Client–Server P Systems in Modeling Molecular Interaction

Gabriel Ciobanu[1], Daniel Dumitriu[2], Dorin Huzum[2],
Gabriel Moruz[2], Bogdan Tanasă[3]

[1] National University of Singapore, School of Computing,
gabriel@comp.nus.edu.sg
[2] "A.I.Cuza" University of Iaşi, Romania
{danield,huzzy,gabbym}@infoiasi.ro
[3] Institute of Computer Science, Romanian Academy, Iaşi,
bogdan@iit.tuiasi.ro

Abstract. We present a new version of P systems called *Client–Server P Systems* (CSPS). The client membranes are characterized by their states; the server membrane stores the states of the clients and triggers the corresponding interaction rules. We show that CSPS have the same expressive power as Turing machines. CSPS is used to model various molecular processes in which interaction and state transitions are causally linked. Signaling pathways and T cell activation are described by using a CSPS software environment called MOlNET (MOlecular NETworks). MOlNET can describe the dynamics of molecular interactions, including both qualitative and quantitative aspects and simulating the signaling pathways that tune the activation thresholds for T cells.

1 Introduction

Membrane computing is based on *membrane systems* or P systems, a new class of distributed and parallel computing devices introduced in [12]. The approach is based on hierarchical systems: finite cell-structures consisting of cell-membranes embedded in a main membrane called the *skin*. The membranes determine *regions* where *objects*, elements of a finite set, and evolution rules can be placed. The objects evolve according to given *rules* associated with a region. Objects may also move between regions. A *computation* starts from an initial configuration of the system, and terminates when no further rule can be applied. A software simulator of membrane systems is presented in [3].

Hierarchical systems are well-known structures in computer science, and the notion of computation based on evolution rules is common. The interpretation of the computation is rather new: the result of a computation is a multiset of objects collected in the output cell or sent out of the system. The behaviour of the whole system is obtained by combining the resulting multisets, or considering the multiplicity of objects present in a specific output membrane of a final configuration.

According to many authors, P systems are not created to model biological systems, although similarities can be observed. Since their introduction, many results of universality were proved and several theory problems could be explained in an easier and elegant manner, with the help of formal languages. Yet, the subject is far from being exhausted; new properties are discovered, as well as connections with already known concepts. There is still a need to find more connections with the applied computer science and approaches that could prove themselves useful in practice as well as in theory. We present a new version of P systems, one related to the applied computer science and molecular biology. Trying to strengthen the connection between P systems and biological systems, we introduce P systems of interaction, considering the client–server model used for process communication in computer networks.

We use a new version of P systems called Client-Server P Systems (CSPS) to model molecular processes as signaling pathways and T cell activation. A living cell is a complex system where the interactions among its components provide structure, mediate reactions, and perform functions. Different cell processes (such as T cell activation) are triggered mainly by protein-protein interactions. Specific and productive interactions change consequently the states of the interacting proteins (such as in signal transduction). Such molecular processes are modeled in MOlNET, a CSPS software environment more meaningful for biologists and for their goals.

The paper is organized as follows. Section 2 presents the client-server P systems, showing that they have the same computational power as Turing machines. Section 3 briefly describes the important notion of a signaling pathway and T cell activation. Section 4 presents MOlNET as a CSPS software environment. MOlNET is based on the client–server model of the computer networks. After a brief presentation of MOlNET, emphasizing on its components and their behaviour similar to CSPS, we simulate the signaling pathways that finely tune the T cell activation thresholds taking into consideration both qualitative and quantitative aspects.

2 Client–Server P Systems

A formal description of a *P system* can be found in [12]. It is basically composed of a *membrane structure*, consisting of several membranes that do not intersect, and a *skin membrane*, surrounding them all; outside the skin membrane lies the *environment*. The membranes delimit regions, numbered accordingly, and contain originally multisets of *objects*, as well as *evolution rules* involving objects (and possibly priorities for rules).

This is the initial state of the P system. In each step, rules are applied non-deterministically and in a maximal and parallel manner, in all membranes – in other words, rules and objects are randomly chosen, all objects that can be involved really are, and all chosen rules are applied in parallel.

