathways. In this talk I will present the Bio-PEPA formalism and the analysis techniques which it supports.

This is joint work with Federica Ciocchetta.

Rule-based modelling of cellular signalling

Russ Harmer

University Paris-Diderot

russ.harmer@gmail.com

Abstract

During its progression through the cell cycle, a cell must continually make choices based primarily on its external environment. For example, growth arrest —quiescence—can arise if the cell considers its immediate vicinity to be overcrowded; or in the absence of sufficient nutrients. In order to make such decisions, cells must link specialized transmembrane receptor proteins, that sample external conditions, to transcriptional (and other) regulation via what we call intracellular signalling pathways/networks. These networks act as a form of computation that integrates incoming signals—representing presence or absence of, for example, growth, survival or death signals—and appropriately selects the cell's fate. The signalling system can thus be seen as a computational medium in its own right and it becomes valid to ask what kind of programming a cell can intrinsically engage in with these means. For, indeed, the means seem highly limited: much of signalling can be reduced to binding and unbinding of proteins accompanied by potential modification of one protein by another—such as phosphorylation or ubiquitination.

We present the kappa-calculus, a formal language of agents and rules, representing proteins and their interactions, which captures this simple, yet apparently highly expressive, computational paradigm. Rules in κ directly represent, rather than encode (as do ODEs), biological knowledge and can thus be seen as self-documenting and as units of discussion in their own right. Moreover, the construction of a model reduces to the writing of—or selection, from an existing database, of—the rules that describe the interactions of that system. A model can thus be more easily built in the first place and far more easily extended or modified. Moreover, rules make explicit not only the agents involved in an interaction but also their sites (representing their binding motifs or domains). This renders practical the modelling of phenomena such as point mutations, receptor antibodies or kinase inhibitors. We illustrate these points with a large example (roughly 300 rules) of a model of the ErbB signalling network.

Part II

Contributed papers

Bio-PEPA with SBML-like events

Federica Ciocchetta

Laboratory for Foundations of Computer Science,

The University of Edinburgh, Edinburgh EH9 3JZ, Scotland

fciocche@inf.ed.ac.uk

Abstract

In this work we present an extension of Bio-PEPA, a language recently defined for the modelling and analysis of biological systems, to handle *events*. Broadly speaking, events are constructs that represent changes in the system due to some trigger conditions. Some mappings from Bio-PEPA with events to analysis tools are reported. in order to test our approach, we present the translation of two biological models into Bio-PEPA with events.

1 Introduction

Computational models play an important role in systems biology. Indeed they help to study, analyze and predict the behaviour of biological systems. In recent years there have been some applications of process algebras for the analysis of biological systems [20, 18, 5, 6]. In most cases the analysis is performed using Gillespie's stochastic simulation algorithm [11]. Other possibilities exist, such as the mapping to differential equations [4].

Many biological models need to capture both discrete and continuous phenomena [1, 2, 16]. These models are called *hybrid systems*. A first example of hybrid system involves the activation of a certain activity when the concentration of enabling quantities is above the desired threshold. A second example considers a signal or stimuli that becomes null after some time leading to some changes in the interactions of the system.

In this work we present an extension of Bio-PEPA [6, 7], a language recently defined for the modelling and analysis of biological systems, to handle *events*. Broadly speaking, events are constructs that represent changes in the system due to some trigger conditions. Here we are interested in simple forms of events. Specifically we refer to the definition of events reported in the SBML specification [15]. These kinds of events can be found in biochemical networks, such as the ones in BioModels database [17] or defined in some experimental settings. Indeed, in order to model some experiments, it may be necessary to render the possible change to the system, due, for instance, to the introduction of some reagents or the interruption of some external stimuli.

The idea underlying our work is the following:

Biological models with events $\Longrightarrow$ *Bio-PEPA with events* $\Longrightarrow$ *Analysis*

A first challenge concerns the modelling: we need to add events to the Bio-PEPA system. A second aspect is the analysis. Some maps must be defined from Bio-PEPA to analysis tools. Specifically we map our language to Hybrid Automata (HA) [12]. Furthermore, we can consider modifications of Gillespie's algorithm [11] or ODEs in order to tackle events. Events are added to our language as a set of elements and the rest of the syntax is unchanged. There are two motivations for this choice: first of all we keep the specification of the model as simple as possible, secondly this approach is appropriate when we study the same biochemical system but with different experimental regimes. Indeed we can modify the list of events without any changes to the rest of the system.

The use of mathematical formalisms in order to represents discrete changes in biological systems is not new [3, 1, 2, 16]. In [3] the authors present a map from stochastic Concurrent Constraint Programming (sCCP) to HA. The HA generated in this way are said to be able to capture some aspects of the dynamics which are lost if standard differential equations are used instead. In [1] the authors proposed a hybrid system approach to modelling an intra-cellular network using continuous differential equations to model some part of the system and mode-switching to describe the changes in the underlying dynamics. The authors of [16] discuss the use of discrete changes in biological systems and present some examples by using the formalism HybridSAL [14]. Finally, hybrid Concurrent Constraint Programming is used to model some biological systems with both discrete and continuous changes in [2]. In none of these works are SBML-like events considered explicitly, but the focus is on general hybrid systems.

An approach to model events similar to the ones considered in this paper has been proposed in the *Beta Workbench (BetaWB)* [8] and in the associated programming language *BlenX* [21]. In both the cases the analysis is limited to the stochastic simulation by Gillespie. The BetaWB is a tool for modelling and simulating biological processes, based on Beta-binders, a recently introduced process algebra suitable for the biological applicative domain. The language allows us to represent some specific cases of events. Events can be considered as global rules of the environment, triggered only when the conditions associated with them are satisfied. Each event is the composition of a condition and an action verb. The possible actions are the join of two entities, the split of one entity into two, the delete and the creation of a new entity. Each event is associated with a rate.

In BlenX more general events can be represented. As in the BetaWB, a single event is the composition of a condition and of an action, but the conditions can involve also the simulation time and the step size, in addition to the number of entities. Specifically, conditions are used to trigger the execution of an event when some elements are presents in the system, when a particular condition is met, with a given rate or at a precise simulation time or simulation step.

The rest of the paper is organised as follows. Section 2 reports a description of Bio-PEPA. In Section 3 we extend Bio-PEPA with SBML-like events and we discuss the possible kinds of analysis that can be performed from it. After that, Section 4 illustrates the modelling in Bio-PEPA of a biochemical network with an event. Finally, in Section 5, some conclusions are reported.

2 Bio-PEPA

Bio-PEPA [6, 7] is a new language for the modelling and analysis of biochemical networks. It may be seen as an extension of the reagent-centric view in PEPA [5]. In both cases we have the abstraction "processes as species": each sequential component represents a species (and not a single molecule as in other process algebras) and it is

parametric in terms of concentration levels. In particular the granularity of the system is expressed by a step size h, equal for all the species. A main feature of Bio-PEPA with respect to PEPA is the possibility to represent stochiometry in an explicit way and to consider kinetic laws different from mass-action. These laws are expressed by using functional rates. For details see [6, 7].

The syntax of Bio-PEPA is designed in order to collect the biological information we need:

$$S ::= (\alpha, \kappa)\ \mathsf{op}\ S \mid S + S \mid C \qquad P ::= P \underset{\mathcal{L}}{\bowtie} P \mid S(l)$$

where $\mathsf{op} = \downarrow \mid \uparrow \mid \oplus \mid \ominus \mid \odot$.

The component S is called a *sequential component* (or *species component*) and represents the species whereas the component P, called a *model component*, describes the system and the interactions among components. The parameter $l \in \mathbb{N}$ represents the discrete level of concentration. The prefix term $(\alpha, \kappa)\ \mathsf{op}\ S$ contains information about the role of the species in the reaction associated with the action type α: κ is the *stoichiometry coefficient* of the species and the *prefix combinator* "op" represents the role of the element in the reaction. Specifically, $\downarrow$ indicates a *reactant*, $\uparrow$ a *product*, $\oplus$ an *activator*, $\ominus$ an *inhibitor* and $\odot$ a generic *modifier*. The operator "+" expresses choice between possible actions and the constant C is defined by a equation $C \stackrel{def}{=} S$. Finally, the process $P \underset{\mathcal{L}}{\bowtie} Q$ denotes the cooperation between components: the set $\mathcal{L}$ determines those activities on which the operands are forced to synchronize.

In order to fully describe a biochemical network in Bio-PEPA we need to define structures that collect information about the compartments, the maximum concentrations, number of levels for all the species, the constant parameters and the functional rates. We can define the Bio-PEPA system in the following way:

Definition 1. *A Bio-PEPA system $\mathcal{P}$ is a 6-nuple $\langle \mathcal{V}, \mathcal{N}, \mathcal{K}, \mathcal{F}_R, Comp, P \rangle$, where: $\mathcal{V}$ is the set of compartments, $\mathcal{N}$ is the set of quantities describing each species, $\mathcal{K}$ is the set of parameter definitions, $\mathcal{F}_R$ is the set of functional rate definitions, $Comp$ is the set of definitions of sequential components, P is the model component describing the system.*

The behaviour of the system is defined in terms of an operational semantics. The rules are reported in [7]. We defined two relations over the processes. The former, called the *capability relation*, supports the derivation of quantitative information and it is auxiliary to the latter which is called the *stochastic relation*. The stochastic relation gives us the rates associated with each action. The rates are obtained by evaluating the functional rates associated with the action, divided by the step size. This rate represents the parameter of a negative exponential distribution. The dynamic behaviour of processes is determined by a *race condition*: all enabled activities attempt to proceed but only the fastest succeeds.

We have the following correspondences between a biochemical network and a Bio-PEPA system: each species i in the network is described by a species component C_i, each reaction j is associated with an action type α_j and its dynamics is described by a specific function $f_{\alpha_j} \in \mathcal{F}_R$.

A *Stochastic Labelled Transition System* can be defined for a Bio-PEPA system. It is worth noting that Bio-PEPA can be seen as an *intermediate, formal, compositional* representation of biological systems, from which different kinds of analysis can be performed. We have defined some mappings from Bio-PEPA to ODEs, CTMC, Gillespie's model and PRISM [19].

3 Bio-PEPA with events

3.1 Events and assumptions

In this work we limit our attention to events as defined in the SBML specification [15]. SBML events describe explicit discontinuous state changes in the model. Specifically, an SBML event has the following structure:

"*event_id*, if *trigger* then *event_assignment_list* with *delay*"

where *event_id* is the event identifier, *trigger* is a mathematical expression that, when it is evaluated to true, makes the event fire, *delay* is the length of time between when the event fires and when the event assignments are executed, *event_assignment_list* is a list of assignments that are made when the event is executed. The trigger and the list of assignments can involve parameters, species concentrations and compartment sizes.

We make the following assumptions for the events considered in this work.

1. Triggers can involve time and species (together or one of them), while assignments can involve constants, parameters, species, functional rates [1];
2. The events are all immediate and the transitions are deterministic;
3. The triggers are only unidirectional [2];
4. The events are sequential and compatible with each other.

These assumptions are not restrictive. Indeed these events allow us to represent a large number of discontinuous changes that we can find in biological models.

3.2 The definition of the language

Generally speaking we can add events to the Bio-PEPA model by introducing a *set* of elements that have the form (*id*, *trigger*, *event_assignment*, *delay*), where *id* is the name of the event, *trigger* is a mathematical expression involving the components of Bio-PEPA model and time, *event_assignment* is a list of assignments, *delay* is a real positive number or 0 [3](i.e. immediate events). Formally, we have the following definitions:

trigger ::= cond | cond **or** cond | cond **and** cond | **not** cond;
cond ::= t eq value | $exp(\overline{C}, \overline{k})$ eq value | $exp(\overline{C}, \overline{k})$ eq $exp(\overline{C}, \overline{k})$
eq ::= $=$ | $\neq$ | $\rangle$ | $\langle$ | $\leq$ | $\geq$ delay ::= value
event_assignment ::= assignment ; event_assignment
assignment ::= k $\leftarrow$ value | level(C) $\leftarrow$ value | $f_\alpha \leftarrow exp(\overline{C}, \overline{k})$
event ::= (id, trigger, event_assignment, delay)

where C stands for any sequential component and k for any parameter, the variable $t \in \mathbb{R}^+$ represents the global time of the system, $exp(\overline{C}, \overline{k})$ is an arithmetic expression involving a set of components (denoted $\overline{C}$) and a set of parameters (denoted $\overline{k}$), $value \in \mathbb{R}^+$ and *id* is a string indicating the event name. The function $level(C)$ associates a level with the component C. When we need the original value for the concentration, we write $C = l_C\{value_C\}$, where $value_C$ is the value of the concentration and l_C is the associated level. The set of events is then defined as:

[1] We do not consider events based on volume size, since in Bio-PEPA compartment are assumed constant and static.

[2] Bidirectional triggers can be decomposed into two unidirectional triggers.

[3] In the present work we consider only immediate actions, but generally we could have non-immediate actions.

$$Events ::= [] \mid event::Events$$

Definition 2. *A Bio-PEPA system with events $\mathcal{P}_E$ is a 8-nuple $\langle \mathcal{V}, \mathcal{N}, \mathcal{K}, \mathcal{F}, Comp, P, Events, t \rangle$, where Events is the set of events, $t \in \mathbb{R}^+$ is the variable expressing time and the other elements are as in the standard Bio-PEPA.*

A Bio-PEPA system is well-defined if all the elements are well-defined. The definition of well-definedness for all the elements, with the exception of events, is reported in [7]. Given a Bio-PEPA system with events, the set of events *Events* is well-defined if and only if: 1) triggers involve time or species, assignments involve species, parameters and compartments, 2) all the elements used in the events are defined in the Bio-PEPA system, for each event assignment, 3) the different assignments are independent (i.e. involve different elements). In the following we refer to Bio-PEPA with events simply as Bio-PEPA. Only well-defined Bio-PEPA systems are considered.

3.3 Analysis

In this section we discuss some maps from Bio-PEPA to analysis tools.

Hybrid Automata Hybrid automata [12] combine discrete transition graphs with continuous dynamical systems. They are used to formally model hybrid systems, dynamical systems with both discrete and continuous components. An hybrid automaton consists of a finite set of *real-values variables* $\{X_1, X_2, ..., X_n\}$ and a finite labelled graph, whose vertices correspond to *control modes* (states), described by differential equations, and whose edges are *control switches*, corresponding to discrete events. In addition, we have some labels for the edges, specifying the *jump conditions* (activation conditions) and labels for the vertices, containing information about initial and invariant conditions. The variables evolve continuously in time, apart from some changes induced by events. When an event happens there is a change in the mode. The dynamic behaviour of each mode is described by a set of differential equations, generally different from mode to mode. We can use HA both for simulation (see for instance *the SHIFT language* [9]) and model checking (see *HyTech* [13]). For a formal definition and details about the formalism see [12].

Here we present briefly the map from Bio-PEPA to HA. Let $\mathcal{P}_0$ be the initial Bio-PEPA system. We have the following correspondences:

1. Each species component C_i in $Comp$ is associated with a variable X_i. The set of variables is then given by: $\{X_1, X_2, ..., X_{N_{Comp}}, t\}$, where t is the variable expressing the time (described by the trivial differential equation $dt/dt = 1$) and N_{Comp} is the number of species components.

2. The initial conditions of the variables are derived from $Comp$. The variable t is initially set to 0.

3. For each event $i \in Events$, we can consider the trigger tr_i. We use these triggers to define the jump conditions. In the case of sequential events, the number of possible jump conditions is N_{Events} (the number of events in the system). Note that if we consider non-sequential events, we have a number of triggers greater than N_{Events}. Indeed for each set of triggers that can happen simultaneously, we have to define jump conditions to represent one trigger at a time and all the possible combinations.

4. Each mode is described by a specific instance of the Bio-PEPA system. Indeed modes are defined according to either the initial system or the system modified with the event assignments relative to a trigger. If just sequential components are assumed, the number of modes is $N_{Events} + 1$. We indicate the modes with σ and the set of modes Σ. In each mode some invariant conditions are added in order to force the change of mode when the trigger becomes true. We have that:

   - The initial mode σ_0 is defined from the initial system $\mathcal{P}_0$. It is described in terms of an ODE system and this is derived from the Bio-PEPA model by considering the map π_{ODE} in [7]. Therefore we have $\sigma_0 = \pi_{ODE}(\mathcal{P}_0)$.
   - Given a mode $\sigma_i = \pi_{ODE}(\mathcal{P}_i)$, let tr_{ij} be one possible jump condition that can be satisfied from it. We define the Bio-PEPA system $\mathcal{P}_j = \mathcal{P}_i[event\_assignment_{ij}]$ as the reset of the previous system $\mathcal{P}_i$ according to the event assignments associated with the trigger. The mode σ_j is then defined as $\sigma_j = \pi_{ODE}(\mathcal{P}_j)$.

Gillespie and ODE analysis These algorithms have to be modified in order to consider events. These are tackled by adding some conditions and some checks along the simulation. We start at time $t = 0$, with the Bio-PEPA system at the initial conditions. We assume that initially all the triggers are evaluated to false. When one of the conditions is satisfied, the simulation stops and the system is reset according to the event assignments associated with the trigger. After that, the simulation can start again until another condition becomes true or the simulation time is reached. According to our assumptions, triggers are compatible with each other and are all different.

For both deterministic and stochastic simulations we propose the following procedure.

1. Let $\mathcal{P}_0$ be the initial Bio-PEPA system and $time\_S$ the maximum simulation time. Let N_{Events} be the number of events in the system.
2. While $t\langle time\_S$ and $trigger_i = false$ for $i = 1, 2, ..., N_{Events}$, simulate.
3. If $t \geq time\_S$ then stop.
4. If $t\langle time\_S$ and there exists a $trigger_i$ such that it is true, reset the Bio-PEPA system according to the event assignments associated with that trigger: $\mathcal{P}'(t) = \mathcal{P}(t)[event\_assignment_i]$. Go to (2).

3.4 A simple example involving concentrations

A gene X activates the expression of gene Y; above a certain threshold, gene Y inhibits expression of gene X. The reactions describing this situation are:

- activation of Y: $X \xrightarrow{r_1} X + Y$, with $r_1 = 0.01$;
- degradation of X: $X \xrightarrow{r_2} \emptyset$ with $r_2 = 0.02$;
- creation of X: $\emptyset \xrightarrow{r_3} X$ with $r_3 = 0.01$, possible when the concentration of Y is less than 0.8.

This simple model is translated into Bio-PEPA as [4]:

[4]Note that we use X and Y (capital letters) to indicate the names of the species and the name of the Bio-PEPA component, whereas x and y indicate the associate species concentration.

$$X \stackrel{def}{=} (\alpha_1, 1) \oplus X + (\alpha_2, 1)\downarrow X + (\alpha_3, 1)\uparrow X\,; \qquad Y \stackrel{def}{=} (\alpha_1, 1)\uparrow Y;$$
$$Res \stackrel{def}{=} (\alpha_2, 1) \odot Res\,; \qquad CF \stackrel{def}{=} (\alpha_3, 1) \odot CF;$$

$$((X(0) \underset{\alpha_1}{\bowtie} Y(0)) \underset{\alpha_3}{\bowtie} CF(1)) \underset{\alpha_2}{\bowtie} Deg(0)$$

with the addition of the set of events $[(event_1, Y = 1\{0.8\}, r_3 \leftarrow 0, 0)]$ and where Res and CF are two auxiliary components to represent the degradation and the synthesis, respectively. The initial values are zero for both the genes.

Analysis by means of ODEs is reported in Fig.1.

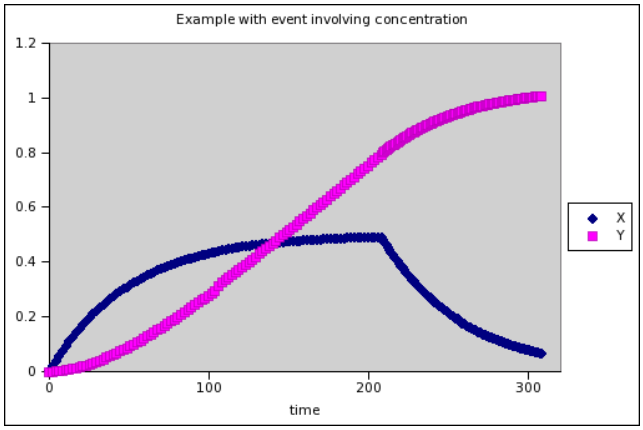

Figure 1: Simulation results for the example with a simple event involving concentration.

A description in terms of HA is reported in Fig.2. We have two modes, one describing the case $y\langle 0.8$ (initial mode) and the other the case $y \geq 0.8$. The systems in the two modes are the similar, but in the former case the reaction α_3 is activated, in the second case not. The guard to move from one mode to the other is "$y = 0.8$".

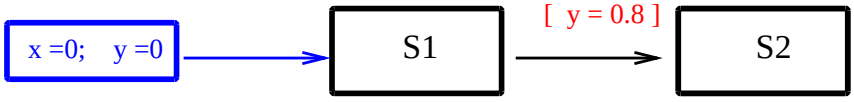

Figure 2: HA representation for the example involving concentration.

In the Figure 2 $S1$ and $S2$ represents the two ODE models representing the system when $y\langle 0.8$ and $y \geq 0.8$, respectively. The system $S1$ is:

$$\begin{cases} \frac{dx}{dt} &= -0.02 * x + 0.01; \\ \frac{dy}{dt} &= 0.01 * x \end{cases}$$

and the system $S2$ is:

$$\begin{cases} \frac{dx}{dt} &= -0.02 * x; \\ \frac{dy}{dt} &= 0.01 * x \end{cases}$$

The initial conditions are $x = 0,\ y = 0$.