The objects can pass membranes, even to or from the outside of the system; in this way we obtain *transitions* between *configurations* of the system. A sequence of transitions constitutes a *computation*; this *halts* if at one moment, no rule can be applied anymore in the system. When getting a halting computation, we collect its *result* by counting the objects that ended in the *output membrane.*

These are the general P systems, and many other variants and classes were introduced starting from this simple description. For our paper, a particular kind of P systems is of special interest, *P systems with symport/antiport rules*; they take benefit from the communication that is intensively used during computations. The specificity of this type of P systems lies in the form of their rules, which can be one of:

– $(a, in), (a, out)$: object a can pass through a membrane by itself (*uniport rules*),

– $(ab, in), (ab, out)$: objects a and b can pass through a membrane only together, in the same direction (*symport rules*), and

– $(a, out; b, in)$: objects a and b can pass through a membrane only together, but in different directions (*antiport rules*).

There is a number of results on this type of P systems [11]; generally, they take into consideration the number of membranes and the *weight* of the port (i.e. the number of objects involved in an antiport or symport rule). A relation of inverse proportionality between these two parameters could be observed – that is, in order to obtain computational universality, one has to increase the weight of the ports when decreasing the number of membranes, and reversely. In the proofs from this paper, we use some results and techniques rooting in the formal language theory that can be found in [14], for example.

A *matrix grammar with appearance checking* is defined as a construct $G = (N, T, S, M, F)$, where N, T are disjoint alphabets, $S \in N$, M is a finite set of sequences of the form $(A_1 \rightarrow x_1, \ldots, A_n \rightarrow x_n)$, $n \geq 1$, (called *matrices*) of context-free rules over $N \cup T$ (with $A_i \in N, x_i \in (N \cup T)^*$), and F is a set of occurrences of rules in M.

For $w, z \in (N \cup T)^*$ we write $w \Longrightarrow z$ if there exists a matrix $(A_1 \rightarrow x_1, \ldots, A_n \rightarrow x_n)$ in M and the strings $w_i \in (N \cup T)^*, 1 \leq i \leq n+1$, such that $w = w_1, z = w_{n+1}$, and, for all $1 \leq i \leq n$, either $w_i = w_i' A_i w_i'', w_{i+1} = w_i' x_i w_i''$, for some $w_i', w_i'' \in (N \cup T)^*$, or $w_i = w_{i+1}$, A_i doesn't appear in w_i, and the rule $A_i \rightarrow x_i$ appears in F. (The rules of a matrix are applied in order, possibly skipping the rules in F if they cannot be applied, hence the grammar's name.)

The language generated by G is defined by $L(G) = \{w \in T^* \mid S \Longrightarrow^* w\}$. The family of languages of this form is denoted by MAT_{ac}. It is known that matrix grammars with appearance checking generate the family RE of recursively enumerable languages. A matrix grammar G with the notations made above is in the *binary normal form* if $N = N_1 \cup N_2 \cup \{S, \#\}$, all sets disjoint, and the matrices in M are in one of the following forms:

1. $(S \rightarrow XA)$, with $X \in N_1, A \in N_2$,
2. $(X \rightarrow Y, A \rightarrow x)$, with $X, Y \in N_1, A \in N_2, x \in (N_2 \cup T)^*, |x| \leq 2$,
3. $(X \rightarrow Y, A \rightarrow \#)$, with $X, Y \in N_1, A \in N_2$,
4. $(X \rightarrow \lambda, A \rightarrow x)$, with $X \in N_1, A \in N_2, x \in T^*, |x| \leq 2$.

Moreover, there is only one matrix of type 1 and F consists exactly of all rules $A \to \#$ appearing in matrices of type 2; $\#$ is a trap-symbol (it is never removed once introduced). A matrix of type 4 is used only once, in the last step of a derivation. According to Lemma 1.3.7 of [6], for each matrix grammar, there exists an equivalent matrix in the binary normal form.

2.1 Description

We formalize a client–server model complying with the following description. The clients are characterized by their states, whereas the server stores the current states of clients and also interaction rules defined over these states. When two clients can interact, the server notifies them, supplying at the same time the corresponding rule. The clients interact and send their new states to the server, thus making the model consistent.