4 The acetylcholine receptor model

This example concerns the functional properties of the *nicotin Acetylcholine Receptors*. These are transmembrane proteins that mediate inter-conversions between open and

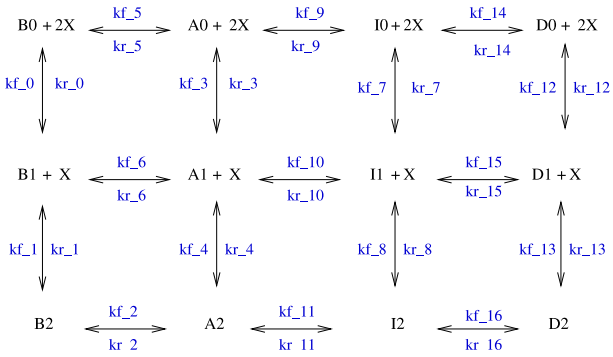

Figure 3: Schema of the acetylcholine receptor model.

closed channel states under the control of neurotransmitters. The detailed description of the model is reported in [10].

A schema of the model is shown in Figure 3. B (Basal state), A (Active state), D (Desensitized state) and I (Inactivable state) represent the different states of the acetylcholine receptors. The numbers 0, 1, 2 associated with the state represent the number of ligands (denoted X) bound to a receptor. Each column corresponds to a series of ligand binding actions at two identical sites per receptor whereas each row corresponds to a series of transactions between conformational states. All the reactions are reversible and the dynamics are described by mass-action laws. For each reaction i, with $i = 1, 2, ...16$, the rate of the forward direction is kf_i and the rate of the reverse reaction kr_i.

In addition, there is an event to describe the recovery upon removal of free agonist at a given time. This is expressed by constraining the reaction rates of each second-order ligand-receptor reaction to zero. These constraints prevent ligand binding reactions from happening after that time, hence the states evolve as if the free ligands were completely removed from the system. The event is immediate, the trigger is "$t = t_2$", where $t_2 = 20\ s$, and the event assignments are $kf_0 = 0$, $kf_1 = 0$, $kf_3 = 0$, $kf_4 = 0$, $kf_7 = 0$, $kf_8 = 0$, $kf_{12} = 0$, $kf_{13} = 0$.

The Bio-PEPA system associated with the Edelstein's model

- *Definition of the compartment list $\mathcal{V}$.* In the model we have a single three-dimensional compartment, defined as "$comp1$: $1e$-16, l;", where l is litre.

- *Definition of the set $\mathcal{N}$.* Each species is associated with a species component. For each species component we have to declare the step-size, the number of levels, the initial and maximum concentrations and the compartment where the species is. The ligand is not represented explicitly. In the case of $B0$ we have: $B0{:}h, N_{B0}, M_0, M_{B0}, comp1, \mu M$, where the step size h is $1.66e$-5, the number of levels N_{B0} is 2, the initial concentration M_0 is $1.66e$-$5\mu M$ (M is molar, that is $moles/l$) and coincides with the maximum concentration M_{B0}.

- *Definition of functional rates ($\mathcal{F}_R$) and parameters ($\mathcal{K}$).* Each reversible reaction in the model is decomposed in two irreversible reactions. For each reaction i with $i = 0, ..., 16$ we have the kinetic laws $f_{\alpha\_f_i} = fMA(kf\_i)$; and $f_{\alpha\_r_i} = fMA(kr\_i)$, where fMA stands for mass-action. All the parameters are defined in the set $\mathcal{K}$. The values are the ones reported in the paper.

- *Definition of species components (Comp) and of the model component (P).*

  In the following we report the definition for $B0$, $B1$ and $B2$; the other species are dealt with similarly.

$$
\begin{array}{lll}
B0 & \stackrel{def}{=} & (\alpha\_f_0, 1)\downarrow B0 + (\alpha\_r_0, 1)\uparrow B0 + (\alpha\_f_5, 1)\downarrow B0 + (\alpha\_r_5, 1)\uparrow B0; \\
B1 & \stackrel{def}{=} & (\alpha\_f_0, 1)\uparrow B1 + (\alpha\_r_0, 1)\downarrow B1 + (\alpha\_f_6, 1)\downarrow B1 + (\alpha\_r_6, 1)\uparrow B1 + \\
 & & (\alpha\_f_1, 1)\uparrow B1 + (\alpha\_r_1, 1)\downarrow B1; \\
B2 & \stackrel{def}{=} & (\alpha\_f_2, 1)\downarrow B2 + (\alpha\_r_2, 1)\uparrow B2 + +(\alpha\_f_1, 1)\uparrow B2 + (\alpha\_r_1, 1)\downarrow B2;
\end{array}
$$

  The system is described as:

$$
\begin{array}{c}
B0(1) \underset{L1}{\bowtie} B1(0) \underset{L2}{\bowtie} B2(0) \underset{L3}{\bowtie} A0(0) \underset{L3}{\bowtie} A1(0) \underset{L4}{\bowtie} A2(0) \underset{L5}{\bowtie} \\
I0(0) \underset{L6}{\bowtie} I1(0) \underset{L7}{\bowtie} I2(0) \underset{L8}{\bowtie} D0(0) \underset{L9}{\bowtie} D1(0) \underset{L10}{\bowtie} D2(0)
\end{array}
$$

  where L_i, $i = 1, ..., 10$ are the cooperation sets.

- *Definition of events*. We have only one event: $[(event_1, t = 20,\ kf_0 = 0\ ;\ kf_1 = 0\ ;\ kf_3 = 0\ kf_4 = 0\ ;\ kf_7 = 0\ ;\ kf_8 = 0\ ;\ kf_{12} = 0\ ;\ kf_{13} = 0, 0)]$

Some simulation results are reported in Fig. 4. The simulations are made by using Gillespie's algorithm. The initial number of molecules for $B0$ is given $M_0 \times V \times Na = (1.66e\text{-}5\ \mu M) \times (1.e\text{-}16\ l) \times (6.022 \times e{+}23\ (moles)^{-1}) = 1000$, where Na is the Avogadro number[5]. All the other species are initially null. The graph reproduces results in agreement with the ones reported in the paper [10]. Following the ligand removal, the state $I2$ loses agonist molecules and is transformed to the state $B0$ very rapidly, while $D2$ loses ligand molecules to form $D0$. Since the data occur on a wide range of times we represent the time on a logarithmic scale.

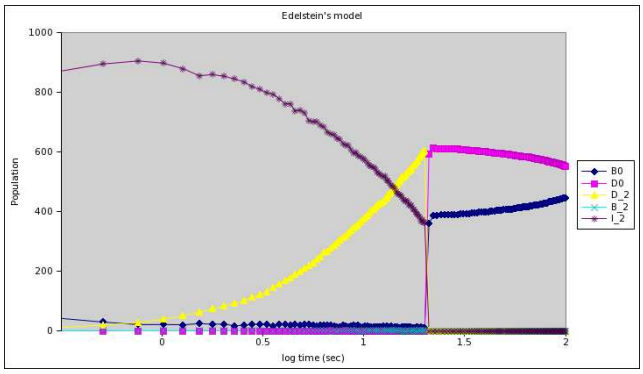

Figure 4: Simulation results for Edelstein's model (time in log scale).

Concerning the translation into HA, the result is similar to the one reported in Section 3.4 for the simple example. Also in this case we have two modes, described by two different sets of differential equations. The trigger condition involves time and it is "$t = 20\ s$". The details are not reported.

[5]It is the number of "entities" (atoms or molecules) in one mole of substance.

5 Conclusions

In this work we present an extension of Bio-PEPA to handle *events*. Events are constructs that represent changes in the system due to some trigger conditions. The events considered here are simple, but nevertheless able to describe most of the discontinuous changes in models and experiments. Events are added to our language without any modification to the rest of the syntax. The motivation of this choice is that we want to keep the specification of the model as simple as possible.

A topic for the future concerns the study of more general events (for instance, non-immediate or simultaneous events) and the possible extension to other kinds of hybrid systems in biology. Furthermore we plan to exploit the possible kinds of analysis involving hybrid systems in the context of systems biology. The implementation of the mappings from Bio-PEPA to the analysis tools is under development.

Acknowledgements

The author thanks Jane Hillston for her helpful comments. The author is supported by the EPSRC under the CODA project "Process Algebra Approaches for Collective Dynamics" (EP/c54370x/01 and ARF EP/c543696/01).

References

[1] R. Alur, C. Belta, F. Ivancic, V. Kumar, M. Mintz, G. Pappa, H. Rubin and J. Schug. Hybrid modeling and simulation of biomolecular networks. In Proc. of *4th International Workshop on Hybrid Systems: Computation and Control*, volume LNCS 2034, pages 19-32, 2001.

[2] A. Bockmayr and A. Courtois. Using hybrid concurrent constraint programming to model dynamic biological systems. In Proc. of the 18th *International Conference on Logic Programming*, volume 2401, Springer-Verlag, 2002.

[3] L. Bortolussi and A. Policriti. Hybrid Approximation of Stochastic Concurrent Constraint Programming, Proc. of *PASTA 2007*, 2007.

[4] M. Calder, S. Gilmore and J. Hillston. Automatically deriving ODEs from process algebra models of signalling pathways, Proc. of *CMSB'05*, pages 204–215, 2005.

[5] M. Calder, S. Gilmore, and J. Hillston. Modelling the influence of RKIP on the ERK signalling pathway using the stochastic process algebra PEPA. *T. Comp. Sys. Biology*, VII, volume 4230 of LNCS, pages 1–23, Springer, 2006.

[6] F. Ciocchetta, and J. Hillston. Bio-PEPA: an extension of the process algebra PEPA for biochemical networks. Proc. of *FBTC 2007*, volume 194/3 of ENTCS, pages 103–117, 2008.

[7] F. Ciocchetta, and J. Hillston. Bio-PEPA: a framework for the modelling and analysis of biological systems. Technical report EDI-INF-RR-1231, University of Edinburgh, 2008.

[8] L. Dematté, C. Priami and A. Romanel. The BlenX Language: a tutorial. Chapter for the tutorial of SFM-08:Bio, LNCS, volume 5016, 2008.

[9] A. Deshpande, A. Gll, L. Semenzato. SHIFT Programming Language and Run-Time System for Dynamic Networks of Hybrid Automata. PATH Report, available at http://path.berkeley.edu/SHIFT/publications.html.

[10] S.J. Edelstein, O. Schaad, E. Henry, D. Bertrand and J.P. Changgeux. A kinetic mechanism for nicotin acetylcholine receptors based on multiple allosteric transitions. *Biol. Cybern.*, 75, pages 361–379, 1996.

[11] D.T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *Journal of Physical Chemistry*, volume 81, pages 2340–2361, 1977.

[12] T.A. Henzinger. The Theory of Hybrid Automata. In the proceedings of the 11th Annual IEEE Symposium on Logic in Computer Science, LICS'96, 1996.

[13] T.A. Henzinger, P.-H. Ho and H. Wong-Toi. HyTech: A Model Checker for Hybrid Systems. *Software Tools for Technology Transfer*, volume 1, pages 110–122, 1997.

[14] HybridSal home page, http://sal.csl.sri.com/hybridsal/.

[15] M. Hucka, A. Finney, S. Hoops, S. Keating and Le Novére. Systems Biology Markup Language (SBML) Level 2: Structures and Facilities for Model Definitions. Available at http://sbml.org/documents/.

[16] P. Lincoln and A. Tiwari. Symbolic systems biology: Hybrid modeling and analysis of biological networks. In proc. *Hybrid Systems: Computation and Control 7th Intl. Workshop*, LNCS 2993, pages 660–672, 2004.

[17] N. Le Novére, B. Bornstein, A. Broicher, M. Courtot, M. Donizelli, H. Dharuri, L. Li, H. Sauro, M. Schilstra, B. Shapiro, J.L. Snoep, and M. Hucka. BioModels Database: a Free, Centralized Database of Curated, Published, Quantitative Kinetic Models of Biochemical and Cellular Systems. *Nucleic Acids Research*, volume 34, pages D689–D691, 2006.

[18] C. Priami and P. Quaglia. Beta-binders for biological interactions. Proc. of *CMSB'04*, Volume 3082 of LNCS, pages 20–33, Springer, 2005.

[19] Prism web site. http://www.prismmodelchecker.org/

[20] C. Priami, A. Regev, W. Silverman and E. Shapiro. Application of a stochastic name-passing calculus to representation and simulation of molecular processes. *Information Processing Letters*, volume 80, pages 25–31, 2001.

[21] A. Romanel, L. Dematté and C. Priami. The Beta Workbench. *Technical report TR-03-2007, Microsoft Research-University of Trento Centre for Computational and Systems Biology.*

In Silico Modelling and Analysis of Ribosome Kinetics and aa-tRNA Competition

D. Bošnački* [1]
T.E. Pronk† [2]
E.P. de Vink‡ [3]

* Dept. of Biomedical Engineering, Eindhoven University of Technology

† Swammerdam Institute for Life Sciences, University of Amsterdam

‡ Dept. of Mathematics and Computer Science, Eindhoven University of Technology

Abstract

We present a formal analysis of ribosome kinetics using probabilistic model checking and the tool Prism. We compute different parameters of the model, like probabilities of translation errors and average insertion times per codon. The model predicts strong correlation to the quotient of the concentrations of the so-called cognate and near-cognate tRNAs, in accord with experimental findings and other studies. Using piecewise analysis of the model, we are able to give an analytical explanation of this observation.

1 Introduction

The translation mechanism that synthesizes proteins based on mRNA sequences is a fundamental process of the living cell. Conceptually, an mRNA can be seen as a string of codons, each coding for a specific amino acid. The codons of an mRNA are sequentially read by a ribosome, where each codon is translated using an amino acid specific transfer-RNA (aa-tRNA), building one-by-one a chain of amino acids, i.e. a protein. In this setting, aa-tRNA can be interpreted as molecules containing a so-called anticodon, and carrying a particular amino acid. Dependent on the pairing of the codon under translation with the anticodon of the aa-tRNA, plus the stochastic influences such as the changes in the conformation of the ribosome, an aa-tRNA, arriving by Brownian motion, docks into the ribosome and may succeed in adding its amino acid to the chain under construction. Alternatively, the aa-tRNA dissociates in an early or later stage of the translation.

Since the seventies a vast amount of research has been devoted, unraveling the mRNA translation mechanism and related issues. By now, the overall process of translation is reasonably well understood from a qualitative perspective. The translation

[1] Supported by FP6 LTR ESIGNET.
[2] Funded by the BSIK project Virtual Laboratory for e-Science VL-e.
[3] Corresponding author, e-mail evink@win.tue.nl.

process consists of around twenty small steps, a number of them being reversible. For the model organism *Escherichia coli*, the average frequencies of aa-tRNAs per cell have been collected, but regarding kinetics relatively little is known exactly. Over the past few years, Rodnina and collaborators have made good process in capturing the time rates for various steps in the translation process for a small number of specific codons and anticodons [14, 17, 18, 9]. Using various advanced techniques, they were able to show that the binding of codon and anticodon is crucial at a number of places for the time and probability for success of elongation. Based on these results, Viljoen and co-workers started from the assumption that the rates found by Rodnina et al. can be used in general, for all codon-anticodon pairs as estimates for the reaction dynamics. In [7], a complete detailed model is presented for all 64 codons and all 48 aa-tRNA classes for *E. coli*, on which extensive Monte Carlo experiments are conducted. In particular, using the model, codon insertion times and frequencies of erroneous elongations are established. Given the apparently strong correlation of the ratio of so-called near-cognates vs. cognate and pseudo-cognates, and near-cognates vs. cognates, respectively, it is argued that competition of aa-tRNAs, rather than their availability decides both speed and fidelity of codon translation.

In the present paper, we propose to exploit abstraction and model checking of continuous-time Markov chains (CTMCs) with Prism [13, 10]. The abstraction conveniently reduces the number of states and classes of aa-tRNA to consider. The tool provides built-in performance analysis algorithms and path chasing machinery, relieving its user from mathematical calculations. More importantly, from a methodological point of view, the incorporated CSL-logic [2] allows to establish quantitative results for parts of the system, e.g. for first-passage time for a specific state. Such piecewise analysis proves useful when explaining the relationships suggested by the data collected from the model. Additionally, in our case, the Prism tool enjoys rather favourably response times compared to simulation.

Related work The present investigation started from the Monte-Carlo experiments of mRNA translation reported in [7]. A similar stochastic model, but based on ordinary differential equations, was developed in [11]. It treats insertion times, but no translation errors. The model of mRNA translation in [8] assumes insertion rates that are directly proportional to the mRNA concentrations, but assigns the same probability of translation error to all codons.

Currently, there exist various applications of formal methods to biological systems. A selection of recent papers from model checking and process algebra includes [16, 4, 5]. More specifically pertaining to the current paper, [3] applies the Prism modelchecker to analyze stochastic models of signaling pathways. Their methodology is presented as a more efficient alternative to ordinary differential equations models, including properties that are not of probabilistic nature. Also [10] employs Prism on various types of biological pathways, showing how the advanced features of the tool can be exploited to tackle large models.

Organization of the paper Section 2 provides the biological background, discussing the mRNA translation mechanism. Its Prism model is introduced in Section 3. In Section 4, it is explained how error probabilities are obtained from the model and why they correlate with the near-cognate/cognate fraction. This involves adequate estimates of specific stochastic subbehaviour. Insertion times are the subject of Section 5. There too, it is illustrated how the quantitative information of parts of the systems is instrumental in deriving the relationship with the ratio of pseudo-cognate and near-cognates vs. cognates.[4]

Acknowledgments We are grateful to Timo Breit, Christiaan Henkel, Erik Luit,

[4] An appendix presents supplementary figures and data.

Jasen Markovski, and Hendrik Viljoen for fruitful discussions and constructive feedback.

2 A kinetic model of mRNA translation

In nature, there is a fixed correspondence of a codon and an amino acid. This is the well-known genetic code. Thus, an mRNA codes for a unique protein. However, the match of a codon and the anticodon of a tRNA is different from pair to pair. The binding influences the speed of the actual translation.[5] Here, we give a brief overview of the translation mechanism. Our explanation is based on [17, 12]. Two main phases can be distinguished: peptidyl transfer and translocation.

The peptidyl transfer phase runs through the following steps. aa-tRNA arrives at the A-site of the ribosome-mRNA complex by diffusion. The initial binding is relatively weak. Codon recognition comprises (i) establishing contact between the anticodon of the aa-tRNA and the current codon in the ribosome-mRNA complex, and (ii) subsequent conformational changes of the ribosome. *GTP*ase-activation of the elongation factor *EF-Tu* is largely favoured in case of a strong complementary matching of the codon and anticodon. After *GTP*-hydrolysis, producing inorganic phosphate P_i and *GDP*, the affinity of the ribosome for the aa-tRNA reduces. The subsequent accommodation step also depends on the fit of the aa-tRNA.

Next, the translocation phase follows. Another GTP-hydrolysis involving elongation factor *EF-G*, produces *GDP* and P_i and results in unlocking and movement of the aa-tRNA to the P-site of the ribosome. The latter step is preceded or followed by P_i-release. Reconformation of the ribosome and release of *EF-G* moves the tRNA, that has transferred its amino acid to the polypeptide chain, into the E-site of the ribosome. Further rotation eventually leads to dissociation of the used tRNA.

At present, there is little quantitative information regarding the translation mechanism. For *E. coli*, a number of specific rates have been collected [17, 9], whereas some steps are known to be relatively rapid. The fundamental assumption of [7], that we also adopt here, is that experimental data found by Rodnina et al. for the *UUU* and *CUC* codons, extrapolate to other codons as well. However, further assumptions are necessary to fill the overall picture. In particular, Viljoen proposes to estimate the delay due to so-called non-cognate aa-tRNA, that are blocking the ribosomal A-site, as 0.5ms. Also, accurate rates for the translocation phase are largely missing. Again following [7], we have chosen to assign, if necessary, high rates to steps for which data is lacking. This way these steps will not be rate limiting.

3 The Prism model

The abstraction of the biological model as sketched in the previous section is twofold: (i) Instead of dealing with 48 classes of aa-tRNA, that are identified by the their anticodons, we use four types of aa-tRNA distinguished by their matching with the codon under translation. (ii) We combine various detailed steps into one transition. The first reduction greatly simplifies the model, more clearly eliciting the essentials of the underlying process. The second abstraction is more a matter of convenience, though it helps in compactly presenting the model.

For a specific codon, we distinguish four types of aa-tRNA: cognate, pseudo-cognate, near-cognate, non-cognate. Cognate aa-tRNAs have an anticodon that strongly couples with the codon. The amino acid carried by the aa-tRNA is always the right

[5] See Figure 2 and Figure 3 in the appendix.

one, according to the genetic code. The binding of the anticodon of a pseudo-cognate aa-tRNA or a near-cognate aa-tRNA is weaker, but sufficiently strong to occasionally result in the addition of the amino acid to the nascent protein. In case the amino acid of the aa-tRNA is, accidentally, the right one for the codon, we call the aa-tRNA of the pseudo-cognate type. If the amino acid does not coincide with the amino acid the codon codes for, we speak in such a case of a near-cognate aa-tRNA.[6] The match of the codon and the anticodon can be very poor too. We refer to such aa-tRNA as being non-cognate for the codon. This type of aa-tRNA does not initiate a translation step at the ribosome.

The Prism model can be interpreted as the superposition of four stochastic automata, each encoding the interaction of one of the types of aa-tRNA. The automata for the cognates, pseudo-cognates and near-cognates are very similar; the cognate type automaton only differs in its value of the rates from those for pseudo-cognates and near-cognates, while the automata for pseudo-cognates and for near-cognates only differ in their arrival process. The automaton for non-cognates is rather simple.