A *Client–Server P System* (CSPS) is a P system composed of elementary membranes (except for the skin), state-objects for all of them minus one (the server) and objects modelling the real communication channels between the clients. All rules are of symport type. A CSPS resembles the original client–server model by working with information and not using rules with creation/destruction of objects, but merely the power of communication. In a formal notation, the CSPS contains the skin membrane (numbered 1), m membranes each representing a client (numbered from 2 to $m+1$) and one for the server, numbered $(m+2)$, all arranged inside the skin membrane. We have state-objects for possible states of clients and rule-objects.

A rule-object $\eta_{\alpha_1\alpha_2\alpha_3\alpha_4\alpha_5}$ has the following meaning: two clients defined by states α_1 and α_3 can interact and pass to states α_2 and α_4, respectively; at the same time, it is possible to obtain a supplementary information, α_5.
Let Π be our CSPS:

$$\Pi = (V, \mu, w_1, \dots, w_{m+2}, w_e, M_e, R_1, \dots, R_{m+2}, m+2)$$

where:

1. $V = A \cup B$, with A, B disjoint sets constructed as follows:
   – $A = \bigcup_{i=2}^{m+1} S_i$, where S_i is the set of states for client i;
   – $B = \{\eta_{\alpha_1\alpha_2\alpha_3\alpha_4\alpha_5} \mid \alpha_5 \in M_e \cup \{\lambda\}, \alpha_i \in A \cup M_e,\ 1 \leq i \leq 4$, where $\alpha_1 + \alpha_2 \Rightarrow \alpha_3 + \alpha_4 + \alpha_5$ is an interaction rule$\}$,
2. $\mu = [_1\, [_2\,]_2 \dots [_{m+2}\,]_{m+2}\,]_1$,
3. $w_1 = \emptyset$,
   $w_{m+2} = B \cup S_{initial}$, where $S_{initial} = \{s_2, s_3, \dots, s_{m+1}\}$, $s_i \in S_i$, $2 \leq i \leq m+1$ (the initial state of each client),
   $w_i = S_i \setminus \{s_i\}$, $s_i \in S_{initial}$, $2 \leq i \leq m+1$,
   $M_e = A$ (multiset),
4. $R_1 = \{(\alpha_j\ \alpha_k\ \eta_{\alpha_1\alpha_2\alpha_3\alpha_4\alpha_5}, out) \mid j \in \{3,4\}, k \in \{1,2\}, j-k \neq 2, \alpha_j, \alpha_{j+2} \in A, \alpha_k, \alpha_{k+2}, \alpha_5 \in M_e i\}, \cup \{(\alpha_3\ \alpha_4\ \eta_{\alpha_1\alpha_2\alpha_3\alpha_4\alpha_5}, out) \mid \alpha_i \in A,\ 1 \leq i \leq 4,\ \alpha_5 \in M_e\} \cup \{(\alpha_3\alpha_4\alpha_5\eta_{\alpha_1\alpha_2\alpha_3\alpha_4\alpha_5}, in) \mid \alpha_i \in A \cup M_e,\ 1 \leq i \leq 5\}$
   $R_{m+2} = \{(\alpha_1\ \alpha_2\ \eta_{\alpha_1\alpha_2\alpha_3\alpha_4\alpha_5}, out),\ (\alpha_3\ \alpha_4\ \alpha_5\ \eta_{\alpha_1\alpha_2\alpha_3\alpha_4\alpha_5}, in) \mid \alpha_i \in A \cup M_e,\ 1 \leq i \leq 4,\ \alpha_5 \in M_e \cup \{\lambda\}\}$,

$R_i = \{(\alpha_j\ \eta_{\alpha_1\alpha_2\alpha_3\alpha_4\alpha_5}, in),\ (\alpha_{j+2}\ \eta_{\alpha_1\alpha_2\alpha_3\alpha_4\alpha_5}, out) \mid j \in \{1,2\},\ \alpha_j, \alpha_{j+2} \in S_i\},\ 2 \leq i \leq m+1$.

Inside the server membrane there are state-objects (representing the current states of the clients) and rule-objects. When two state-objects can be combined according to a rule given by a rule-object, the server membrane gives a "signal" to the respective client-membranes.