Below, we are considering average transition times and probabilities for reachability based on exponential distributions. Therefore, following common practice in performance analysis, there is no obstacle to merge two subsequent sequential transitions with rates λ and μ, say, into a combined transition of rate $\lambda\mu/(\lambda+\mu)$. This way, an equivalent but smaller model can be obtained. However, it is noted, that in general, such a simplification is not compositional and should be taken with care.

For the modeling of continuous-time Markov chains, Prism commands have the form `[label] guard` $\rightarrow$ `rate : update ;`. In short, from the commands whose guards are fulfilled in the current state, one command is selected proportional to its relative rate. Subsequently, the update is performed on the state variables. So, a probabilistic choice is made among commands. Executing the selected command results in a progress of time according to the exponential distribution for the particular rate. We refer to [13, 10] for a proper introduction to the Prism modelchecker.

Initially, control resides in the common start state `s=1` of the Prism model with four boolean variables `cogn`, `pseu`, `near` and `nonc` set to false. Next, an arrival process selects one of the booleans that is to be set to true. This is the initial binding of the aa-tRNA. The continuation depends on the type of aa-tRNA: cognate, pseudo-cognate, near-cognate or non-cognate. In fact, a race is run that depends on the concentrations `c_cogn`, `c_pseu`, `c_near` and `c_nonc` of the four types of aa-tRNA and a kinetic constant `k1f`. Following Markovian semantics, the probability in the race for `cogn` to be set to true (the others remaining false) is the relative concentration `c_cogn`/(`c_cogn` + `c_pseu` + `c_near` + `c_nonc`).

```
// initial binding
[ ] (s=1) -> k1f * c_cogn : (s'=2) & (cogn'=true) ;
[ ] (s=1) -> k1f * c_pseu : (s'=2) & (pseu'=true) ;
[ ] (s=1) -> k1f * c_near : (s'=2) & (near'=true) ;
[ ] (s=1) -> k1f * c_nonc : (s'=2) & (nonc'=true) ;
```

As the aa-tRNA, that is just arrived, may dissociate too, the reversed reaction is in the model as well. However, control does not return to the initial state directly, but, for modelchecking purposes, first to the state `s=0` representing dissociation. At the same time, the boolean that was true is reset. Here, cognates, pseudo-cognates and near-cognates are handled with the same rate `k2b`. Non-cognates always dissociate as captured by the separate rate `k2bx`.

```
// dissociation
```

[6] The notion of a pseudo-cognate comes natural in our modeling. However, the distinction between a pseudo-cognate and a near-cognate is non-standard. Usually, a near-cognate refers to both type of tRNA.

```
[ ] (s=2) &  ( cogn | pseu | near ) -> k2b :
        (s'=0) & (cogn'=false) & (pseu'=false) & (near'=false) ;
[ ] (s=2) &  nonc -> k2bx : (s'=0) & (nonc'=false) ;
```

An aa-tRNA that is not a non-cognate can continue from state s=2 in the codon recognition phase, leading to state s=3. This is a reversible step in the translation mechanism, so there are transitions from state s=3 back to state s=2. However, the rates for cognates vs. pseudo- and near-cognates, viz. k3bc, k3bp and k3bn, differ significantly (see Table 1). Note that the values of the booleans do not change.

```
// codon recognition
[ ] (s=2) & ( cogn | pseu | near ) -> k2f : (s'=3) ;
[ ] (s=3) & cogn -> k3bc : (s'=2) ;
[ ] (s=3) & pseu -> k3bp : (s'=2) ;
[ ] (s=3) & near -> k3bn : (s'=2) ;
```

The next forward transition, from state s=3 to state s=4, is a combination of detailed steps involving the processing of GTP. The transition is one-directional, again with a significant difference in the rate k3fc for a cognate aa-tRNA and the rates k3fp and k3fn for pseudo-cognate and near-cognate aa-tRNA, that are equal.

```
// GTPase activation, GTP hydrolysis, EF-Tu conformation change
[ ] (s=3) & cogn -> k3fc : (s'=4) ;
[ ] (s=3) & pseu -> k3fp : (s'=4) ;
[ ] (s=3) & near -> k3fn : (s'=4) ;
```

In state s=4, the aa-tRNA can either be rejected, after which control moves to the state s=5, or accommodates, i.e. the ribosome reconforms such that the aa-tRNA can hand over the amino acid it carries, so-called peptidyl transfer. In the latter case, control moves to state s=6. As before, rates for cognates and those for pseudo-cognates and near-cognates are of different magnitudes.

```
// rejection
[ ] (s=4) & cogn -> k4rc : (s'=5) & (cogn'=false) ;
[ ] (s=4) & pseu -> k4rp : (s'=5) & (pseu'=false) ;
[ ] (s=4) & near -> k4rn : (s'=5) & (near'=false) ;
// accommodation, peptidyl transfer
[ ] (s=4) & cogn -> k4fc : (s'=6) ;
[ ] (s=4) & pseu -> k4fp : (s'=6) ;
[ ] (s=4) & near -> k4fn : (s'=6) ;
```

After a number of movements back-and-forth between state s=6 and state s=7, the binding of the EF-G complex becomes permanent. In the detailed translation mechanism a number of (mainly sequential) steps follows, that are summarized in the Prism model by a single transition to a final state s=8, that represents elongation of the protein in nascent with the amino acid carried by the aa-tRNA. The synthesis is successful if the aa-tRNA was either a cognate or pseudo-cognate for the codon under translation, reflected by either cogn or pseu being true. In case the aa-tRNA was a near-cognate (non-cognates never pass beyond state s=2), an amino acid that does not correspond to the codon in the genetic code has been inserted. In the later case, an insertion error has occurred.

```
// EF-G binding
[ ] (s=6) -> k6f : (s'=7) ;
[ ] (s=7) -> k7b : (s'=6) ;
// GTP hydrolysis, unlocking, tRNA movement and Pi release,
// rearrangements of ribosome and EF-G, dissociation of GDP
[ ] (s=7) -> k7f : (s'=8) ;
```

A number of transitions, linking the dissociation state s=0 and the rejection state s=5 back to the start state s=1, where a race of aa-tRNAs of the four types commences a new, and looping at the final state s=8, complete the Prism model.

```
// no entrance, re-entrance at state 1
[ ] (s=0) -> FAST : (s'=1) ;
// rejection, re-entrance at state 1
[ ] (s=5) -> FAST : (s'=1) ;
// elongation
[ ] (s=8) -> FAST : (s'=8) ;
```

Table 1 collects the rates as gathered from the biological literature [17, 7] and used in the Prism model above.

k1f	140	k3fc	260	k4rc	60	k6f	150
k2f	190	k3fp, k3fn	0.40	k4rp, k4rn	FAST	k7f	145.8
k2b	85	k3bc	0.23	k4fc	166.7	k7b	140
k2bx	2000	k3bp, k3bn	80	k4fp, k4fn	46.1		

Table 1: Rates of the Prism model.

In the next two sections, we will study the Prism model described above for the analysis of the probability for insertion errors, i.e. extension of the peptidyl chain with a different amino acid than the codon codes for, and of the average insertion times, i.e. the average time it takes to process a codon up to elongation.

4 Insertion errors

In this section we show how the model checking features of Prism can be used to predict the misreading frequencies for individual codons. The translation of mRNA into a polypeptide chain is performed by the ribosome machinery with high precision. Experimental measurements show that on average, only one in 10,000 amino acids is added wrongly.[7]

For a codon under translation, a pseudo-cognate anticodon carries precisely the amino acid that the codon codes for. Therefore, successful matching of a pseudo-cognate does not lead to an insertion error. In our model, the main difference of cognates vs. pseudo-cognates and near-cognates is in the kinetics. At various stages of the peptidyl transfer the rates for true cognates differ from the others up to three orders of magnitude.

Figure 1 depicts the relevant abstract automaton, derived from the Prism model discussed above. In case a transition is labeled with two rates, the leftmost number concerns the processing of a cognate aa-tRNA, the rightmost number that of a pseudo-cognate or near-cognate. In three states a probabilistic choice has to be made. The probabilistic choice in state 2 is the same for cognates, pseudo-cognates and near-cognates alike, the ones in state 3 and in state 4 differs for cognates and pseudo-cognates or near-cognates.

For example, after recognition in state 3, a cognate aa-tRNA will go through the hydrolysis phase leading to state 4 for a fraction 0.999 of the cases (computed as $260/(0.23 + 260)$), a fraction being close to 1. In contrast, for a pseudo-cognate or near-cognate aa-tRNA this is 0.005 only. Cognates will accommodate and continue to state 6 with probability 0.736, while pseudo-cognates and near-cognates will do so

[7] Our findings, see Table 4, based on the kinetic rates available are slightly higher.

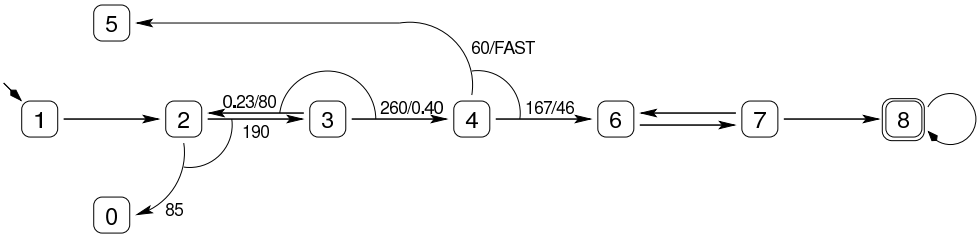

Figure 1: Abstract automaton for error insertion

with the small probability 0.044, the constant FAST being set to 1000 in our experiments. As the transition from state 4 to state 6 is irreversible, the rates of the remaining transitions are not of importance here.

The probability for reaching state 8 in one attempt can be easily computed by Prism via the CSL-formula

```
P=?  [ (s!=0 & s!=5) U (s=8) {(s=2) & cogn} ].
```

The formula asks to establish the probability for all paths where `s` is not set to `0` nor `5`, until `s` have been set to `8`, starting from the (unique) state satisfying `s=2 & cogn`. We obtain $p_s^c = 0.508$, $p_s^p = 0.484 \cdot 10^{-4}$ and $p_s^n = 0.484 \cdot 10^{-4}$, with p_s^c the probability for a cognate to end up in state 8 —and elongate the peptidyl chain— without going through state 0 nor state 5; p_s^p and p_s^n the analogues for pseudo- and near-cognates, respectively. Note that these values are the same for every codon. Different among codons are the concentrations of cognates, pseudo-cognates and near-cognates.[8] Ultimately, the frequencies f_c, f_p and f_n of the types of aa-tRNA in the cell, i.e. the actual number of molecules of the kind, determine the rates for an arrival

As reported in [7], the probability for an erroneous insertion, is strongly correlated with the quotient of the number of near-cognate anticodons and the number of cognate anticodons.[9] In the present setting, this correlation can be formally derived. We have that an insertion error occurs if a near-cognate succeeds to attach its amino acid. Therefore,

$$\begin{aligned} P(\text{error}) &= P(\text{near \& elongation} \mid \text{elongation}) \\ &= \frac{p_s^n \cdot (f_n/tot)}{p_s^c \cdot (f_c/tot) + p_s^p \cdot (f_p/tot) + p_s^n \cdot (f_n/tot)} \approx \frac{p_s^n \cdot f_n}{p_s^c \cdot f_c} \sim \frac{f_n}{f_c} \end{aligned}$$

with $tot = f_c + f_p + f_n$, and where we have used that

$$P(\text{elongation}) = (f_c/tot) \cdot p_s^c + (f_p/tot) \cdot p_s^p + (f_n/tot) \cdot p_s^n$$

and that $p_s^p, p_s^n \ll p_s^c$. Note, the ability to calculate the latter probabilities, illustrating that the approach of piecewise analysis, is instrumental in obtaining the above result.

5 Competition and insertion times

We continue the analysis of the Prism model for translation and discuss the correlation of the average insertion time for the amino acid specified by a codon, on the one hand, and the relative abundance of pseudo-cognate and near-cognate aa-tRNAs, on the other hand. The insertion time of a codon is the average time it takes to elongate the protein in nascent with an amino acid.

The average insertion time can be computed in Prism using the concept of *rewards* (also known as *costs* in Markov theory). Each state is assigned a value as its reward.

[8] See Table 3 in the appendix.

[9] See Figure 4 in the appendix.

Further, the reward of each state is weighted per unit of time. Hence, it is computed by multiplication with the average time spent in the state. The cumulative reward of a path in the chain is defined as a sum over all states in the path of such weighted rewards per state. Thus, by assigning to each state the value 1 as reward, we obtain the total average time for a given path. For example, in Prism the CSL formula `R=? [ F (s=8) ]` which asks to compute the expected time to reach state `s=8`. Recall, in state `s=8` the amino acid is added to the polypeptide chain. So, a script modelchecking the above formula then yields the expected insertion time per codon.[10] A little bit more ingenuity is needed to establish average exit times, for example for a cognate to pass from state `s=2` to state `s=8`. The point is that conditional probabilities are involved. However, since dealing exponential distributions, elimination of transition in favour of adding their rates to that of the remaining ones, does the trick. Various results, some of them used below, are collected in Table 2. (The probabilities of failure and success for the non-cognates are trivial, $p_f^x = 1$ and $p_s^x = 0$, with a time per failed attempt $T_f^x = 0.5 \cdot 10^{-3}$ seconds.)

p_s^c	0.5079	p_f^c	0.4921	T_s^c	0.03182	T_f^c	$9.342 \cdot 10^{-3}$
p_s^p	$4.847 \cdot 10^{-4}$	p_f^p	0.9995	T_s^p	3.251	T_f^p	0.3914
p_s^n	$4.847 \cdot 10^{-4}$	p_f^n	0.9995	T_s^n	3.251	T_f^n	0.3914

Table 2: Exit probabilities and times (in seconds) for three types of aa-tRNA. Failure for exit to states `s=0` or `s=5`; success for exit to state `s=8`.

There is a visible correlation between the quotient of the number of near-cognate aa-tRNA and the number of cognate aa-tRNA.[11] In fact, the average insertion time for a codon is approximately proportional to the near-cognate/cognate ratio. This can be seen as follows. The insertion of the amino acid is completed if state `s=8` is reached, either for a cognate, pseudo-cognate or near-cognate. As we have seen, the probability for the latter two is negligible. Therefore, the number of cognate arrivals is decisive. With p_f^c and p_s^c being the probability for a cognate to fail, i.e. exit at state `s=0` or `s=5`, or to succeed, i.e. reach of state `s=8`, the insertion time T_{ins} can be regarded as a geometric series. (Note the exponent i below.) Important are the numbers of arrivals of the other aa-tRNA types per single cognate arrival, expressed in terms of frequencies. We have

$$\begin{aligned} T_{ins} &= \textstyle\sum_{i=0}^{\top} (p_f^c)^i p_s^c \cdot ((\text{average delay for } i+1 \text{ cognate arrivals}) + T_s^c) \\ &= \textstyle\sum_{i=0}^{\top} (p_f^c)^i p_s^c \cdot (i \cdot (T_f^c + \frac{f_p}{f_c} T_f^p + \frac{f_n}{f_c} T_f^n + \frac{f_x}{f_c} T_f^x) + T_s^c) \\ &\approx \frac{f_p+f_n}{f_c} p_s^c \, T_f^n \textstyle\sum_{i=0}^{\top} i\,(p_f^c)^i \;\sim\; \frac{f_p+f_n}{f_c} . \end{aligned}$$

We have used that T_f^c and T_s^c are negligible, T_f^p equals T_f^n, and $\frac{f_x}{f_c} T_f^x$ is relatively small. Note that the estimate is not accurate for small values of $f_p + f_n$. Nevertheless, closer inspection show that for these values the approximation remains order-preserving. Again, the results obtained for parts of the systems are pivotal in the derivation.

6 Concluding remarks

In this paper, we presented a stochastic model of the translation process based presently available data of ribosome kinetics. We used the CTMC facilities of the Prism tool. Compared to simulation, our approach is computationally more reliable (independent

[10] See Table 5 in the appendix.

[11] See Figure 5 in the appendix.

on the number of simulations) and has faster response times (taking seconds rather then minutes or hours). More importantly, modelchecking allowed us to perform piecewise analysis of the system, yielding better insight in the model compared to just observing the end-to-end results with a monolithic model. Based on this, we improved on earlier observations, regarding error probabilities and insertion times, by actually deriving the correlation suggested by the data. In conclusion, we have experienced aa-tRNA competition as a very interesting biological case study of intrinsic stochastic nature, falling in the category of the well known lambda-phage example [1].

Our model opens a new avenue for future work on biological systems that possess intrinsically probabilistic properties. It would be interesting to apply our method to processes which, similarly to translation, require high precision, like DNA repair, charging of the tRNAs with amino acids, etc. Also, using our model one could check if amino acids with similar biochemical properties substitute erroneously for one another with greater probabilities than dissimilar ones.

References

[1] A. Arkin et al. Stochastic kinetic analysis of developmental pathway bifurcation in phage lambda-infected *Escherichia coli* cells. *Genetics*, 149:1633–1648, 1998.

[2] C. Baier et al. Approximate symbolic model checking of continuous-time Markov chains. In *Proc. CONCUR'99*, pages 146–161. LNCS 1664, 1999.

[3] M. Calder et al. Analysis of signalling pathways using continuous time Markov chains. In *Transactions on Computational Systems Biology VI*, pages 44–67. LNBI 4220, 2006.

[4] N. Chabrier and F. Fages. Symbolic model checking of biochemical networks. In *Proc. CMSB 2003*, pages 149–162. LNCS 2602, 2003.

[5] V. Danos et al. Rule-based modelling of cellular signalling. In *Proc. CONCUR*, pages 17–41. LNCS 4703, 2007.

[6] H. Dong et al. Co-variation of tRNA abundance and codon usage in *Escherichia coli* at different growth rates. *Journal of Molecular Biology*, 260:649–663, 1996.

[7] A. Fluitt et al. Ribosome kinetics and aa-tRNA competition determine rate and fidelity of peptide synthesis. *Computational Biology and Chemistry*, 31:335–346, 2007.

[8] M.A. Gilchrist and A. Wagner. A model of protein translation including codon bias, nonsense errors, and ribosome recycling. *Journal of Theoretical Biology*, 239:417–434, 2006.

[9] K.B. Gromadski and M.V. Rodnina. Kinetic determinants of high-fidelity tRNA discrimination on the ribosome. *Molecular Cell*, 13(2):191–200, 2004.

[10] J. Heath et al. Probabilistic model checking of complex biological pathways. In *Proc. CMSB 2006*, pages 32–47. LNBI 4210, 2006.

[11] A.W. Heyd and D.A. Drew. A mathematical model for elongation of a peptide chain. *Bulletin of Mathematical Biology*, 65:1095–1109, 2003.

[12] G. Karp. *Cell and Molecular Biology, 5th ed.* Wiley, 2008.

[13] M. Kwiatkowska et al. Probabilistic symbolic model cheking with Prism: a hybrid approach. *Journal on Software Tools for Technology Transfer*, 6:128–142, 2004. See also `http://www.prismmodelchecker.org/`.

[14] T. Pape et al. Complete kinetic mechanism of elongation factor Tu-dependent binding of aa-tRNA to the A-site of *E. coli*. *EMBO Journal*, 17:7490–7497, 1998.

[15] D. Parker. *Implementation of Symbolic Model Checking for Probabilistic Systems*. PhD thesis, University of Birmingham, 2002.

[16] C. Priami et al. Application of a stochastic name-passing calculus to represent ation and simulation of molecular processes. *Information Processing Letters*, 80:25–31, 2001.

[17] M.V. Rodnina and W. Wintermeyer. Ribosome fidelity: tRNA discrimination, proofreading and induced fit. *TRENDS in Biochemical Sciences*, 26(2):124–130, 2001.

[18] A. Savelsbergh et al. An elongation factor G-induced ribosome rearrangement precedes tRNA–mRNA translocation. *Molecular Cell*, 11:1517–1523, 2003.

Appendix: suplementary figures and data

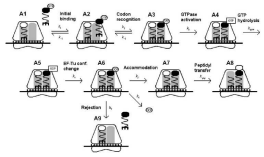

Figure 2: Kinetic scheme of peptidyl transfer taken from [7].