The meaning of the rule $(\alpha_1\ \alpha_2\ \eta_{\alpha_1\alpha_2\alpha_3\alpha_4\alpha_5}, out) \in R_{m+2}$ is the following: the clients represented by membranes i and p (where $\alpha_1 \in S_i$ and $\alpha_2 \in S_p$) can interact according to the rule described by the rule-object η; as a result, these three objects (the current states and the rule-object) exit the server region. At this moment the involved client-membranes absorb their own state-objects and the rule-object η (by means of rules $(\alpha_1\ \eta_{\alpha_1\alpha_2\alpha_3\alpha_4\alpha_5}, in) \in R_i$ or $(\alpha_2\ \eta_{\alpha_1\alpha_2\alpha_3\alpha_4\alpha_5}, in) \in R_i$, and similarly for membrane p). Then they release for further use their new states and the rule-object into the skin membrane, by $(\alpha_3\ \eta_{\alpha_1\alpha_2\alpha_3\alpha_4\alpha_5}, out) \in R_i$ or $(\alpha_4\ \eta_{\alpha_1\alpha_2\alpha_3\alpha_4\alpha_5}, out) \in R_i$ (and similarly for p). If $\alpha_5 \neq \lambda$, the supplementary information is brought in from the environment with rules of R_1.
We emphasize the fact that the notifications of clients and the interactions between them take place in a parallel manner.

2.2 The Computational Power

We show that the Client–Server P Systems are computationally universal, i.e. they have the same computational power as Turing machines. Because of the characteristics of the model, it is more convenient to use in the proof a slightly modified grammar. Given a general matrix grammar $G = (N, T, S, M, F)$ in the binary normal form, let us denote by $G' = (N', T, S, M', F')$ the grammar constructed as follows:

- $N' = N_1 \cup N_2 \cup N_3 \cup \{S, \#\}$, all sets disjoint, with $N_3 = \{u' \mid u \in N_2\}$; we define the bijection $corresponding : N_2 \to N_3, corresponding(u) = u'$;
- the rule $(S \to XA) \in M$ is replaced in M' by $(S \to XA')$, with $X \in N_1, A' \in N_3, A' = corresponding(A)$.
- we keep the matrices of G of types 2, 3, 4 and:
  - for each matrix of type 2, $(X \to Y, A \to x) \in G$, we add $(X \to Y, A' \to x_1)$ with $corresponding(A) = A' \in N_3, x_1 \in (N_3 \cup N_2 \cup T)^*$; if $1 : x \in N_2$ then $1 : x_1 = corresponding(1 : x)$ and $2 : x_1 = 2 : x$, else $x_1 = x$;
  - for each matrix of type 3 and 4, we add the corresponding rule: $(X \to Y, A' \to \#)$ or $(X \to \lambda, A' \to x)$, with $X, Y \in N_1, corresponding(A) = A' \in N_3, x \in T^*, |x| \leq 2$.
- F' consists of rules in F plus added rules of type $A' \to \#$, where $A' \in N_3$, $A' = corresponding(A)$.

In this definition we use the notation $\alpha : x$ representing the α^{th} element of the sequence x (for instance, $2 : x$ means the second element of x).

We prove that the two grammars are equivalent, so we can further use any of them, depending on which of them fits better the goal.

Lemma 1. *$G \equiv G'$.*

Proof. We show that every derivation of G can be simulated in G' and vice versa. This is proved by induction on the length of derivation. □

Notations:
Given a system Π as defined in the previous section, the set of all numbers computed by Π is denoted by $N(\Pi)$. We denote by $NCSP_{m,p}$ the family of all sets $N(\Pi)$ computed by CSPS of degree at most $m \geq 1$, using symport rules of weight at most $p \geq 1$. The weight of a symport rule $(a_1 \dots a_n, in/out)$ represents the number n of the objects appearing in the rule. For instance the weight of $(a\,b\,c, in)$ is 3. We denote by $N'RE$ the family of all recursively enumerable sets of non-null numbers. We prove that for $m = p = 4$, CSPS are computationally universal (except that they cannot yield the number 0).

Theorem 1. *$NCSP_{4,4} = N'RE$.*

Proof. The direct inclusion can be proved in a straightforward manner.