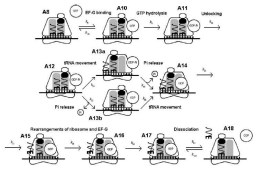

Figure 3: Kinetic scheme of translocation taken from [7].

```
// translation model

stochastic

// constants
const double ONE=1;
const double FAST=1000;

// tRNA rates
const double c_cogn ;
const double c_pseu ;
const double c_near ;
const double c_nonc ;

const double k1f = 140;
const double k2b =  85;
const double k2bx=2000;
const double k2f = 190;
const double k3bc=   0.23;
const double k3bp=  80;
const double k3bn=  80;
const double k3fc= 260;
const double k3fp=   0.40;
const double k3fn=   0.40;
const double k4rc=  60;
const double k4rp=FAST;
const double k4rn=FAST;
const double k4fc= 166.7;
const double k4fp=  46.1;
const double k4fn=  46.1;
const double k6f = 150;
const double k7b = 140;
const double k7f = 145.8;

module ribosome

s : [0..8] init 1 ;
cogn : bool init false ;
pseu : bool init false ;
near : bool init false ;
nonc : bool init false ;

// initial binding
[ ] (s=1) -> k1f * c_cogn : (s'=2) & (cogn'=true) ;
[ ] (s=1) -> k1f * c_pseu : (s'=2) & (pseu'=true) ;
[ ] (s=1) -> k1f * c_near : (s'=2) & (near'=true) ;
[ ] (s=1) -> k1f * c_nonc : (s'=2) & (nonc'=true) ;
[ ] (s=2) &  ( cogn | pseu | near ) -> k2b : (s'=0) &
      (cogn'=false) & (pseu'=false) & (near'=false) ;
[ ] (s=2) &  nonc -> k2bx : (s'=0) & (nonc'=false) ;

// codon recognition
[ ] (s=2) & ( cogn | pseu | near ) -> k2f : (s'=3) ;
[ ] (s=3) & cogn -> k3bc : (s'=2) ;
[ ] (s=3) & pseu -> k3bp : (s'=2) ;
[ ] (s=3) & near -> k3bn : (s'=2) ;

// GTPase activation, GTP hydrolysis, reconformation
[ ] (s=3) & cogn -> k3fc : (s'=4) ;
[ ] (s=3) & pseu -> k3fp : (s'=4) ;
[ ] (s=3) & near -> k3fn : (s'=4) ;

// rejection
[ ] (s=4) & cogn -> k4rc : (s'=5) & (cogn'=false) ;
[ ] (s=4) & pseu -> k4rp : (s'=5) & (pseu'=false) ;
[ ] (s=4) & near -> k4rn : (s'=5) & (near'=false) ;

// accommodation, peptidyl transfer
[ ] (s=4) & cogn -> k4fc : (s'=6) ;
[ ] (s=4) & pseu -> k4fp : (s'=6) ;
[ ] (s=4) & near -> k4fn : (s'=6) ;

// EF-G binding
[ ] (s=6) -> k6f : (s'=7) ;
[ ] (s=7) -> k7b : (s'=6) ;

// GTP hydrolysis, unlocking,
// tRNA movement and Pi release,
// rearrangements of ribosome and EF-G,
// dissociation of GDP
[ ] (s=7) -> k7f : (s'=8) ;

// no entrance, re-entrance at state 1
[ ] (s=0) -> FAST*FAST : (s'=1) ;
// rejection, re-entrance at state 1
[ ] (s=5) -> FAST*FAST : (s'=1) ;
// elongation
[ ] (s=8) -> FAST*FAST : (s'=8) ;

endmodule

rewards
  true : 1;
endrewards
```

codon	cognate	pseudo-cognate	near-cognate	non-cognate	codon	cognate	pseudo-cognate	near-cognate	non-cognate
UUU	1037	0	2944	67493	GUU	5105	0	0	66369
UUC	1037	0	9904	60533	GUC	1265	3840	7372	58997
UUG	2944	0	2324	66206	GUG	3840	1265	1068	65301
UUA	1031	1913	2552	65978	GUA	3840	1265	9036	57333
UCU	2060	344	0	69070	GCU	3250	617	0	67607
UCC	764	1640	4654	64416	GCC	617	3250	8020	59587
UCG	1296	764	2856	66558	GCG	3250	617	1068	66539
UCA	1296	1108	1250	67820	GCA	3250	617	9626	57981
UGU	1587	0	1162	68725	GGU	4359	2137	0	64978
UGC	1587	0	4993	64894	GGC	4359	2137	4278	60700
UGG	943	0	4063	66468	GGG	2137	4359	0	64978
UGA	6219	0	4857	60398	GGA	1069	5427	11807	53171
UAU	2030	0	0	69444	GAU	2396	0	4717	64361
UAC	2030	0	3388	66056	GAC	2396	0	10958	58120
UAG	1200	0	5230	65044	GAG	4717	0	3464	63293
UAA	7200	0	4576	59698	GAA	4717	0	10555	56202
CUU	943	5136	4752	60643	AUU	1737	1737	2632	65368
CUC	943	5136	1359	64036	AUC	1737	1737	6432	61568
CUG	5136	943	2420	62975	AUG	706	1926	4435	64407
CUA	666	5413	1345	64050	AUA	1737	1737	6339	61661
CCU	1301	900	4752	64521	ACU	2115	541	0	68818
CCC	1913	943	2120	66498	ACC	1199	1457	4338	64480
CCG	1481	720	5990	63283	ACG	1457	1199	4789	64029
CCA	581	1620	1430	67843	ACA	916	1740	2791	66027
CGU	4752	639	0	66083	AGU	1408	0	1287	68779
CGC	4752	639	2302	63781	AGC	1408	0	5416	64650
CGG	639	4752	6251	59832	AGG	420	867	6318	63869
CGA	4752	639	2011	64072	AGA	867	420	4248	65939
CAU	639	0	6397	64438	AAU	1193	0	1924	68357
CAC	639	0	3308	67527	AAC	1193	0	6268	64013
CAG	881	764	6648	63181	AAG	1924	0	6523	63027
CAA	764	881	1886	67943	AAA	1924	0	2976	66574

Table 3: Frequencies of cognate, pseudo-cognate, near-cognate and non-cognates for *E. coli* as molecules per cell [6].

UUU	0.002741862683943581	CUU	0.004663729080892617
UUC	0.009117638314789647	CUC	0.0013623408749670932
UUG	7.588473846528858e-4	CUG	4.487561228352708e-4
UUA	0.0023468531911491246	CUA	0.0018888580411442013
UCU	2.8056841829690867e-10	CCU	0.0034116470820387637
UCC	0.005606123319450197	CCC	0.0010419283146932763
UCG	0.002032726835647694	CCG	0.003761852345052361
UCA	9.090727755350428e-4	CCA	0.0022775137744062385
UGU	6.966884002285479e-4	CGU	1.207693755014732e-10
UGC	0.0030362362683066077	CGC	4.587111916100053e-4
UGG	0.003978308597370318	CGG	0.008874544692533565
UGA	7.498426342500918e-4	CGA	3.9837866155798695e-4
UAU	2.8061598550623636e-10	CAU	0.009105588393934699
UAC	0.001568960520388667	CAC	0.004745578685847523
UAG	0.004132405628997547	CAG	0.0069400807775903016
UAA	6.039804446811093e-4	CAA	0.0022666704102712373

GUU	1.122602539973544e-10	AUU	0.0014440395784868422
GUC	0.005495266825145313	AUC	0.0035043308185745276
GUG	2.6820764780942726e-4	AUG	0.005831774423967932
GUA	0.0022306329982350647	AUA	0.0034390541040541776
GCU	1.766661283697676e-10	ACU	2.725325694334536e-10
GCC	0.01245896879253996	ACC	0.0034184472357413403
GCG	3.1789705950373547e-4	ACG	0.003167334470509804
GCA	0.002818616263545499	ACA	0.0029111153328695892
GGU	1.3246548978903072e-10	AGU	8.70279113272123e-4
GGC	9.396128218189778e-4	AGC	0.003719031341166648
GGG	2.7206107910251926e-10	AGG	0.01406993213919797
GGA	0.010230631644252862	AGA	0.004811394879822719
GAU	0.0018570532571304608	AAU	0.0015239834703624298
GAC	0.004322322632194155	AAC	0.00493586499554021
GAG	7.090294740031601e-4	AAG	0.003209595977078994
GAA	0.002136227458736717	AAA	0.0014587873027927622

Table 4: Probabilities per codon for erroneous elongation

UUU	0.3327	CUU	0.8901	GUU	0.0527	AUU	0.2733
UUC	0.8404	CUC	0.6286	GUC	0.7670	AUC	0.4373
UUG	0.1245	CUG	0.1028	GUG	0.1041	AUG	0.8115
UUA	0.4436	CUA	0.9217	GUA	0.2604	AUA	0.4321
UCU	0.0893	CCU	0.4202	GCU	0.0756	ACU	0.0943
UCC	0.7409	CCC	0.1992	GCC	1.5622	ACC	0.4658
UCG	0.3035	CCG	0.4257	GCG	0.1010	ACG	0.4073
UCA	0.2313	CCA	0.5535	GCA	0.3002	ACA	0.5025
UGU	0.1432	CGU	0.0645	GGU	0.0924	AGU	0.1636
UGC	0.3296	CGC	0.1010	GGC	0.1673	AGC	0.3905
UGG	0.4360	CGG	1.3993	GGG	0.2308	AGG	1.4924
UGA	0.1098	CGA	0.0962	GGA	1.2989	AGA	0.5517
UAU	0.0758	CAU	0.8811	GAU	0.2180	AAU	0.2242
UAC	0.2008	CAC	0.5341	GAC	0.4144	AAC	0.4959
UAG	0.4319	CAG	0.7425	GAG	0.1106	AAG	0.3339
UAA	0.0963	CAA	0.4058	GAA	0.2243	AAA	0.1945

Table 5: Estimated average insertion time per codon in seconds

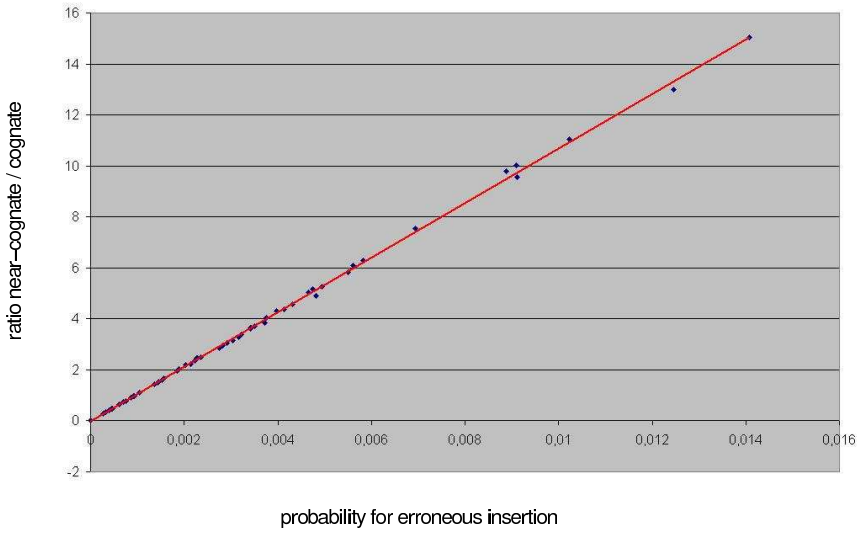

Figure 4: Correlation of $\frac{f_n}{f_c}$ ratio and error probabilities

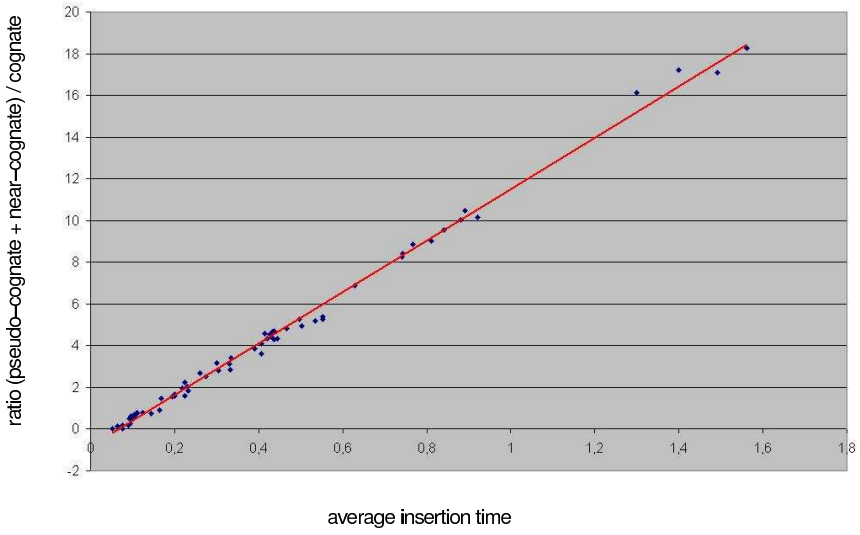

Figure 5: Correlation of $\frac{f_p+f_n}{f_c}$ ratio and average insertion times

A new mathematical model for the heat shock response

Ion Petre[1,2,5]
ipetre@abo.fi

Andrzej Mizera[1,2]
amizera@abo.fi

Claire Hyder[3,4]
chyder@btk.fi

Andrey Mikhailov[3,4]
andrey.mikhailov@btk.fi

John Eriksson[3,4]
john.eriksson@btk.fi

Lea Sistonen[3,4,5]
lea.sistonen@btk.fi

Ralph-Johan Back[1,2,5]
backrj@abo.fi

[1] Åbo Akademi University, Department of Information Technologies

[2] Turku Centre for Computer Science

[3] Åbo Akademi University, Department of Biochemistry

[4] Turku Centre for Biotechnology

[5] Academy of Finland

Abstract

We present in this paper a novel molecular model for the gene regulatory network responsible for the eukaryotic heat shock response. Our model includes the temperature-induced protein misfolding, the chaperone activity of the heat shock

proteins and the backregulation of their gene transcription. We then build a mathematical model for it, based on ordinary differential equations. Finally, we discuss the parameter fit and the implications of the sensitivity analysis for our model.

1 Introduction

One of the most impressive algorithmic-like bioprocesses in living cells, crucial for the very survival of cells is the *heat shock response*: the reaction of the cell to elevated temperatures. One of the effects of raised temperature in the environment is that proteins get misfolded, with a rate that is exponentially dependent on the temperature. As an effect of their hydrophobic core being exposed, misfolded proteins tend to form bigger and bigger aggregates, with disastrous consequences for the cell, see [1]. To survive, the cell needs to increase quickly the level of chaperons (proteins that are assisting in the folding or refolding of other proteins). Once the heat shock is removed, the cell eventually re-establishes the original level of chaperons, see [10, 18, 22].

The heat shock response has been subject of intense research in the last few years, for at least three reasons. First, it is a well-conserved mechanism across all eukaryotes, while bacteria exhibit only a slightly different response, see [5, 12, 23]. As such, it is a good candidate for studying the engineering principle of gene regulatory networks, see [4, 5, 12, 25]. Second, it is a tempting mechanism to model mathematically, since it involves only very few reactants, at least in a simplified presentation, see [18, 19, 22]. Third, the heat shock proteins (the main chaperons involved in the eukaryotic heat shock response) play a central role in a large number of regulatory and of inflammatory processes, as well as in signaling, see [9, 20]. Moreover, they contribute to the resilience of cancer cells, which makes them attractive as targets for cancer treatment, see [3, 15, 16, 27].

We focus in this paper on a new molecular model for the heat shock response, proposed in [19]. We consider here a slight extension of the model in [19] where, among others, the chaperons are also subject to misfolding. After introducing the molecular model in Section 2, we build a mathematical model in Section 3, including the fitting of the model with respect to experimental data. We discuss in Section 4 the results of the sensitivity analysis of the model, including its biological implications.

2 A new molecular model for the eukaryotic heat shock response

The heat shock proteins (hsp) play the key role in the heat shock response. They act as chaperons, helping misfolded proteins (mfp) to refold. The response is controlled in our model through the regulation of the transactivation of the hsp-encoding genes. The transcription of the gene is promoted by some proteins called heat shock factors (hsf) that trimerize and then bind to a specific DNA sequence called heat shock element (hse), upstream of the hsp-encoding gene. Once the hsf trimer is bound to the heat shock element, the gene is transactivated and the synthesis of hsp is thus switched on (for the sake of simplicity, the role of RNA is ignored in our model). Once the level of hsp is high enough, the cell has an ingenious mechanism to switch off the hsp synthesis. For this, hsp bind to free hsf, as well as break the hsf trimers (including those bound to hse, promoting the gene activation), thus effectively halting the hsp synthesis.

Under elevated temperatures, some of the proteins (prot) in the cell get misfolded. The heat shock response is then quickly switched on simply because the heat shock

proteins become more and more active in the refolding process, thus leaving the heat shock factors free and able to promote the synthesis of more heat shock proteins. Note that several types of heat shock proteins exist in an eukaryotic cell. We treat them all uniformly in our model, with hsp70 as common denominator. The same comment applies also to the heat shock factors.

Our molecular model for the eukaryotic heat shock response consists of the following molecular reactions:

1. $2\ \mathsf{hsf} \leftrightarrows \mathsf{hsf}_2$
2. $\mathsf{hsf} + \mathsf{hsf}_2 \leftrightarrows \mathsf{hsf}_3$
3. $\mathsf{hsf}_3 + \mathsf{hse} \leftrightarrows \mathsf{hsf}_3\mathsf{:hse}$
4. $\mathsf{hsf}_3\mathsf{:hse} \rightarrow \mathsf{hsf}_3\mathsf{:hse} + \mathsf{mhsp}$
5. $\mathsf{hsp} + \mathsf{hsf} \leftrightarrows \mathsf{hsp:hsf}$
6. $\mathsf{hsp} + \mathsf{hsf}_2 \rightarrow \mathsf{hsp:hsf} + \mathsf{hsf}$
7. $\mathsf{hsp} + \mathsf{hsf}_3 \rightarrow \mathsf{hsp:hsf} + 2\ \mathsf{hsf}$
8. $\mathsf{hsp} + \mathsf{hsf}_3\mathsf{:hse} \rightarrow \mathsf{hsp:hsf} + 2\ \mathsf{hsf} + \mathsf{hse}$
9. $\mathsf{hsp} \rightarrow \emptyset$
10. $\mathsf{prot} \rightarrow \mathsf{mfp}$
11. $\mathsf{hsp} + \mathsf{mfp} \leftrightarrows \mathsf{hsp:mfp}$
12. $\mathsf{hsp:mfp} \rightarrow \mathsf{hsp} + \mathsf{prot}$
13. $\mathsf{hsf} \rightarrow \mathsf{mhsf}$
14. $\mathsf{hsp} \rightarrow \mathsf{mhsp}$
15. $\mathsf{hsp} + \mathsf{mhsf} \leftrightarrows \mathsf{hsp:mhsf}$
16. $\mathsf{hsp:mhsf} \rightarrow \mathsf{hsp} + \mathsf{hsf}$
17. $\mathsf{hsp} + \mathsf{mhsp} \leftrightarrows \mathsf{hsp:mhsp}$
18. $\mathsf{hsp:mhsp} \rightarrow 2\ \mathsf{hsp}$

It is important to note that the main addition we consider here with respect to the model in [19] is to include the misfolding of hsp and hsf. This is, in principle, no minor extension since in the current model the repairing mechanism is subject to failure, but it is capable to fix itself.

Several criteria were followed when introducing this molecular model:

(i) as few reactions and reactants as possible;

(ii) include the temperature-induced protein misfolding;

(iii) include hsf in all its three forms: monomers, dimers, and trimers;

(iv) include the hsp-backregulation of the transactivation of the hsp-encoding gene;

(v) include the chaperon activity of hsp;

(vi) include only well-documented, textbook-like reactions and reactants.

For the sake of keeping the model as simple as possible, we are ignoring a number of details. E.g., note that there is no notion of locality in our model: we make no distinction between the place where gene transcription takes place (inside nucleus) and the place where protein synthesis takes place (outside nucleus). Note also that protein synthesis and gene transcription are greatly simplified in reaction 4: we only indicate that once the gene is transactivated, protein synthesis is also switched on. On the other hand, reaction 4 is faithful to the biological reality, see [1] in indicating that newly synthesized proteins often need chaperons to form their native fold.

As far as protein degradation is concerned, we only consider it in the model for hsp. If we considered it also for hsf and prot, then we should also consider the compensating mechanism of protein synthesis, including its control. For the sake of simplicity and also based on experimental evidence that the total amount of hsf and of prot is somewhat constant, we ignore the details of synthesis and degradation for hsf and prot.

3 The mathematical model

We build in this section a mathematical model associated to the molecular model 1–18. Our mathematical model is in terms of coupled ordinary differential equations and its formulation is based on the principle of mass-action.

3.1 The principle of mass-action

The mass-action law is widely used in formulating mathematical models in physics, chemistry, and engineering. Introduced in [6, 7], it can be briefly summarized as follows: *the rate of each reaction is proportional to the concentration of reactants*. In turn, the rate of each reaction gives the rate of consuming the reactants and the rate of producing the products. E.g., for a reaction

$$R_1\colon A + B \to C,$$

the rate according to the principle of mass action is $f_1(t) = kA(t)B(t)$, where $k \geq 0$ is a constant and $A(t)$, $B(t)$ are functions of time giving the level of the reactants A and B, respectively. Consequently, the rate of consuming A and B, and the rate of producing C is expressed by the following differential equations:

$$\frac{dA}{dt} = \frac{dB}{dt} = -k\,A(t)\,B(t), \qquad \frac{dC}{dt} = k\,A(t)\,B(t).$$

For a reversible reaction

$$R_2\colon A + B \leftrightarrows C,$$

the rate is $f_2(t) = k_1\,A(t)\,B(t) - k_2\,C(t)$, for some constants $k_1, k_2 \geq 0$. The differential equations are written in a similar way:

$$\frac{dA}{dt} = \frac{dB}{dt} = -f_2(t), \qquad \frac{dC}{dt} = f_2(t). \tag{*}$$

For a set of coupled reactions, the differential equations capture the combined rate of consuming and producing each reactant as an effect of all reactions taking place simultaneously. E.g., for reactions

$$R_3\colon A + B \leftrightarrows C, \qquad R_4\colon B + C \leftrightarrows A, \qquad R_5\colon A + C \leftrightarrows B,$$

the associated system of differential equations is

$$\begin{aligned} dA/dt &= -f_3(t) + f_4(t) - f_5(t), \\ dB/dt &= -f_3(t) - f_4(t) + f_5(t), \\ dC/dt &= f_3(t) - f_4(t) - f_5(t), \end{aligned}$$

where $f_i(t)$ is the rate of reaction R_i, for all $3 \leq i \leq 5$, formulated according to the principle of mass action.