For the reverse one, we use a method often employed when proving results about universality of different types of P systems, namely exploiting the properties of the matrix grammars with appearance checking. Yet, because of model's characteristics, it is more convenient to use in the proof a matrix grammar modified as above, that is having the set of non-terminals differently partitioned and the rules modified accordingly.

It is known that matrix grammars with appearance checking generate the family RE of recursively enumerable languages. Taking this into consideration, as well as the results regarding the equivalence of the general matrix grammars and the ones in the binary normal form, and the latter with our modified type (see Lemma 1), we construct a client–server P system of degree 4 and symport weight 4 which simulates the derivations of the "modified" grammar. Let this be $G' = (N', T, S, M', F')$, with $N = N_1 \cup N_2 \cup N_3$. For every matrix $(X \to Y, A \to \alpha_1\alpha_2)$ in G', with $X, Y \in N_1, A \in N_2 \cup N_3$ and $\alpha_1, \alpha_2 \in N_2 \cup N_3 \cup \{a, \lambda\}$, we have an associated object of type η: $\eta_{XAY\alpha_1\alpha_2}$. For the rules $(X \to \lambda, A \to \alpha_1\alpha_2)$ and $(X \to Y, A \to \#)$ we introduce the objects $\eta_{XAt\alpha_1\alpha_2}$ and η_{XAYb}, respectively. In order to be able to eliminate these objects η from the output membrane (region) at the end of the derivation, we introduce some special objects s and s' for every object η.

A sketch for the simulation of a general derivation for the case $A \in N_3$ can be seen in Figure 1. At the beginning, in membrane 2 there are the elements of N_1, in membrane 3 there are the elements of N_3 and the terminal a, while in 4 we have the right hand side of the starting rule of grammar and the rule-objects. In the environment there are the elements of N_2 and the terminal a, in an infinite number of copies. In the case when $A \in N_2$, the system functions in a similar manner, except for the fact that steps 4 and 5 are skipped, and in step 6 the rule applied is $(YA\,\eta_{XAY\alpha_1\alpha_2}, out) \in R_1$.

We use a matrix $(X \to \lambda, A \to x)$ of type 4 in G only when an object t appears in membrane 4, triggering the mechanism for removing the objects

1: $(XA\eta_{XAY\alpha_1\alpha_2}, out)$
2: $(X\eta_{XAY\alpha_1\alpha_2}, in)$
3: $(Y\eta_{XAY\alpha_1\alpha_2}, out)$
4: $(A\eta_{XAY\alpha_1\alpha_2}, in)$
5: $(\alpha_1\eta_{XAY\alpha_1\alpha_2}, out)$
6: $(Y\alpha_1\eta_{XAY\alpha_1\alpha_2}, out)$
7: $(Y\alpha_1\alpha_2\eta_{XAY\alpha_1\alpha_2}, in)$
8: $(Y\alpha_1\alpha_2\eta_{XAY\alpha_1\alpha_2}, in)$