We recall that for a system of differential equations

$$\begin{aligned} \frac{dX_1}{dt} &= f_1(X_1, \ldots, X_n), \\ &\cdots \\ \frac{dX_n}{dt} &= f_n(X_1, \ldots, X_n), \end{aligned}$$

we say that $(x_1, x_2, \ldots, x_n)$ is a *steady states* (also called *equilibrium points*) if it is a solution of the algebraic system of equations $f_i(X_1, \ldots, X_n) = 0$, for all $1 \leq i \leq n$, see [24, 28]. Steady states are particularly interesting because they characterize situations where although reactions may have non-zero rates, their combined effect is zero. In other words, the concentration of all reactants and of all products are constant.

We refer to [11, 17, 29] for more details on the principle of mass action and its formulation based on ordinary differential equations.

3.2 Our mathematical model

Let $\mathbb{R}_+$ be the set of all positive real numbers and $\mathbb{R}_+^n$ the set of all n-tuples of positive real numbers, for $n \geq 2$. We denote each reactant and bond between them in the molecular model 1–18 according to the convention in Table 3.2. We also denote by $\kappa \in \mathbb{R}_+^{17}$ the vector with all reaction rate constants as its components, see Table 3.2: $\kappa = (k_1^+, k_1^-, k_2^+, k_2^-, k_3^+, k_3^-, k_4, k_5^+, k_5^-, k_6, k_7, k_8, k_9, k_{11}^+, k_{11}^-, k_{12}, k_{13}^+, k_{13}^-, k_{14}, k_{15}^+, k_{15}^-, k_{16})$.

The mass action-based formulation of the associated mathematical model in terms of differential equations is straightforward, leading to the following system of equations:

$$\begin{aligned} dX_1/dt &= f_1(X_1, X_2, \ldots, X_{14}, \kappa) && (1) \\ dX_2/dt &= f_2(X_1, X_2, \ldots, X_{14}, \kappa) && (2) \\ dX_3/dt &= f_3(X_1, X_2, \ldots, X_{14}, \kappa) && (3) \\ dX_4/dt &= f_4(X_1, X_2, \ldots, X_{14}, \kappa) && (4) \\ dX_5/dt &= f_5(X_1, X_2, \ldots, X_{14}, \kappa) && (5) \\ dX_6/dt &= f_6(X_1, X_2, \ldots, X_{14}, \kappa) && (6) \\ dX_7/dt &= f_7(X_1, X_2, \ldots, X_{14}, \kappa) && (7) \\ dX_8/dt &= f_8(X_1, X_2, \ldots, X_{14}, \kappa) && (8) \\ dX_9/dt &= f_9(X_1, X_2, \ldots, X_{14}, \kappa) && (9) \\ dX_{10}/dt &= f_{10}(X_1, X_2, \ldots, X_{14}, \kappa) && (10) \\ dX_{11}/dt &= f_{11}(X_1, X_2, \ldots, X_{14}, \kappa) && (11) \\ dX_{12}/dt &= f_{12}(X_1, X_2, \ldots, X_{14}, \kappa) && (12) \\ dX_{13}/dt &= f_{13}(X_1, X_2, \ldots, X_{14}, \kappa) && (13) \\ dX_{14}/dt &= f_{14}(X_1, X_2, \ldots, X_{14}, \kappa) && (14) \end{aligned}$$

Metabolite	Variable	Initial value	A steady state (T=42)
hsf	X_1	0.669	0.669
hsf_2	X_2	$8.73 \cdot 10^{-4}$	$8.73 \cdot 10^{-4}$
hsf_3	X_3	$1.23 \cdot 10^{-4}$	$1.23 \cdot 10^{-4}$
hsf_3: hse	X_4	2.956	2.956
mhsf	X_5	$3.01 \cdot 10^{-6}$	$2.69 \cdot 10^{-5}$
hse	X_6	29.733	29.733
hsp	X_7	766.875	766.875
mhsp	X_8	$3.45 \cdot 10^{-3}$	$4.35 \cdot 10^{-2}$
hsp: hsf	X_9	1403.13	1403.13
hsp: mhsf	X_{10}	$4.17 \cdot 10^{-7}$	$3.72 \cdot 10^{-6}$
hsp: mhsp	X_{11}	$4.78 \cdot 10^{-4}$	$6.03 \cdot 10^{-3}$
hsp: mfp	X_{12}	71.647	640.471
prot	X_{13}	$1.14 \cdot 10^{8}$	$1.14 \cdot 10^{8}$
mfp	X_{14}	517.352	4624.72

Table 1: The list of variables in the mathematical model, their initial values, and their values in one of the steady states of the system, for $T = 42$. Note that the initial values give one of the steady states of the system for $T = 37$.

where

$$
\begin{aligned}
f_1 &= -k_2^+ X_1 X_2 + k_2^- X_3 - k_5^+ X_1 X_7 + k_5^- X_9 + 2\,k_8 X_4 X_7 + k_6 X_2 X_7 \\
&\quad -\varphi(T) X_1 + k_{14} X_{10} + 2\,k_7 X_3 X_7 - 2\,k_1^+ X_1^2 + 2\,k_1^- X_2 \\
f_2 &= -k_2^+ X_1 X_2 + k_2^+ X_3 - k_6 X_2 X_7 + k_1^+ X_1^2 - k_1^- X_2 \\
f_3 &= -k_3^+ X_3 X_6 + k_2^+ X_1 X_2 - k_2^- X_3 + k_3^- X_4 - k_7 X_3 X_7 \\
f_4 &= k_3^+ X_3 X_6 - k_3^- X_4 - k_8 X_4 X_7 \\
f_5 &= \varphi(T) X_1 - k_{13}^+ X_5 X_7 + k_{13}^- X_{10} \\
f_6 &= -k_3^+ X_3 X_6 + k_3^- X_4 + k_8 X_4 X_7 \\
f_7 &= -k_5^+ X_1 X_7 + k_5^- X_9 - k_{11}^+ X_7 X_{14} + k_{11}^- X_{12} - k_8 X_4 X_7 - k_6 X_2 X_7 \\
&\quad -k_{13}^+ X_5 X_7 + (k_{13}^- + k_{14}) X_{10} - (\varphi(T) + k_9) X_7 - k_{15}^+ X_7 X_8 \\
&\quad -k_7 X_3 X_7 + (k_{15}^- + 2\,k_{16}) X_{11} + k_{12} X_{12} \\
f_8 &= k_4 X_4 + \varphi(T) X_7 - k_{15}^+ X_7 X_8 + k_{15}^- X_{11} \\
f_9 &= k_5^+ X_1 X_7 - k_5^- X_9 + k_8 X_4 X_7 + k_6 X_2 X_7 + k_7 X_3 X_7 \\
f_{10} &= k_{13}^+ X_5 X_7 - (k_{13}^- + k_{14}) X_{10} \\
f_{11} &= k_{15}^+ X_7 X_8 - (k_{15}^- + k_{16}) X_{11} \\
f_{12} &= k_{11}^+ X_7 X_{14} - (k_{11}^- + k_{12}) X_{12} \\
f_{13} &= k_{12} X_{12} - \varphi(T) X_{13} \\
f_{14} &= -k_{11}^+ X_7 X_{14} + k_{11}^- X_{12} + \varphi(T) X_{13}
\end{aligned}
$$

The rate of protein misfolding $\varphi(T)$ with respect to temperature T has been investigated experimentally in [13, 14], and a mathematical expression for it has been proposed in [18]. We have adapted the formula in [18] to obtain the following misfolding

Kinetic constant	Reaction	Numerical value
k_1^+	(1), forward	3.49091
k_1^-	(1), backward	0.189539
k_2^+	(2), forward	1.06518
k_2^-	(2), backward	$1 \cdot 10^{-9}$
k_3^+	(3), forward	0.169044
k_3^-	(3), backward	$1.21209 \cdot 10^{-6}$
k_4	(4)	0.00830045
k_5^+	(5), forward	9.73665
k_5^-	(5), backward	3.56223
k_6	(6)	2.33366
k_7	(7)	$4.30924 \cdot 10^{-5}$
k_8	(8)	$2.72689 \cdot 10^{-7}$
k_9	(9)	$3.2 \cdot 10^{-5}$
k_{11}^+	(11), forward	0.00331898
k_{11}^-	(11), backward	4.43952
k_{12}	(12)	13.9392
k_{13}^+	(15), forward	0.00331898
k_{13}^-	(15), backward	4.43952
k_{14}	(16)	13.9392
k_{15}^+	(17), forward	0.00331898
k_{15}^-	(17), backward	4.43952
k_{16}	(18)	13.9392

Table 2: The numerical values for the fitted model.

rate per second:

$$\varphi(T) = (1 - \frac{0.4}{e^{T-37}}) \cdot 0.8401033733 \cdot 10^{-6} \cdot 1.4^{T-37}\ s^{-1},$$

where T is the temperature of the environment in Celsius degrees, with the formula being valid for $37 \leq T \leq 45$.

The following result gives three mass-conservation relations for our model.

Theorem 3.1. *There exists $K_1, K_2, K_3 \geq 0$ such that:*

(i) $X_1(t) + 2\,X_2(t) + 3\,X_3(t) + 3\,X_4(t) + X_5(t) + X_9(t) = K_1$,

(ii) $X_4(t) + X_6(t) = K_2$,

(iii) $X_{13}(t) + X_{14}(t) + X_{12}(t) = K_3$,

for all $t \geq 0$.

Proof. We only prove here part (ii), as the others may be proved analogously. For this, note that from equations (4) and (6), it follows that

$$\frac{d(X_4 + X_6)}{dt} = (f_4 + f_6)(X_1, \ldots, X_{14}, \kappa, t) = 0,$$

i.e., $(X_4 + X_6)(t)$ is a constant function. □

The steady states of the model (1)-(14) satisfy the following algebraic relations, where x_i is the numerical value of X_i in the steady state, for all $1 \leq i \leq 14$.

$$\begin{aligned}
0 &= -k_2^+ x_1 x_2 + k_2^- x_3 - k_5^+ x_1 x_7 + k_5^- x_9 + 2 k_8 x_4 x_7 + k_6 x_2 x_7 \\
&\quad -\varphi(T) x_1 + k_{14} x_{10} + 2 k_7 x_3 x_7 - 2 k_1^+ x_1^2 + 2 k_1^- x_2 && (15)\\
0 &= -k_2^+ x_1 x_2 + k_2^+ x_3 - k_6 x_2 x_7 + k_1^+ x_1^2 - k_1^- x_2 && (16)\\
0 &= -k_3^+ x_3 x_6 + k_2^+ x_1 x_2 - k_2^- x_3 + k_3^- x_4 - k_7 x_3 x_7 && (17)\\
0 &= k_3^+ x_3 x_6 - k_3^- x_4 - k_8 x_4 x_7 && (18)\\
0 &= \varphi(T) x_1 - k_{13}^+ x_5 x_7 + k_{13}^- x_{10} && (19)\\
0 &= -k_3^+ x_3 x_6 + k_3^- x_4 + k_8 x_4 x_7 && (20)\\
0 &= -k_5^+ x_1 x_7 + k_5^- x_9 - k_{11}^+ x_7 x_{14} + k_{11}^- x_{12} - k_8 x_4 x_7 - k_6 x_2 x_7 \\
&\quad -k_{13}^+ x_5 x_7 + (k_{13}^- + k_{14}) x_{10} - (\varphi(T) + k_9) x_7 - k_{15}^+ x_7 x_8 - k_7 x_3 x_7 \\
&\quad +(k_{15}^- + 2 k_{16}) x_{11} + k_{12} x_{12} && (21)\\
0 &= k_4 x_4 + \varphi(T) x_7 - k_{15}^+ x_7 x_8 + k_{15}^- x_{11} && (22)\\
0 &= k_5^+ x_1 x_7 - k_5^- x_9 + k_8 x_4 x_7 + k_6 x_2 x_7 + k_7 x_3 x_7 && (23)\\
0 &= k_{13}^+ x_5 x_7 - (k_{13}^- + k_{14}) x_{10} && (24)\\
0 &= k_{15}^+ x_7 x_8 - (k_{15}^- + k_{16}) x_{11} && (25)\\
0 &= k_{11}^+ x_7 x_{14} - (k_{11}^- + k_{12}) x_{12} && (26)\\
0 &= k_{12} x_{12} - \varphi(T) x_{13} && (27)\\
0 &= -k_{11}^+ x_7 x_{14} + k_{11}^- x_{12} + \varphi(T) x_{13} && (28)
\end{aligned}$$

It follows from Theorem 3.1 that only eleven of the relations above are independent. E.g., relations (15)-(17), (19), (21)-(27) are independent. The system consisting of the corresponding differential equations is called the *reduced system* of (1)-(14).

3.3 Fitting the model to experimental data

The experimental data available for the parameter fit is from [10] and reflects the level of DNA binding, i.e., variable X_4 in our model, for various time points up to 4 hours, with continuous heat shock at 42 °C. Additionally, we require that the initial value of the variables of the model is a steady state for temperature set to 37 °C. This is a natural condition since the model is supposed to reflect the reaction to temperatures raised above 37 °C.

Mathematically, the problem we need to solve is one of global optimization, as formulated below. For each 17-tuple κ of positive numerical values for all kinetic constants, and for each 14-tuple α of positive initial values for all variables in the model, the function $X_4(t)$ is uniquely defined for a fixed temperature T. We denote the value of this function at time point τ, with parameters κ and α by $x_4^T(\kappa, \alpha, \tau)$. Note that this property holds for all the other variables in the model and it is valid in general for any mathematical model based on ordinary differential equations (one calls such models *deterministic*). We denote the set of experimental data in [10] by

$$E_n = \{(t_i, r_i) \mid t_i, r_i \rangle 0,\ 1 \leq i \leq N\},$$

where $N \geq 1$ is the number of observations, t_i is the time point of each observation and r_i is the value of the reading.

With this setup, we can now formulate our optimization problem as follows: find $\kappa \in \mathbb{R}_+^{17}$ and $\alpha \in \mathbb{R}_+^{14}$ such that:

(i) $f(\kappa, \alpha) = \frac{1}{N} \sum_{i=1}^{N} (x_4^{42}(\kappa, \alpha, t_i) - r_i)^2$ is minimal and

(ii) α is a steady state of the model for $T = 37$ and parameter values given by κ.

The function $f(\kappa, \alpha)$ is a cost function (in this case *least mean squares*), indicating numerically how the function $x_4^T(\kappa, \alpha, t)$, $t \geq 0$, compares with the experimental data.

Note that in our optimization problem, not all 31 variables (the components of κ and α) are independent. On one hand, we have the three algebraic relations given by Theorem 3.1. On the other hand, we have eleven more independent algebraic relations given by the steady state equations (15)-(17), (19), (21)-(27). Consequently, we have 17 independent variables in our optimization problem.

Given the high degree of the system (1)-(14), finding the analytical form of the minimum points of $f(\kappa, \alpha)$ is very challenging. This is a typical problem when the system of equations is non-linear. Adding to the difficulty of the problem is the fact that the eleven independent steady state equations cannot be solved analytically, given their high overall degree.

Since an analytical solution to the model fitting problem is often intractable, the practical approach to such problems is to give a numerical simulation of a solution. Several methods exist for this, see [2, 21]. The trade-off with all these methods is that typically they offer an estimate of a *local* optimum, with no guarantee of it being a *global* optimum.

Obtaining a numerical estimation of a local optimum for (i) is not difficult. However, such a solution may not satisfy (ii). To solve this problem, for a given local optimum $(\kappa_0, \alpha_0) \in \mathbb{R}_+^{17} \times \mathbb{R}_+^{14}$ one may numerically estimate a steady state $\alpha_1 \in \mathbb{R}_+^{14}$ for $T = 37$. Then the pair (κ_0, α_1) satisfies (ii). Unfortunately, (κ_0, α_1) may not be close to a local optimum of the cost function in (i).

Another approach is to replace the algebraic relations implicitly given by (ii) with an optimization problem similar to that in (i). Formally, we replace all algebraic relations $R_i = 0$, $1 \leq i \leq 11$, given by (ii) with the condition that

$$g(\kappa, \alpha) = \frac{1}{M} \sum_{j=1}^{M} R_i^2(\kappa, \alpha, \delta_j)$$

is minimal, where $0 \langle \delta_1 \langle \cdots \langle \delta_M$ are some arbitrary (but fixed) time points. Our problem thus becomes one of optimization with cost function (f, g), with respect to the order relation $(a, b) \leq (c, d)$ if and only if $a \leq c$ and $b \leq d$. The numerical values in Table 3.2 give one solution to this problem obtained based on Copasi [8]. The plot in Figure 1 shows the time evolution of function $X_4(t)$ up to $t = 4$ hours, with the experimental data of [10] indicated with crosses.

The solution in Table 3.2 has been compared with a number of other available experimental data (such as behavior at 41 °C and at 43 °C), as well as against qualitative, non-numerical data. The results were satisfactory and better than those of previous models reported in the literature, such as [18, 22]. For details on the model validation analysis we refer to [19].

Note that the steady state of the system of differential equations (1)-(14), for the initial values in Table 3.2 and the parameter values in Table 3.2 is *asymptotically stable*. To prove it, it is enough to consider its associated *Jacobian*:

$$J(t) = \begin{pmatrix} \partial f_1/\partial X_1 & \partial f_1/\partial X_2 & \dots & \partial f_1/\partial X_{14} \\ \partial f_2/\partial X_1 & \partial f_2/\partial X_2 & \dots & \partial f_2/\partial X_{14} \\ \vdots & \vdots & & \vdots \\ \partial f_{14}/\partial X_1 & \partial f_{14}/\partial X_2 & \dots & \partial f_{14}/\partial X_{14} \end{pmatrix}$$

As it is well-known, see [28, 24], a steady state is asymptotically stable if and only if all eigenvalues of the Jacobian at the steady state have negative real parts. A

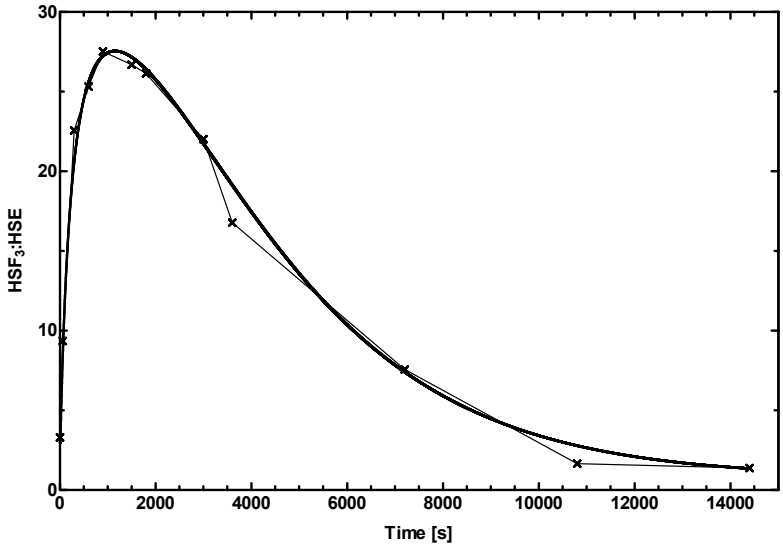

Figure 1: The continuous line shows a numerical estimation of function $X_4(t)$, standing for DNA binding, for the initial data in Table 3.2 and the parameter values in Table 3.2. With crossed points we indicated the experimental data of [10].

numerical estimation done with *Copasi* [8] shows that the steady state for $T = 42$, see Table 3.2, is indeed asymptotically stable.

4 Sensitivity analysis

Sensitivity analysis is a method to estimate the changes brought into the system through small changes in the parameters of the model. In this way one may estimate both the robustness of the model against small changes in the model, as well as identify possibilities for bringing a certain desired changed in the system. E.g., one question that is often asked of a biochemical model is what changes should be done to the model so that the new steady state satisfies certain properties. In our case we are interested in changing some of the parameters of the model so that the level of mfp in the new steady state of the system is smaller than in the standard model, thus presumably making it easier for the cell to cope with the heat shock. We also analyze a scenario in which we are interested in increasing the level of mfp in the new steady state, thus increasing the chances of the cell not being able to cope with the heat shock. Such a scenario is especially meaningful in relation with cancer cells that exhibit the properties of an excited cell, with increased levels of hsp, see [3, 15, 16, 27]. In this section we follow in part a presentation of sensitivity analysis due to [26].

We consider the partial derivatives of the solution of the system with respect to the parameters of the system. These are called *first-order local concentration sensitivity coefficients*. Second- or higher-order sensitivity analysis considering the simultaneous change of two or more parameters is also possible. If we denote $X(t, \kappa) = (X_1(t, \kappa), X_2(t, \kappa), \ldots, X_{14}(t, \kappa))$ the solution of the system (1)-(14) with respect to the parameter vector κ, then the concentration sensitivity coefficients are the time functions $\partial X_i / \partial \kappa_j(t)$, for all $1 \leq i \leq 14$, $1 \leq j \leq 17$. Differentiating the system (1)-(14)

with respect to κ_j yields the following set of *sensitivity equations*:

$$\frac{d}{dt}\frac{\partial X}{\kappa_j} = J(t)\frac{\partial X}{\partial \kappa_j} + \frac{\partial f(t)}{\partial \kappa_j}, \qquad \text{for all } 1 \leq j \leq 17, \tag{29}$$

where $\partial X/\partial\kappa_j = (\partial X_1/\partial\kappa_j, \ldots, \partial X_{14}/\partial\kappa_j)$ is the component-wise vector of partial derivatives, $f = (f_1, \ldots, f_{14})$ is the model function in (1)-(14), and $J(t)$ is the corresponding Jacobian. The initial condition for the system (29) is that $\partial X/\partial\kappa_j(0) = 0$, for all $1 \leq j \leq 17$.