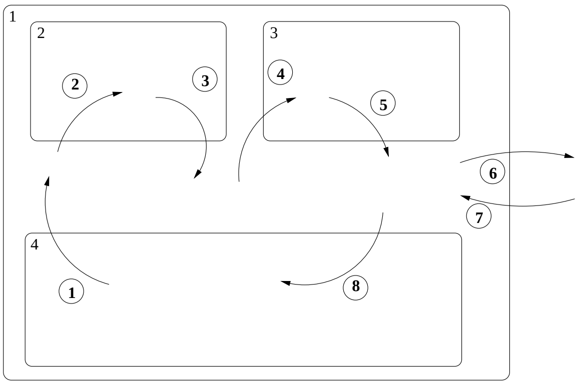


Fig. 1. A client-server P system of degree 4 and symport weight 4

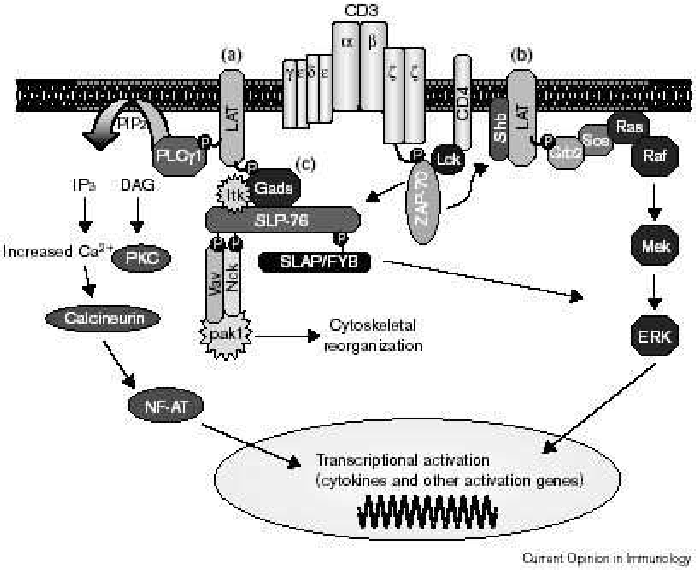
η from the output membrane. When a matrix $(X \to Y, A \to \#)$ is used in G', the trap-object b is introduced in membrane 4 preventing the end of the computation. □

Having introduced the CSPS as above, one question arises: are there problems that could be more naturally described (and *solved*) by CSPS than by other P systems? A positive answer comes from biology. We have detected that CSPS approach has a very interesting biological feedback when it is used to describe the T cell activation.

3 Signaling Pathways and T Cell Activation

Signaling pathways

Many biological activities depend on the ability of proteins to communicate specifically with each other and with other molecules. This is particularly obvious for the signaling pathways that operate in living cells. These signaling pathways allow the cell to receive, process and respond to various *signals* from its environment. The signals are molecules (hormones, neurotransmitters, cytokines) that indicate the beginning and the termination of one or more intracellular processes. The *signal transduction* refers to the process by which the signals are transmitted via receptors to the interior of the cell through the signaling pathways. Classically, a linear signal transduction pathway can be described as follows. A first messenger molecule (the signal) outside the cell is sensed at the cell membrane by a receptor. Receptor binding is transduced into an intracellular event by activation of enzymes (PLC, for instance) that synthesizes second messenger molecules (e.g. DAG, IP3). These ones promote the covalent modifications (by protein kinase or protein phosphatase activities) and allosteric regulation of other intracellular proteins that ultimately cause specific changes in gene expression. In living cells, these linear signaling pathways cross-talk each other and form a network of interactions. This signaling network has several emergent properties that the individual pathways do not have [1]. We consider the signaling network that grounds T cell activation as a case study of our model. When a T cell recognizes a foreign antigen, it initiates several signaling pathways and the cell activates. The recognition of foreign antigen is an extremely *sensitive*, *specific* and *reliable* process and models for T cell signaling are necessary to understand how these properties arise. So far, the study of T cell activation has benefited from the use of mathematical models [2] or are based on Boolean formalisms [8].



Current Opinion in Immunology

Fig. 2. TCR signaling pathways

T cell activation

T cells play an important role in orchestrating the immune responses to foreign aggressors. The key event for T cell activation is an appropriate interaction between the T cells armed with T cell antigen receptors (TCR) and the professional antigen presenting cells (APC). TCR only recognizes the foreign antigen in the form of short peptides presented in the groove of a molecule on the surface of the APC known as the major histocompatibility complex (MHC). The physical interaction of TCR with MHC-peptide complexes is unique among signaling systems in that it takes place over a continuum of binding values. The recognition of antigen initiates signal transduction. This can be broken down into series of discrete steps that are related to molecular events (interactions, state transitions) within the signaling pathways. These are shown in Figure 2, reprinted from [10].

The integration of distinct branches of the TCR-induced signaling pathways (Ras/MAP kinase pathway, Ca^{+2}/calcineurin pathways) results in activation of distinct transcriptional factors that coordinate and regulate gene expression and cell activation. Other signals (costimulatory signals) also function in T cell activation by amplifying TCR signaling. The signaling pathways are modeled in MOlNET and the network behaviour is traced out both qualitatively and quantitatively.