The solution of the system (29) can be numerically integrated, thus obtaining a numerical approximation of the time evolution of the sensitivity coefficients. Very often however, the focus is on sensitivity analysis around steady states. If the considered steady state is asymptotically stable, then one may consider the limit

$$\lim_{t\to\infty}\left(\tfrac{\partial X}{\partial\kappa_j}\right)(t),$$

called *stationary sensitivity coefficients*. They reflect the dependency of the steady state on the parameters of the model. Mathematically, they are given by a set of algebraic equations obtained from (29) by setting $d/dt(\partial X/\kappa_j) = 0$. We then obtain the following algebraic equations:

$$\left(\tfrac{\partial X}{\partial\kappa_j}\right) = -J^{-1}F_j, \qquad \text{for all } 1 \leq j \leq 17, \tag{30}$$

where J is the value of the Jacobian at the steady state and F_j is the j-th column of the matrix $F = (\partial f_r/\partial\kappa_s)_{r,s}$ computed at the steady state.

When used for comparing the relative effect of a parameter change in two or more variables, the sensitivity coefficients must have the same physical dimension or be dimensionless, see [26]. Most often, one simply considers the matrix S' of (dimensionless) *normalized* (also called *scaled*) sensitivity coefficients:

$$S'_{ij} = \frac{\kappa_j}{X_i(t,\kappa)} \cdot \frac{\partial X_i(t,\kappa)}{\partial\kappa_j} = \frac{\partial \ln X_i(t,\kappa)}{\partial \ln \kappa_j}$$

Numerical estimations of the normalized sensitivity coefficients for a steady state may be obtained, e.g. with Copasi. For X_{14} (standing for the level of mfp in the model), the most significant (with the largest module) sensitivity coefficients are the following:

- $\partial\ln(X_{14})/\partial\ln(T) = 14.24$,
- $\partial\ln(X_{14})/\partial\ln(k_1^+) = -0.16$,
- $\partial\ln(X_{14})/\partial\ln(k_2^+) = -0.16$,
- $\partial\ln(X_{14})/\partial\ln(k_5^+) = 0.49$,
- $\partial\ln(X_{14})/\partial\ln(k_5^-) = -0.49$,
- $\partial\ln(X_{14})/\partial\ln(k_6) = 0.16$,
- $\partial\ln(X_{14})/\partial\ln(k_9) = 0.15$,
- $\partial\ln(X_{14})/\partial\ln(k_{11}^+) = -0.99$,
- $\partial\ln(X_{14})/\partial\ln(k_{11}^-) = 0.24$,
- $\partial\ln(X_{14})/\partial\ln(k_{12}) = -0.24$.

These coefficients being most significant is consistent with the biological intuition that the level of mfp in the model is most dependant on the temperature (parameter T), on the rate of mfp being sequestered by hsp (parameters k_{11}^+ and k_{11}^-) and the rate of protein refolding (parameter k_{12}). However, the sensitivity coefficients also reveal less intuitive, but significant dependencies such as the one on the reaction rate of hsf being sequestered by hsp (parameters k_5^+ and k_5^-), on the rate of dissipation of hsf dimers (parameter k_6), or on the rate of dimer- and trimer-formation (parameters k_1^+ and k_2^+).

Note that the sensitivity coefficients reflect the changes in the steady state for *small* changes in the parameter. E.g., increasing the temperature from 42 with 0.1% yields an increase in the level of mfp with 1.43%, roughly as predicted by $\partial\ln(X_{14})/\partial\ln(T) =$

14.24. An increase of the temperature from 42 with 10% yields however an increase in the level of mfp of 311.93%.

A similar sensitivity analysis may also be performed with respect to the initial conditions, see [26]. If we denote by $X^{(0)} = X(0, \kappa)$, the initial values of the vector X, for parameters κ, then the *initial concentration sensitivity coefficients* are obtained by differentiating system (1)-(14) with respect to $X^{(0)}$:

$$\frac{d}{dt}\frac{\partial X}{\partial X^{(0)}} = J(t)\frac{\partial X}{\partial X^{(0)}}(t), \tag{31}$$

with the initial condition that $\partial X/\partial X^{(0)}(0)$ is the identity matrix. It follows then that the initial concentration sensitivity matrix is given by the following matrix exponential:

$$\frac{\partial X}{\partial X^{(0)}}(t) = e^{J(t)} = \sum_{k=0}^{\top} \frac{J(t)^k}{k!}.$$

Similarly as for the parameter-based sensitivity coefficients, it is often useful to consider the normalized, dimensionless coefficients

$$\frac{\partial X_i}{\partial {X^{(0)}}_j}(t) \cdot \frac{{X^{(0)}}_j(t)}{X_i(t)} = \frac{\partial \ln(X_i)}{\partial \ln({X^{(0)}}_j)}.$$

A numerical estimation of the initial concentration sensitivity coefficient of mfp around the steady state given in Table 3.2 for $T = 42$, shows that all are negligible except for the following two coefficients: $\partial\ln(X_{14})/\partial\ln(X_9^{(0)}) = -0.497748$ and $\partial\ln(X_{14})/\partial\ln(X_{13}^{(0)}) = 0.99$. While the biological significance of the dependency of mfp on the initial level of prot is obvious, its dependency on the initial level of hsp: hsf is perhaps not. Moreover, it turns out that several other variables have a significant dependency on the initial level of hsp: hsf:

- $\partial\ln(X_1)/\partial\ln(X_9(0)) = 0.49$,
- $\partial\ln(X_2)/\partial\ln(X_9(0)) = 0.49$,
- $\partial\ln(X_3)/\partial\ln(X_9(0)) = 1.04$,
- $\partial\ln(X_4)/\partial\ln(X_9(0)) = 0.49$,
- $\partial\ln(X_{10})/\partial\ln(X_9(0)) = 0.49$,
- $\partial\ln(X_6)/\partial\ln(X_9(0)) = -0.04$,
- $\partial\ln(X_7)/\partial\ln(X_9(0)) = 0.49$,
- $\partial\ln(X_9)/\partial\ln(X_9(0)) = 0.99$,
- $\partial\ln(X_{14})/\partial\ln(X_9(0)) = -0.49$,
- $\partial\ln(X_{11})/\partial\ln(X_9(0)) = 0.49$,

E.g., increasing $X_9^{(0)}$ by 1% increases the steady state values of X_7 by 0.49% and decreases the level of X_{14} by 0.49%. Increasing $X_9^{(0)}$ by 10% increases the steady state values of X_7 by 4.85% and decreases the level of X_{14} by 4.63%.

The biological interpretation of this significant dependency of the model on the initial level of hsp: hsf is based on two arguments. On one hand, the most significant part (about two thirds) of the initial available molecules of hsp in our model are present in bonds with hsf. On the other hand, the vast majority of hsf molecules are initially bound to hsp. Thus, changes in the initial level of hsp: hsf have an immediate influence on the two main drivers of the heat shock response: hsp and hsf. Interestingly, the dependency of the model on the initial levels of either hsp or hsf is negligible.

Acknowledgments This work has been partially supported by the following grants from Academy of Finland: project 108421 and 203667 (to I.P.), the Center of Excellence on Formal Methods in Programming (to R-J.B.).

References

[1] Bruce Alberts, Alexander Johnson, Julian Lewis, Martin Raff, Keith Roberts, and Peter Walter. *Essential Cell Biology, 2nd edition*. Garland Science, 2004.

[2] R.L. Burden and J. Douglas Faires. *Numerical Analysis*. Thomson Brooks/Cole, 1996.

[3] Daniel R. Ciocca and Stuart K. Calderwood. Heat shock proteins in cancer: diagnostic, prognostic, predictive, and treatment implications. *Cell Stress and Chaperones*, 10(2):86–103, 2005.

[4] H. El-Samad, H. Kurata, J. Doyle, C.A. Gross, and M. Khamash. Surviving heat shock: control strategies for robustness and performance. *PNAS*, 102(8):2736–2741, 2005.

[5] H. El-Samad, S. Prajna, A. Papachristodoulu, M. Khamash, and J. Doyle. Model validation and robust stability analysis of the bacterial heat shock response using sostools. In *Proceedings of the 42nd IEEE Conference on Decision and Control*, pages 3766–3741, 2003.

[6] C.M. Guldberg and P. Waage. Studies concerning affinity. *C. M. Forhandlinger: Videnskabs-Selskabet i Christiana*, 35, 1864.

[7] C.M. Guldberg and P. Waage. Concerning chemical affinity. *Erdmann's Journal fr Practische Chemie*, 127:69–114, 1879.

[8] Stefan Hoops, Sven Sahle, Ralph Gauges, Christine Lee, Jrgen Pahle, Natalia Simus, Mudita Singhal, Liang Xu, Pedro Mendes, and Ursula Kummer. Copasi – a COmplex PAthway SImulator. *Bioinformatics*, 22(24):3067–3074, 2006.

[9] Harm K. Kampinga. Thermotolerance in mammalian cells: protein denaturation and aggregation, and stress proteins. *J. Cell Science*, 104:11–17, 1993.

[10] Michael P. Kline and Richard I. Morimoto. Repression of the heat shock factor 1 transcriptional activation domain is modulated by constitutive phosphorylation. *Molecular and Cellular Biology*, 17(4):2107–2115, 1997.

[11] E. Klipp, R. Herwig, A. Kowald, C. Wierling, and H. Lehrach. *Systems Biology in Practice*. Wiley-VCH, 2006.

[12] H. Kurata, H. El-Samad, T.M. Yi, M. Khamash, and J. Doyle. Feedback regulation of the heat shock response in e.coli. In *Proceedings of the 40th IEEE Conference on Decision and Control*, pages 837–842, 2001.

[13] James R. Lepock, Harold E. Frey, and Kenneth P. Ritchie. Protein denaturation in intact hepatocytes and isolated cellular organelles during heat shock. *The Journal of Cell Biology*, 122(6):1267–1276, 1993.

[14] James R. Lepock, Harold E. Frey, A. Michael Rodahl, and Jack Kruuv. Thermal analysis of chl v79 cells using differential scanning calorimetry: Implications for hyperthermic cell killing and the heat shock response. *Journal of Cellular Physiology*, 137(1):14–24, 1988.

[15] Bei Liu, Anna M. DeFilippo, and Zihai Li. Overcomming immune toerance to cancer by heat shock protein vaccines. *Molecular cancer therapeutics*, 1:1147–1151, 2002.

[16] Katalin V. Lukacs, Olivier E. Pardo, M.Jo Colston, Duncan M. Geddes, and Eric WFW Alton. Heat shock proteins in cancer therapy. In Habib, editor, *Cancer Gene Therapy: Past Achievements and Future Challenges*, pages 363–368. 2000.

[17] David L. Nelson and Michael M. Cox. *Principles of Biochemistry, 3rd edition*. Worth Publishers, 2000.

[18] A. Peper, C.A. Grimbergent, J.A.E. Spaan, J.E.M. Souren, and R. van Wijk. A mathematical model of the hsp70 regulation in the cell. *Int. J. Hyperthermia*, 14:97–124, 1997.

[19] Ion Petre, Claire L. Hyder, Andrzej Mizera, Andrey Mikhailov, John E. Eriksson, Lea Sistonen, and Ralph-Johan Back. Two metabolites are enough to drive the eukaryotic heat shock response.

[20] A. Graham Pockley. Heat shock proteins as regulators of the immune response. *The Lancet*, 362(9382):469–476, 2003.

[21] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flammery. *Numerical Recipes: The Art of Scientific Computing*. Cambridge Universoty Press, 2007.

[22] Theodore R. Rieger, Richard I. Morimoto, and Vassily Hatzimanikatis. Mathematical modeling of the eukaryotic heat shock response: Dynamics of the hsp70 promoter. *Biophysical Journal*, 88(3):1646–58, 2005.

[23] R. Srivastava, M.S. Peterson, and W.E. Bentley. Stochastic kinetic analysis of the escherichia coli stres circuit using σ^{32}-targeted antisense. *Biotechnology and Bioengineering*, 75(1):120–129, 2001.

[24] Clifford Henry Taubes. *Modeling Differential Equations in Biology*. Cambridge University Press, 2001.

[25] Claire J. Tomlin and Jeffrey D. Axelrod. Understanding biology by reverse engineering the control. *PNAS*, 102(12):4219–4220, 2005.

[26] Tamás Turányi. Sensitivity analysis of complex kinetic systems. tools and applications. *Journal of Mathematical Chemistry*, 5:203–248, 1990.

[27] Paul Workman and Emmanuel de Billy. Putting the heat on cancer. *Nature Medicine*, 13(12):1415–1417, 2007.

[28] Dennis G. Zill. *A First Course in Differential Equations*. Thomson, 2001.

[29] Dennis G. Zill. *A First Course in Differential Equations with Modeling Applications*. Thomson, 2005.

A Petri-net Formalization of Heat Shock Response Model

Ralph-Johan Back
backj@abo.fi

Tseren-Onolt Ishdorj
tishdorj@abo.fi

Ion Petre
ipetre@abo.fi

Department of Information Technologies
Åbo Akademi University, Turku 20520, Finland

Abstract

A differential equation-based mathematical model of the heat shock response has been introduced in [7] and discussed further in [8]. We discuss in this paper a Petri-net-based model and compute its P- and T-invariants. We also give several results concerning the boundedness and the deadlock of the Petri-net model. Finally, we briefly compare the Petri-net model with the continuous model of [8].

1 The heat shock response

The heat shock response is the reaction of cells to elevated temperatures. Under raised temperature (or other stress stimuli such as heavy metals or radiation), proteins tend to misfold and then form big aggregates that may eventually render the cell unable to survive, see [2]. It is well understood that the main role in the cell's reaction to heat shock is played by the heat shock proteins (HSP), see [9, 3]. They act as chaperons, helping misfolded proteins (MFP) to refold into their native form (PROT). The heat shock proteins have a major contribution also in the resilience of cancer cells, see [1] and they have been suggested as targets in potential cancer treatments, see [5, 13].

In eukaryotes, the heat shock response is controlled through the regulation of the transactivation of the HSP-encoding genes, see [3, 10] (the bacterial mechanisms is slightly different, see [11]). The kinetic details of the control have been disputed in the past few years, with several models proposed in [6, 10, 4]. We follow in this paper a new kinetic model for the heat shock response, recently proposed in [7]. In this model, the transcription of the gene is promoted by some proteins called heat shock factors (HSF) that trimerize and then bind to a specific DNA sequence called heat

shock element (HSE), upstream of the HSP-encoding gene. Once the HSF trimer is bound to the heat shock element, the gene is transactivated and the synthesis of HSP is thus switched on. Once the level of HSP is high enough, the cell has an ingenious mechanism to switch off its own synthesis. For this, HSP bind to free HSF, as well as break the HSF trimers (including those bound to HSE, promoting the gene activation), thus effectively halting the HSP synthesis. In this model we treat uniformly under HSP all types of heat shock proteins. We have a similar convention for treating uniformly all three types of heat shock factors under the name HSF. PROT and MFP group together all proteins and misfolded proteins, respectively, other than HSP and HSF. Table 1 summarizes the list of reactions in the kinetic model of [7]. In there we list each reversible reaction as two irreversible ones, accounting for its two directions.

Metabolites/Places		Reactions/Transitions	
p_1:	HSE	t_1:	$2HSF \to HSF_2$
p_2:	HSF	t_2:	$HSF_2 \to 2HSF$
p_3:	$HSP{:}HSF$	t_3:	$HSF + HSF_2 \to HSF_3$
p_4:	HSF_2	t_4:	$HSF_3 \to HSF_2 + HSF$
p_5:	HSF_3	t_5:	$HSF_3 + HSE \to HSF_3{:}HSE$
p_6:	$HSF_3{:}HSE$	t_6:	$HSF_3{:}HSE \to HSF_3 + HSE$
p_7:	HSP	t_7:	$HSF_3{:}HSE \to HSF_3{:}HSE + HSP$
p_8:	$HSP{:}MFP$	t_8:	$HSP + HSF_3{:}HSE \to HSP{:}HSF + 2HSF + HSE$
p_9:	MFP	t_9:	$HSP + HSF \to HSP{:}HSF$
p_{10}:	$PROT$	t_{10}:	$HSP{:}HSF \to HSP + HSF$
		t_{11}:	$HSP + HSF_2 \to HSP{:}HSF + HSF$
		t_{12}:	$HSP + HSF_3 \to HSP{:}HSF + 2HSF$
		t_{13}:	$PROT \to MFP$
		t_{14}:	$HSP + MFP \to HSP{:}MFP$
		t_{15}:	$HSP{:}MFP \to HSP + MFP$
		t_{16}:	$HSP{:}MFP \to HSP + PROT$
		t_{17}:	$HSP \to 0$

Table 1: The metabolites and reactions in the molecular model of the heat shock response of [7]. They are modeled as places and transitions, respectively in our Petri-net approach.

2 Petri-nets

Consider a Petri-net with the set of places $P = \{p_1, \cdots, p_n\}$ and set of transitions $T = \{t_1, \cdots, t_m\}$, for some $m, n \geq 0$. Its *incidence matrix* C is an $(n \times m)$- matrix (where n denotes the number of places and m the number of transitions). Every matrix entry c_{ij} gives the token change on the place p_i by the firing of the transition t_j. Thus, firing transition t_j changes the state of the system from $S \in \mathbb{N}_0^n$ to state $C_j + S$, where C_j is the j-th column of C. The transition may fire if and only if all entries of $C_j + S$ are nonnegative integers. If several transitions may fire at any given time, one is chosen nondeterministically.

A T-invariant is defined as a non-zero vector $x \in \mathbb{N}_0^m$, which holds the equation $C \cdot x = 0$. A T-invariant represents a multiset of transitions, which have altogether a zero effect on the marking.

Analogously, a P-invariant is defined as a non-zero vector $y \in \mathbb{N}_0^n$ such that $y^t \cdot$

$C = 0$, where y^t is the row-vector transpose of y. A P-invariant characterizes a token conservation rule for a set of places, over which the weighted sum of tokens is constant independently from any firing.

A net is covered by T-invariants (P-invariants), if every transition (place) participates in a T-invariant (P-invariant).

3 Modeling method

Each metabolite in the molecular model is represented as a place in the Petri-net, labeled by $p_1, p_2, \ldots, p_{10}$ as indicated in Table 1. The reactions in the model are represented as transitions injectively labeled by $t_1, t_2, \ldots, t_{17}$, see Table 1.

The 17 molecular reactions can be classified in 8 basic types of reactions:

(i) $A + A \to B$; (ii) $A + B \to C$; (iii) $A + B \to C + D$; (iv) $A \to B + C$;
(v) $A \to A + B$; (vi) $A \to B$; (vii) $A \to 0$; (viii) $A \to B + B$.

We illustrate in Figure 1 each type (i)–(viii) with reactions from our model. When composing the Petri-net components corresponding to all reactions (by merging the places with identical labels), we obtain the Petri-net model in Figure 2.

The tokens in a place represents the number of copies of the corresponding metabolite existing in the model at the time.

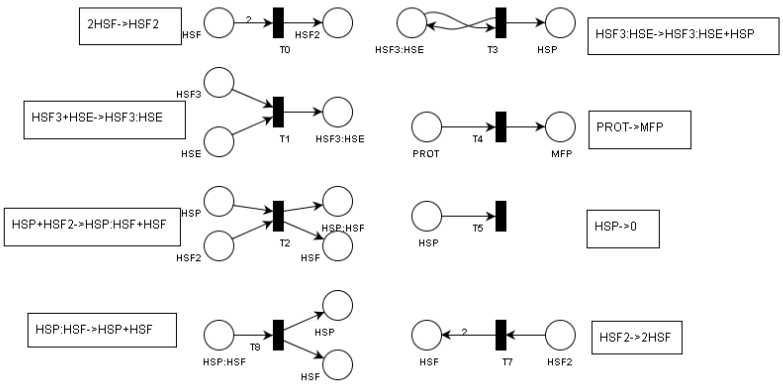

Figure 1: Petri-net components for HSR model.

The Petri-net model $\mathcal{P}$ of the heat shock response, obtained by composing the blocks corresponding to each reaction in Table 1, is shown in Figure 2. The net consists of 10 places and 17 transitions, which are listed by their ID and biological reactions in Table 1. The net structure consists of two parts connected to each other by the place HSP. The first part is devoted to the back-regulation if the HSP transactivation and it is the dominant part of the model under physiological conditions (at temperature 37C). The second part is devoted to the misfolding of proteins and the chaperone activity of HSP, whose activity is greatly increased under raised temperature. The incidence matrix of the Petri-net model $\mathcal{P}$ is in Table 2.

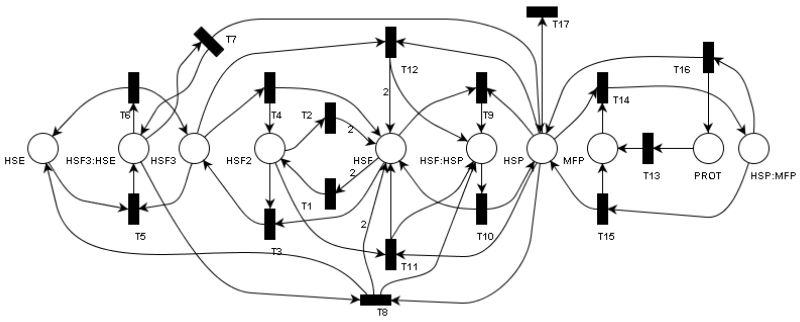

Figure 2: A Petri-net of the HSR.

	t_1	t_{17}	t_2	t_9	t_7	t_5	t_{10}	t_4	t_6	t_3	t_{11}	t_{12}	t_8	t_{13}	t_{14}	t_{15}	t_{16}
p_1	0	0	0	0	0	−1	0	0	1	0	0	0	1	0	0	0	0
p_2	−2	0	2	−1	0	0	1	1	0	−1	1	2	2	0	0	0	0
p_3	0	0	0	1	0	0	−1	0	0	0	1	1	1	0	0	0	0
p_4	1	0	−1	0	0	0	0	1	0	−1	−1	0	0	0	0	0	0
p_5	0	0	0	0	0	−1	0	−1	1	1	0	−1	0	0	0	0	0
p_6	0	0	0	0	0	1	0	0	−1	0	0	0	−1	0	0	0	0
p_7	0	−1	0	−1	1	0	1	0	0	0	−1	−1	−1	0	−1	1	1
p_8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	−1	−1
p_9	0	0	0	0	0	0	0	0	0	0	0	0	0	1	−1	1	0
p_{10}	0	0	0	0	0	0	0	0	0	0	0	0	0	−1	0	0	1

Table 2: The incidence matrix of the Petri-net depicted in Figure 2.

4 Analysis of the Petri-Net and its biological interpretations

In this section, we calculate the P-invariants and the T-invariants of the Petri-net (Figure 2). Based on the invariants, certain analysis for the heat shock response model behavior will then be given.

To calculate the P-invariants of our model, we solve the system $x \cdot C = 0$ over nonnegative integers, where C is the incidence matrix of the model and $x \in \mathbb{N}_0^{10}$. We obtain the following three independent solutions:

	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}
$x' =$	(1	0	0	0	0	1	0	0	0	0)
$x'' =$	(0	1	1	2	3	3	0	0	0	0)
$x''' =$	(0	0	0	0	0	0	0	1	1	1)

Based on the above vectors, three P-invariant equations of the heat shock response model are written:

$$S(HSE) + S(HSF_3{:}HSE) = K_1, \tag{1}$$

$$S(HSF) + 2S(HSF_2) + 3S(HSF_3) + S(HSP{:}HSF) + 3S(HSF_3{:}HSE) = K_2, \tag{2}$$

$$S(HSP{:}MFP) + S(MFP) + S(PROT) = K_3, \tag{3}$$

for some constants $K_1, K_2, K_3 \geq 0$ and for any state S of the Petri-net. Here we denote by $S(X)$ the number of tokens in place X in the state S.

The first invariant says that the total number of heat shock elements in the model, either free or bound to HSF3, is constant. The second invariant tells that the total amount of heat shock factors, in their various forms, is also constant. The third invariant shows that the total amount of proteins other than HSP and HSF, either correctly folded, or misfolded, is also constant. All these invariants have an intuitive biological interpretation. The first one is evident: since the heat shock elements are specific regions of DNA, their total number is clearly constant. As far as the second one goes, it is clear from the list of reactions in Table 1 that HSFs are neither synthesized, nor degraded. Rather, they participate in various reactions, forming bonds with various metabolites. The third invariant is evident for the same reasons: neither $PROT$, nor MFP is either synthesized, or degraded.

We can calculate T-invariants by considering the system $C \cdot x = 0$ over nonnegative integers. We obtain the following solution, written by indicating the reactions for each T-invariant:

$$(2HSF \to HSF_2) + (HSF{:}HSP \to HSF + HSP) + (HSP + HSF_2 \to HSF{:}HSP + HSF) \quad (4)$$

$$(2HSF \to HSF_2) + (HSF{:}HSP \to HSF + HSP) + (HSP + HSF_3 \to HSF{:}HSP + 2HSF) + (HSF + HSF_2 \to HSF_3) \quad (5)$$

$$(2HSF \to HSF_2) + (HSF_3 + HSE \to HSF_3{:}HSE) + (HSF{:}HSP \to HSP + HSF) + (HSF + HSF_2 \to HSF3) + (HSP + HSF_3{:}HSE \to HSF{:}HSP + 2HSF + HSE) \quad (6)$$

$$(HSP \to 0) + (HSF_3{:}HSE \to HSF_3{:}HSE + HSP) \quad (7)$$

$$(PROT \to MFP) + (HSP + MFP \to HSP{:}MFP) + (HSP{:}MFP \to HSP + PROT) \quad (8)$$

$$(HSP + MFP \to HSP{:}MFP) + (HSP{:}MFP \to HSP + MFP) \quad (9)$$

$$(2HSF \to HSF_2) + (HSF_2 \to 2HSF) \quad (10)$$

$$(HSP + HSF \to HSF{:}HSP) + (HSF{:}HSP \to HSP + HSF) \quad (11)$$

$$(HSF + HSF_2 \to HSF_3) + (HSF_3 \to HSF_2 + HSF) \quad (12)$$

$$(HSF_3 + HSE \to HSF_3{:}HSE) + (HSF_3{:}HSE \to HSF_3 + HSE) \quad (13)$$

The T-invariants (9) - (13) are trivial, indicating the two directions of reversible reactions. Invariant (4) is the HSP – HSF_2 capture cycle; invariant (5) is the HSP – HSF_3 capture cycle. Invariant (6) is the main cycle, with HSF_3 binding to HSE and HSP freeing HSF. Invariant (7) says that the only way to compensate for degraded HSP is by translating it from genes. Invariant (8) is the chaperone activity cycle: proteins get misfolded, HSP binds to them and then releases them as correctly folded proteins.

Consider now the reachability problem for our Petri-net. As an example, let the initial marking be $S_0 = (1, 3, 0, 0, 0, 0, 0, 0, 0, 1)$. In this case, all transitions are eventually enabled, while the network is not bounded. It can be seen that S_0 is a minimal initial marking with this property. In this case the reachability graph is infinite, while the coverability graph consists of 48 nodes. As another example, let the initial marking be $S_0' = (1, 1, 0, 0, 0, 0, 1, 0, 1, 1)$. In this case the net is bounded. There are only 10 markings x that are reachable from S_0', see Table 3 and Figure 3.

	HSE	HSF	HSP:HSF	HSF_2, HSF_3, HSF:HSE	HSP	HSP:MFP	MFP	$PROT$	$\#M$
S_0	HSE	HSF	–	–	HSP	–	MFP	$PROT$	5
S_1	HSE	HSF	–	–	–	HSP:MFP	–	$PROT$	4
S_2	HSE	HSF	–	–	HSP	–	$2MFP$	–	5
S_3	HSE	–	HSP:HSF	–	–	–	MFP	$PROT$	4
S_4	HSE	HSF	–	–	–	–	MFP	$PROT$	4
S_5	HSE	HSF	–	–	HSP	–	–	$2PROT$	5
S_6	HSE	HSF	–	–	–	HSP:MFP	MFP	–	4
S_7	HSE	–	HSP:HSF	–	–	–	$2MFP$	–	4
S_8	HSE	HSF	HSP:HSF	–	–	–	$2MFP$	–	5
S_9	HSE	–	HSP:HSF	–	–	–		$2PROT$	4
S_{10}	HSE	HSF	–	–	–	–		$2PROT$	4

Table 3: Reachable markings within the initial marking $S_0 = (1, 1, 0, 0, 0, 0, 1, 0, 1, 1)$.

The following two results give some results about the reachability problem of our Petri-net and about its possible deadlocks. The first result shows that a deadlock is characterized by the P-invariant given at (2).

Theorem 4.1. *The Petri-net $\mathcal{P}$ modeling the heat shock response may reach a deadlock starting from the initial marking S if and only if $S(HSF) + 2S(HSF_2) + 3S(HSF_3) + S(HSF{:}HSP) + 3S(HSF_3{:}HSE) \leq 1$. Equivalently, $S(HSF_2) = S(HSF_3) = S(HSF_3{:}HSE) = 0$ and $S(HSF) + S(HSF{:}HSP) \leq 1$.*

Proof. Assume first an initial marking $S = (a, n_1, n_2, 0, 0, 0, b, c, d, e)$, with $a, b, c, d, e, n_1, n_2 \in \mathbb{N}$, $n_1 + n_2 = 1$. The following sequence of transitions leads to deadlock:

$$\begin{aligned} S \quad &\longrightarrow^{t_{13}^{e}} (a, n_1, n_2, 0, 0, 0, b, c, d + e, 0) \\ &\longrightarrow^{t_{15}^{c}} (a, n_1, n_2, 0, 0, 0, b + c, 0, c + d + e, 0) \\ &\longrightarrow^{t_{10}^{n_2}} (a, 1, 0, 0, 0, 0, b + c + n_1, 0, c + d + e, 0) \\ &\longrightarrow^{t_{17}^{b+c+n_1}} (a, 1, 0, 0, 0, 0, 0, 0, c + d + e, 0). \end{aligned}$$

Assume now an initial marking S from where $\mathcal{P}$ may reach a deadlock. Note that $P = S(HSF) + 2S(HSF_2) + 3S(HSF_3) + S(HSF{:}HSP) + 3S(HSF_3{:}HSE)$ is a P-invariant of the Petri-net and so, constant throughout the transitions of the Petri-net. To conclude the theorem, it is enough to proof that if $p \geq 2$ then at least one transition is applicable to S. Assume then that $p \geq 2$. If $S(HSF_2) \geq 1$, or $S(HSF_3{:}HSE) \geq 1$, then transitions t_2, t_4, t_{10} and t_6 are applicable to S, respectively. On the other hand, if $S(HSF_2) + S(HSF_3) + S(HSF{:}HSP) + S(HSF_3{:}HSE) = 0$, then $S(HSF) \geq 2$ and so, t_2 is applicable to S. □

Intuitively, Theorem 4.1 shows that, given enough HSFs, the network runs indefinitely, albeit it may run through only a finite number of states. The case where the net runs through an infinite number of states is described in Theorem 4.2.

Our second result relates the reachability problem to the P-invariants (1) and (2).

Theorem 4.2. *The following conditions are equivalent:*

(i) the reachability graph of the Petri-net $\mathcal{P}$ modeling the heat shock response is infinite when starting from the initial marking S;

(ii) the place HSP is not bounded when starting from the initial marking S;

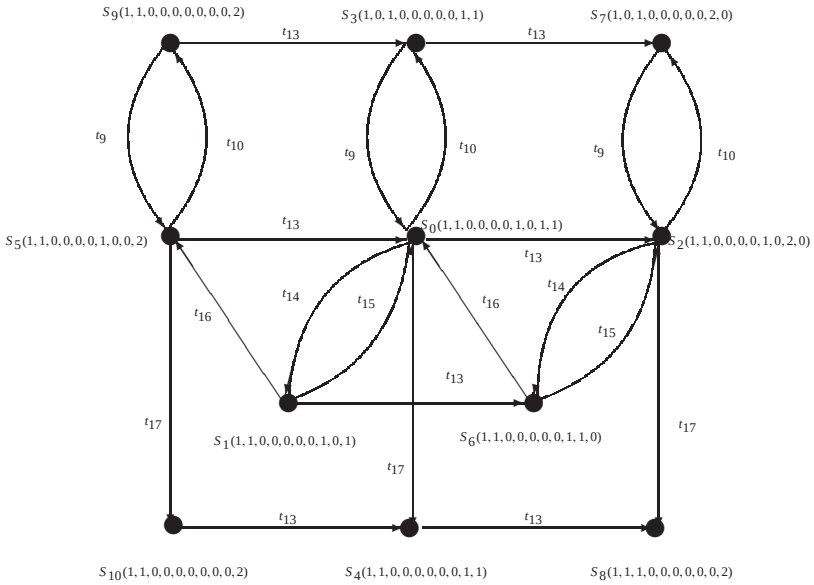

Figure 3: Reachability graph from the marking S_0.

(iii) transition t_7 is eventually enabled when starting from the initial marking S;

(iv) $S(HSE) + S(HSF_3{:}HSE) \geq 1$ *and* $S(HSF) + 2S(HSF_2) + 3S(HSF_3) + S(HSF{:}HSP) + 3S(HSF_3{:}HSE) \geq 3$.

Proof. (*i*) $\Leftrightarrow$ (*ii*). If the reachability graph of $\mathcal{P}$ is infinite, there must exist at least a place which is unbounded. Based on the P-invariant (1)–(3), all places except HSP are bounded. The reverse implication is obvious.

(*ii*) $\Leftrightarrow$ (*iii*). If HSP is unbounded, then there must exists at least a transition which is involved to provide infinitely many tokens into HSP. In our case, transition t_7 plays in this role. Conversely, the place HSP receives infinitely many tokens as long as t_7 fires infinitely.

(*iii*) $\Leftrightarrow$ (*iv*). If t_7 is enabled in state S', then $S'(HSF_3{:}HSE) \geq 1$. Then (*iv*) follows based on the P-invariant (1) and (2).

For reverse direction, if $S(HSF_3{:}HSE) \geq 1$, then t_7 is enabled in S. If not, then $S(HSE) \geq 1$ and $S(HSF) + 2S(HSF_2) + 3S(HSF_3) + S(HSF{:}HSP) \geq 3$. If $S(HSF_3) \geq 1$, then t_7 will be enabled after firing t_5 first. Otherwise, we obtain that $S(HSF) + 2S(HSF_2) + S(HSF{:}HSP) \geq 3$. If $S(HSF_2) \geq 2$, then t_7 gets enabled after firing t_2, t_3 and t_5 (which are all enabled when fired). If $S(HSF_2) = 1$, then $S(HSF) + S(HSF{:}HSP) \geq 1$. Thus, t_3 is either enabled in S, or gets enabled after firing t_{10} in S. Firing t_3 and then t_5 yields a state where t_7 is enabled. If $S(HSF_2) = 0$, then $S(HSF) + S(HSF{:}HSP) \geq 3$. With a discussion similar as above, we notice that we may reach a state with $S(HSF_3) \geq 1$ after which, firing t_5 yields a state where t_7 is enabled.

□

Intuitively, Theorem 4.2 shows that in order to have the network run as expected (potentially run through an infinite number of states), the basic requirement is to have

at least one heat shock element in either of its two forms and at least three heat shock factors, in either of their possible forms.

Corollary 4.3. *The reachability graph of the net $\mathcal{P}$ is bounded if and only if*

$$S(HSF_3{:}HSE) = 0$$

and either $S(HSE) = 0$ or $S(HSF) + 2S(HSF_2) + 3S(HSF_3) + S(HSF{:}HSP) \leq 2$.

Proof. The reachability graph of $\mathcal{P}$ is bounded if place HSP is bounded. HSP is bounded iff $S(HSF_3{:}HSE) = 0$, in other words t_7 is disabled, and also t_5 is disabled. Conversely, if t_7 is disabled, HSP is bounded, thus, reachability graph is bounded. □

5 Conclusion

The invariants of the Petri-net model correspond to properties of the continuous model of [8]: the P-invariants correspond to the mass-conservation relations and the T-invariants correspond to the elementary modes. This relation follows from the fact that the incidence matrix of the Petri-net model coincides with the stoichiometric matrix of the continuous model and has been reported many times before, see, e.g., [12]. The types of analysis one can perform with the two approaches are however completely different. While the continuous model gives interesting steady state analysis, including sensitivity analysis, the Petri-net allows reasoning about the network itself, albeit in qualitative, rather than quantitative terms. E.g., we gave in Theorem 4.1 a simple condition for the network to run indefinitely, regardless of the transitions to be fired along any path. Similarly, we showed in Theorem 4.2 that the model may run through an infinite number of states only by firing transition t_7 infinitely many times, should it ever become enabled.

Acknowledgments This work has been partially supported by the grants 108421 and 203667 from Academy of Finland.

References

[1] Daniel R. Ciocca and Stuart K. Calderwood. Heat shock proteins in cancer: diagnostic, prognostic, predictive, and treatment implications. *Cell Stress and Chaperones*, 10(2):86–103, 2005.

[2] Harm K. Kampinga. Thermotolerance in mammalian cells: protein denaturation and aggregation, and stress proteins. *J. Cell Science*, 104:11–17, 1993.

[3] Michael P. Kline and Richard I. Morimoto. Repression of the heat shock factor 1 transcriptional activation domain is modulated by constitutive phosphorylation. *Molecular and Cellular Biology*, 17(4):2107–2115, 1997.

[4] O. Lipan, J-M. Navenot, Z.Wang, L. Huang, and S.C. Peiper. Heat shock response in cho mammalian cells is controlled by a nonlinear stochastic process. *PLoS Computational Biology*, 3(10):1859–1870, 2007.

[5] Katalin V. Lukacs, Olivier E. Pardo, M.Jo Colston, Duncan M. Geddes, and Eric WFW Alton. Heat shock proteins in cancer therapy. In Habib, editor, *Cancer Gene Therapy: Past Achievements and Future Challenges*, pages 363–368. Kluwer, 2000.

[6] A. Peper, C.A. Grimbergent, J.A.E. Spaan, J.E.M. Souren, and R. van Wijk. A mathematical model of the hsp70 regulation in the cell. *Int. J. Hyperthermia*, 14:97–124, 1997.

[7] Ion Petre, Claire L. Hyder, Andrzej Mizera, Andrey Mikhailov, John E. Eriksson, Lea Sistonen, and Ralph-Johan Back. Two metabolites are enough to drive the eukaryotic heat shock response. *manuscript*, 2008.

[8] Ion Petre, Andrzej Mizera, Claire L. Hyder, Andrey Mikhailov, John E. Eriksson, Lea Sistonen, and Ralph-Johan Back. A new mathematical model for the heat shock response. In Joost Kok, editor, *Algorithmic Bioprocesses*. Springer, 2008.

[9] A. Graham Pockley. Heat shock proteins as regulators of the immune response. *The Lancet*, 362(9382):469–476, 2003.

[10] Theodore R. Rieger, Richard I. Morimoto, and Vassily Hatzimanikatis. Mathematical modeling of the eukaryotic heat shock response: Dynamics of the hsp70 promoter. *Biophysical Journal*, 88(3):1646–58, 2005.

[11] R. Srivastava, M.S. Peterson, and W.E. Bentley. Stochastic kinetic analysis of the escherichia coli stres circuit using σ^{32}-targeted antisense. *Biotechnology and Bioengineering*, 75(1):120–129, 2001.

[12] K. Voss, M. Heiner, and I. Koch. Steady state analysis of metabolic pathways using petri nets. *Journal In Silico Biology*, 3(0031), 2003.

[13] Paul Workman and Emmanuel de Billy. Putting the heat on cancer. *Nature Medicine*, 13(12):1415–1417, 2007.

The Semiotic Perspective in the Study of Cell

Solomon Marcus
solomon.marcus@imar.ro

Institute of Mathematics, Academy of Romania

What is semiotics?

It is the study of sign processes. A sign can be understood in various ways. There is a binary representation of a semiotic system as a couple formed by a signifier and a signified (a sign and its meaning). There is also a triadic representation of a semiotic system, in its modern form being conceived as a sign, its object and its interpretant. For the first view, we can mention Ferdinand de Saussure (beginning of the XX-th century); the author of the second view is Charles Sanders Peirce, one of the most important American mathematicians of the second half of the XIX-th century. He is also considered as the founder of modern semiotics, but the roots of the study of sign processes can be observed already in the Greek antiquity, then in the Middle Age, then in the period of the XVII-th and XVIII-th centuries (John Locke, W. G. Leibniz etc). In respect to the binary view, the word 'horse' and its meaning is a sign system. In the ternary view, the corresponding semiotic system is given, roughly speaking, by the word 'horse', its object, represented by the animals called horses, and its interpretant, the meaning of the word 'horse'.

Concomitantly with the researchers having deliberately as their object of study the sign processes, there are also the so-called "semioticians a la Jourdain", i.e., those authors who are doing semiotics without to be aware of this fact. This situation occurs frequently and we can conjecture that the number of such implicit semioticians is larger than the number of those who deliberately are involved in semiotic studies.

The emergence of biosemiotics

Sign processes occur everywhere in the living universe and, according to some authors, they occur also in the inert universe. Implicitly, i.e., "a la Jourdain", many authors, in various periods of the history, have done semiotic research concerning living non-human beings. The first author who deliberately made a project for the investigation of sign processes in animals was Thomas A. Sebeok, the founder of 'zoosemiotics'. In a further step, he extended his project to all living beings and coined in ths respect the term 'biosemiotics'. All these events occurred in the second half of the past century. But it is important to identify those scientists who, despite their ignorance in semiotics, obtained results having a semiotic significance. Sebeok did a lot of work in this respect. For example, he pointed out the huge semiotic significance of the work done in the first

half of the past century by the German biologist Jakob von Uexkull, by his concept of Umwelt, a new way to understand the subjective surrounding.

The Darwinian tree-model in language evolution and in DNA evolution

During a long period, historical linguistics adopted the Darwinian model, proposed by August Schleicher(1863) as a guiding metaphor, according to the biological foundations of human language. The 'tree-model' became an explanation of the language evolution. Then it was the wave model, proposed by J.Schmidt (1872): a change spreads through a language in the same way as a stone sends ripples across a pool. Recently, P.Forster(1997) succeeded to bridge these two viewpoints, by means of a geometric network model, previously used for reconstructing DNA evolution; the same model was applied to vocabulary lists of closely related language. Forster starts with the remark that it would be desirable to visualize both tree aspects and wave aspects of language evolution is a single diagram and observes that this type of problem is perfectly tailored to network methods originally developed for reconstructing phylogenetic relationships from DNA sequences. During evolution, a given DNA sequence acquires mutations at random positions, causing the progeny sequences to become more and more dissimilar from one another and from their ancestral sequence as time passes, yielding the tree-like aspect of DNA evolution.

From the biological perspective about language to the linguistic perspective in molecular biology

Already in the preceding section, the solidarity between language and DNA clearly appeared. Towards the middle of the past century, the relevance of the linguistic perspective in molecular biology became stronger and stronger. Linguistics became a guide for biology, mainly in respect to the new discoveries in the field of heredity. Under the leadership of Roman Jakobson, a lot of linguistic metaphors used in molecular biology (letters, words, alphabet, grammar, dictionary, code, meaning etc) became object of investigation, in order to test their legitimacy beyond their metaphorical status. I did a synthesis and a continuation of the achievements in this direction in the first part of my "Linguistic structures and generative devices in molecular biology" (1974) and I proposed the following representation: there is a genetic language; DNA and RNA define the two strata of its chemical part, that could be considered the syntactic level, whose phonemes are the four types of nucleotide bases and whose morphemes are the 64 types of codons. There is a semantic level, defined by the biological part, having in its turn two strata: the amino acids and the proteins. DNAs are words over the alphabet of the four types of nucleotides, while proteins are words over the alphabet of the 20 types of amino acids. The so-caled genetic code is a dictionary putting in correspondence the different types of codons with the different types of amino acids. This correspondence is not devoid of synonymy and homonymy phenomena, although they are here far poorer than in natural languages. The phonemic status of nucleotide bases is in details legitimated, as well as the morphemic status of codons. There is also a DNA equivalent to the so-called duality of patterning principle ("la double articulation" introduced by Andre Martinet): DNAs are organized in two levels: the level of some minimally meaningful units, the codons, which, in their turn, are decomposable in some meaningless units, the nucleotide bases. Here, 'meaningful' and 'meaningless' mean 'endowed with', respectively 'devoid of' biological meaning. The number of the meaningless units is much inferior to the number of the meaningful units (here, 4 to

64). The analogy between natural language and genetic language goes deeper, but we cannot develop more here.

The formal grammar approach to DNA

The first steps in approaching DNA-proteins interaction by means of formal grammars were made by Z. Pawlak, B. Vauquois and myself (Marcus 1974). Pawlak used some dependency grammars, only sketched; starting from them, I proposed to Vauquois to transform Pawlak's device into a Chomskian grammar. He obtained a context-free grammar including 50 rules. This happened at the Interdisciplinary Seminar I organized during the Linguistic Institute of America, Buffalo, New York, July-August 1971). However, protein formation is not obtained by means of this grammar; we need to direct attention towards the language of derivations in the respective grammar. It was already known that the language of derivations in a context free grammar is a context sensitive grammar which may not be context free. This is just the case with protein formation. So, we could say that the grammar of proteins is like the natural languages: somewhere between context free and context sensitive. Let us observe that this was the second event related to the relevance of formal grammars in the biology of the human body. Chronologically, the first example in this respect was obtained by W.S. McCulloch and E. Pitts (1943), with a logical calculus(that could be equivalent to a grammatical device) of ideas involved in the activity of nervous systems; the second example related to the nervous system was given by S.C. Kleene(1956)and it was concerned with the representation of events in nerve nets and finite automata (proved to be equivalent to regular grammars).

This methodological similarity between the nervous system and the molecular level of the human existence deserves attention.

Linguistics, a common denominator of interest for computer science, molecular biology and semiotics

Formal grammars have their starting point in the generative approach to natural languages, as it was initiated by Noam Chomsky (1956, 1957). In this way, linguistics belongs to the foundations of computer science, because the syntax and the semantics of programming languages are studied by means of formal grammars. As it was pointed out in the previous sections, molecular biology takes profit from linguistics and from formal grammars, because at all levels (of DNA, of RNA and of proteins) it displays some sequential structures over some finite alphabets, showing strong architectural similarities with natural languages. Linguistics is historically and structurally related to semiotics. One of the roads to semiotics, the binary one, has as one of its main representatives the prominent linguist Ferdinand de Saussure. Language is the most important sign system in the human life and in the human society. Moreover, it was a period, during the first steps of organization of the International Association of Semiotic Studies (IASS), of its main journal 'Semiotica' and of the First Congress of IASS (late sixties, early seventies of the past century), when linguistics was the main source of ideas and of methods for semiotics; the latter was a kind of extension of the former. Then, this "pilot role" of linguistics was no longer recognized, but in the last two decades formal grammars have a very important role in DNA computing, in membrane computing and in computer science in general. Taking also into account the importance of concepts such as text, context, intertext and hypertext, we could say that, if not linguistics, then formal linguistics keeps its universality (see also the re-consideration of formal grammars in computational linguistics). Ultimately, let us recall the biological reality of the functional diversification of the brain hemispheres,

the left one being mainly oriented towards sequential structures, i.e., towards language and logic.

The semiotic claim: life is a semiotic phenomenon

This claim was expressed by authors such as Jesper Hoffmeyer(1997, 1998) and Marcello Barbieri (2007). To the question: Is the cell a semiotic phenomenon? Barbieri gives an affirmative answer, claiming that "signs, meanings and codes exist not only in the mental world, but also at the molecular level", so "the cell is a genuine semiotic system". For Hoffmeyer, life, at its most basic, depends on the survival of messages written in the code of DNA molecules and on the tiny cell - the fertilized egg - that must interpret the message and from it construct an organism. For Hoffmeyer, the problem is to explain how nature could come to mean something to someone. The problem of meaning is crucial. However, going now back to Barbieri, the following statement is interesting: "...the genetic code would be real only if it was associated with the production of meaning, but modern science does not deal with meaning"(p.x in "Introduction to Biosemiotics") Then, similarly: "That is the challenge of biosemiotics: the codes are a fundamental reality and we simply have to learn how to introduce signs and meanings in science" (idem, p. xi). It seems that what Hoffmeyer and Barbieri have in view when they refer to 'meaning' is 'information'. Shannon's theory is dealing with what is called sometimes 'selective information'; it fails to capture the semantic information. This is the price Shannon has to pay in order to obtain the possibility to introduce a unit of information, the bit, and to measure, by means of it, the quantity of information. Another important fact is Hoffmeyer's idea that life is a surface phenomenon; he has in view the membrane and quotes Von Foerster, who has proposed the Moebius strip as a topological representation of the kind of logic pertaining to self-referential cybernetic systems. In this framework one can speak of an outside interior and of an inside exterior. These categories are realized through semiotic loops. Autopiesis (U.Maturana,F. Varela)and semiosis are supplementary categories. Living systems may be seen as consisting essentially of surfaces inside other surfaces.

Towards computational biosemiotics

This is a slogan deserving attention of both biosemioticians and people doing research in biocomputing. My published work in this respect concerns: linguistic structures and generative devices in molecular genetics; an attempt to bridge P systems and genomics; the logical and semiotic status of Jacok von Uexkull's concept of Umwelt; an attempt to bridge Uexkull's Umwelt (conceived as an eco-system) and Conway's game of life; an emergent triangle: semiotics, genomics, computation; the semiotics of the infinitely small: molecular computing and quantum computing; symmetry phenomena in infinite words, with biological, philosophical and aesthetic relevance; quasi-periodic infinite words (whose finite version was inspired by DNA). The work done by Tom Head, Gheorghe Paun, Grzegorz Rozenberg, Arto Salomaa et al. in the field of Watson-Crick finite automata, DNA computing, membrane computing and other fields related to non-classical computation has a semiotic potential deserving to be pointed out.

Emmeche's computational notion of life

Claus Emmeche (The computational notion of life Theoria 9(21), 1994, 1- 30)proposed a computational notion of life, just a moment before Lenard Adleman realized, in 1994 his crucial experiment concerning DNA computation. For this reason, it is important to examine Emmche's ideas in order to better understand the great novelty of Adleman's

result. Emmeche examines the relation between metaphorical notions of living organisms as information processing systems and "the idea that life itself is a computatonal phenomenon". Emmeche believes that "the cell has probably quite specific kinds of 'informational' processes for which we might have no equivalent notions within neither computer science nor the field of bioinformatics" and concludes that "the general question of the biosemiotics of the cell should not be confused with a metaphorical use of informational terms". Let us observe that in both DNA computing (Adleman 1994; Paun, Rozenberg, Salomaa 1998) and computing with membranes (Paun 2000) the use of informational and computational terms is no longer metaphorical; the problem of biological computation is effective. Emmeche accepts to speak about biological information at the intracellular level, but he claims that this information is of a different nature than information in the computational sense. To this claim one could reply that as soon as biological computation enters the scene in both theoretical and experimental sense, one can no longer oppose the computational meaning of information to its biological meaning. Here is an interesting quotation from Emmeche(1994:9):

> Sometimes one sees an explicit and intended use of intentional or cognitive terminology within cell biology, suggesting for instance that the cytoplasm of the cell is an intelligent machine [...] because the cell is seen as having many of the data-processing capacities of the computer.

In this respect, the novelty brought by biological computation is that the cell not only has the capacities of a computer, but it is a (potential) computer.

Let us observe that all factors explaining the success of Adleman's experiment have an important semiotic weight. The first factor, Watson-Crick complementarity, is genuine to heredity, already recognized by some leading authors in the field of the philosophy of biology as a sign process; it was transferred in computer science under the status of Watson-Crick automata (Salomaa 1998). Another factor, the use of a right codification, is obviously of a semiotic nature. A third factor, the use of massive parallelism, concerns the strong transgression of the sequential nature of DNA and RNA structures, giving to molecular computation a power that could not be conceived in terms of classical computation. All these factors lead to the surprising superiority of molecular computation in respect to memory and speed. There is one more aspect explaining the success of Adleman's experiment: the existence of a huge number of DNA molecules in a very small space. It points out the contrast between the spatial and the semiotic size, in favor of the latter. This factor too is strongly involved in the efficiency of molecular computation, which is linear in the number n of vertices, contrasting with classical computation, which is exponential in n.

Taking into account that similar facts occur in the field of quantum computation, we may conclude with the following semiotic message: We are leaving the Leibniz symbolically computational metaphor of mind and we enter a new one, much stronger.

Bridging semiotics, genomics and computation

At the crossroad of molecular biology, computer science, linguistics and mathematics, under the stimulus of the recent Human Genome Project and of the emergence of genomics, important semiotic problemms appear, in a perspecive far away from the framework of classical semiotics. The gene-protein interaction points out a syntactic-semantic interplay, where similarity is a basic tool of investigation. But, as Richard Karp (Mathematical challenges from genomics and molecular biology, Notices of A.M. S. 49, 2002, 5, 544-553) shows, this line of research leads to questions of high computational complexity. Algorithmic and computational biosemiotics seems to be a field that no longer can be ignored.

The Gregor Mendel's conjecture

This conjecture (1865) asserts the existence of some discrete units of information (later called genes) which are responsible for the individual aspects of an organism. It was the sign of departure of a very non-conventional semiotic situation. Let us examine it. Usually, the visible world accounts for the invisible universe, i.e., we are looking for visible signs accounting for the objects or situations which are beyond our perception. The macroscopic world is, in most situations, th source of signs accounting for the quantum world as well as for the cosmic processes. We start with hypotheses concerning the similarity or/and the contiguity between some models built by means of the macroscopic world and some hypothetical phenomena in the world of the infinitely small or in that of the infinitely large.

A non-conventional semiotic scenario

In contrast with this traditional situation, the scenario proposed by the semiotic problem of genetics and of molecular biology, as it was formulated by Mendel and by his followers, is just the opposite. We are no longer looking for macroscopic signs of some non-macroscopic phenomena, but for signs in the infinitely small world, accounting for macroscopic phenomena. Instead to have a presence accounting for an absence, we look, in some respect, for an absence explaining a presence.

From invisible entities to visible aspects of inheritance

How inheritance is the result of a representation process having its source in the life of the cell? This question is a challenge for about 150 years. The functioning of the cell is described in terms of interactions among three classes of macromolecules: DNA, RNA, and proteins. Predicting the 3-dimensional structure of a protein from the knowledge of its linear representation as a sequence of amino acids (which, in its turn, is the result of some RNA, transcription of some DNA) is an important open problem; it is investigated by genomics, whose object is the study of genome, defined as the total of DNA molecules in a living organism. The cell has a systemic organization, with genes and proteins as interacting subsystems. It is for long time known that genes are encoded within DNA molecules, the latter being packaged in chromosomes, included in the cell. In 1953, we learned that DNA has a double-helix structure consisting of two strands connected by a very rigorous rule and carrying the same genetic information. Two codification processes occur: the first one is a simple transcription from DNA to RNA, while the second codification is realized by means of a molecular machine called ribosome and moving from RNA to proteins.

The emergence of linguistic metaphors

The analogy between genetics and linguistics involves the transfer in genetics of many linguistic terms. Nucleotide bases are phonemes, codons are morphemes, they are, like in linguistics, grammatical or lexical; The chemical stratum (DNA, RNA, nucleotide bases, codons) defines the syntax, while the biological stratum (amino acids, proteins) defines the semantics of the genetic language. The dictionary leading from codons to amino acids involves synonymy and homonymy phenomena, just like in natural languages.

Does heredity have really a language structure?

The basic question is: to what extent are these metaphors a symptom of some deeper phenomena, motivating to consider heredity having a language structure not only metaphorically, but in a deeper sense? A tremendous quantity of papers concerned with this problem was published so far. One of the initiators of this trend of research was Roman Jakobson and we took from him this problem, in our article "Linguistic structures and generative devices in molecular genetics" (Cahiers de Linguistique Theorique et Appliquee 11, 1974, 1, 77-104) continued by "Language at the crossroad of computation and biology" (in G. Paun, ed. "Computing with Bio-molecules. Theory and Experiment" Singapore et al.: Springer, 1998, 1-35).

From utterances to cistrons

So, we learn that the linguistic level of utterances has as its genetic correspondent the level of cistrons. According to Z. S. Harris (Structural Linguistics, Chicago Univ. Press, 1961), an utterance is any stretch of talk, by one person, before and after which there is silence on the part of the person. Taking into account that the genetic correspondent of the silence could be the starting codon AUG and the stop codons UAA, UAG, and UGA, we define the cistron as a segment of RNA which begins with the starting codon and ends with one of the stop codons. So, the cistron is a string of codons. Utterances are subjected to the whole syntactic ambiguity of a natural language, while the genetic meaning of a cistron is uniquely determined, because, according to some classical results, there is a one-to-one correspondence between cistrons and polypeptide chains (which replaces the old corresondence between genes and proteins). Other units, such as operons, were also discussed in the literature.

Computational biosemiotics enters the scene

The first aim of genomics is to sequence and compare the genomes of different species. To sequence a genome means to make explicit the bases composing it. According to the analogy between bases and phonemes, the considered operation is similar to what was done in American descriptive linguistics under the name of phonemic segmentation (Harris 1961). From the genome of the individuals one more step leads to the genome of human species. Each individual has its specific genome and this is true for both human and non-human beings. However, insight a definite species the situation is very misleading. For instance, on the one hand, any two humans agree in about 999 bases out of 1,000 (Karp 2002:545), but, on the other hand, the genomes of any two humans differ considerably.

Syntax and semantics

The sequencing operation, a purely syntactic one, is already almost accomplished. The next part of the Human Genome Project (HGP) is directed towards the semantics of the genome. Let us recall that syntax involves only concatenative aspects, such as sequencing DNA or proteins as strings over some finite alphabet. Typical for the semantics of the HGP is the task to determine the functions of the proteins, as they are encoded by various genes, and to find out, for each gene, what protein is produced and activated. This task is very bold, because the human genome contains about three billions base pairs and about 35,000 genes. It seems, however, that, in respect to the aim fixed in 1990, HGP can be appreciated as a great success.

Bridging P systems and genomics

Bridging genomics and P systems could give to the former the possibility to take profit of the computational capacities of the latter. Moreover, suggestions coming from genomics could enrich the study of P systems with new biological and computational ideas. The following lines aim to be a preliminary step in this respect.

Life is DNA software + membrane software

"Life is a surface activity" [...] "Life is fundamentally about insides and outsides" (Jesper Hoffmeyer, "Surfaces inside surfaces", Cybernetics and Human Knowing 5(1), 1998, 33-42; "The biology of signification". Perspectives in Biology and Medicine 43(2), 2000, 252- 268). Relevant parts of the environment are internalised as an inside exterior/inner outside (the so-called Uexkull's Umwelt (J. Uexkull, "The theory of meaning". Semiotica 42(1), 1982, [1940], 25-82). The representation of certain environmental features inside an organism by various means (Uexkull, 1982), while the interior becomes externalised as a outside interior/ outer inside, in the form of the "semiotic niche" (Hoffmeyer 1998), as informed and changed by the inside needs of the organism pertaining to that niche (C. Emmeche, K. Kuhl, F. Stjernfelt, "Reading Hoffmeyer, rethinking biology" Tartu Semiotic Library 3, Tartu University Press, 2002). This inside/outside interplay is made possible by the membrane strictly governing the traffic between them. P systems (Gheorghe Paun, Membrane Computing: An Introduction. Berlin et al.: Springer, 2002) find their starting point in this biological reality, to which a computational dimension is added. In agreement with the ideas of DNA computing and membrane computing, S. Wolfram (A New Kind of Science. Wolfram Media, Inc., October 2001) proposed to see life as a universal Turing machine, to which G. Chaitin (Bulletin of the EATCS 2002) adds the condition of a high program-size complexity. The project of bridging genomics and P systems could have the slogan: Life is DNA software + membrane software.

P systems and the Human Genome Project

The HGP is a good starting point for the problem raised in the above title. of ths A P system with replicatd rewriting is a construct P = ¡ V, T, m, M(1), ..., M(m), R(1), ..., R(m)¿, where V is an alphabet, its elements are called objects; T is contained in V and it is called the output alphabet; m is a membrane structure consisting of m membranes (or regions of a membrane) labeled 1, 2, 3, ..., m, such that each membrane, except the first is completely contained within another; M(1), ..., M(m) are finite languages over V; R(1), ..., R(m) are finite sets of developmental rules. The languages M(i) and the rules R(i) are associated with the regions of m, for any i between 1 and m. This variant of P systems, whose general theory belongs to Gheorghe Paun, , was proposed by J. Aguado, T. Balanescu, T. Cowling, M. Gheorghe, M. Holcombe, F. Ipate in Fundamenta Informaticae 49(1-3), 2002, 17-33. Its advantage for dealing with genomics is the distinction between an input and an output alphabet. The usual , starting interpretation of the objects forming the alphabet of a P system is to think at them as molecules. The general theory of P systems does not depend on the way we interpret these objects; however, the intuitive representation of them decides to a large extent the type of problems which are investigated. Now the question is: which are the P systems accounting for the tasks of genomics: a)the syntactic task: to sequence and compair the genomes of different species and b)the semantic task: to identify the genes and determine the functions of the proteins they encode.

Referring to P systems of the type considered above, a first idea is to work with an alphabet V including both the types of nucleotide bases and the types of amino acids,

while the output alphabet T contained in V will be the set of various types of amino acids. The P system we are looking for should describe the process leading from DNA to its segmentation in nucleotide bases, from this segmentation to the identification of genes, which are privileged substrings of DNA, carrying the genetic information, and finally from genes to protein functions (the latter being hypothetically related to the protein sequencing, i.e., to their decomposition in amino acids). So, the membrane structure should consist of several regions , such as: a region of nucleotide bases, a region of genes, a region of amino acids, a region of DNAs, a region of proteins, all of them being contained in the initial region represented by the cell. We are already faced with a necessary extension of the relation 'contained in', used in the definition of a P system.

Besides its usual meaning, when we refer, for instance, to the fact that DNA is included in the cell, we consider also the substring-string relation, as a variant of 'contained in', accepting so that the region of the nucleotide bases is contained in the region of DNAs (meaning that any element of the former region is a substring of an element of the latter); similarly, the region of genes is contained, in this view, in the region of DNAs; the region of amino acids is contained in the region of proteins, while the region of codons is contained in the region of RNAs and all are contained in the cell. In a similar way we have to cope with cistrons, reads, clones and other objects involved in the cell-processes.

Another aspect deserving a special discussion is the interior- exterior distinction, involved in the structure of a P system. In the light of the ideas exposed above, it should be replaced by a four-steps organization: interior, exterior interior, interior exterior, and exterior, according to Hoffmeyer's approach. Further examination deserves the developmental rules, the phylogenetic trees, and the exons, introns and codons.

Bibliographic information

For more details and supplementary aspects bridging biosemotics and biocomputing, here are some of my articles on such topics:

[1] Membranes vs DNA. *Fundamenta Informaticae* 49 (1/3) 2002, 223-227.

[2] An emergent triangle: semiotics-genomics-computation. Proc. of the *International Congress of German Semiotics Society*, Kassel, 2002. CD-ROM 2003.

[3] Bridging P systems and genomics. In "Membrane Computing", eds. G. Paun, G. Rozenberg A. Salomaa, C. Zandron, *LNCS* 2597, Berlin et al.:Springer, 2003, 371-376.

[4] The duality of patterning in molecular genetics. In N. Jonoska et al. (eds) "Aspects of Molecular Computing", *LNCS* 2950, Berlin et al.:Springer, 2004, 318-321.

[5] The semiotics of the infinitely small: molecular computing and quantum computing. In K. Tsoukala et al. (eds.) "Semiotic Systems and Communication-Action-Interaction-Situation and Change". Proc. of the *6th National Congress of the Hellenic Semiotics Society*. Thessaloniki, 2004, 15-32.

[6] Z. Pawlak, a pioneer of DNA computing and of picture grammars. *Fundamenta Informaticae* 75, 2007, 1/4, 331-334.

Turku
Centre *for*
Computer
Science

Joukahaisenkatu 3-5 B, 20520 Turku, Finland | www.tucs.fi

University of Turku
- Department of Information Technology
- Department of Mathematics

Åbo Akademi University
- Department of Computer Science

Turku School of Economics and Business Administration
- Institute of Information Systems Sciences

ISBN XXX-XXX-XX-XXXX-X
ISSN XXXX-XXXX