4 MOlNET: a CSPS Software Environment

In this section we present MOlNET, a software environment for CSPS, first introduced in [4]. The link between CSPS and MOlNET is given by the fact that each component of MOlNET has a clear correspondent in CSPS, with the same role and behaviour. We use MOlNET to simulate the signaling pathways that tune the activation thresholds for T cells, providing insights on both quantitative and qualitative aspects.

4.1 Description

In every biological interaction, one or both interacting molecules undergo a transition to a new state. Consequently, a MOlNET structure contains information about each of the interacting molecules (name, type, location inside cell and number of relevant domains with respect to a given interaction, if the molecule is a protein). For each protein it is possible to mention the names of its domains and domain states (active/inactive).

Our system allows a user to define molecules (proteins, ions, lipids, DNA) that take part in a simulation, and also the interaction rules between them: protein-protein interactions, protein-DNA interactions and other features of molecular systems as a set of reaction rules. The user can observe, through various windows, dynamic changes in concentration of proteins and other chemical compounds of the cells. Using this software, we can tune up a specific behaviour and we can emphasize the problems related to interaction anomalies.

Our implementation choices were to use the C-standard programming language and the BSD-socket interface approach. The graphic server is written in Gtk 2.0 environment. The main components of our software system as well as the relationships established between them are shown in Figure 3.

The MOlNET architecture follows the definition of CSPS. Hereinafter we describe the components of our software environment, as well as the links with CSPS. The data server manages information about the interactions among the clients. It corresponds to the server membrane in CSPS, having the same role and behaviour. This component uses a non-directed graph defined by the molecules existing in the cell and their productive interactions. By productive interaction we understand that their interaction brings either qualitative or quantitative changes to the cell (the modification of a molecule state or concentration, forming or unbinding of a molecular complex).

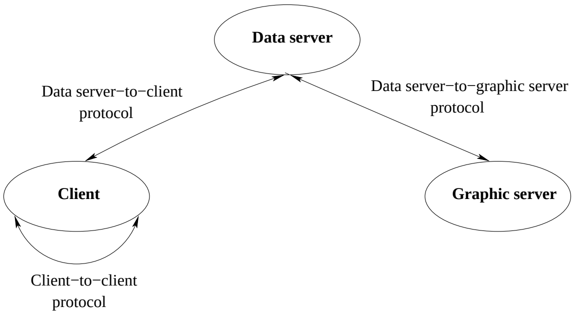
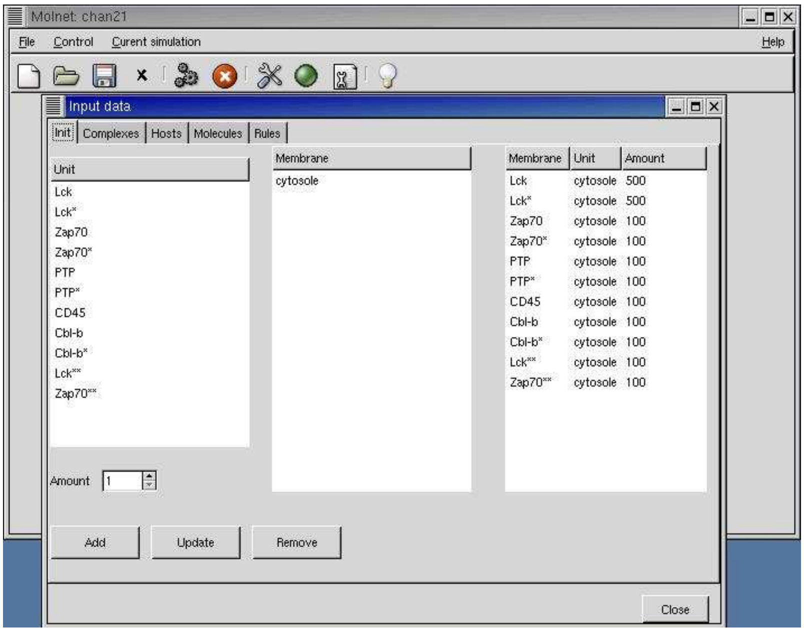

Data server
Data server−to−client protocol
Data server−to−graphic server protocol
Client
Graphic server
Client−to−client protocol

Fig. 3. Simplified MOlNET architecture

The server carries out the decision function with regard to the possible molecule interactions. When it receives a request from a client, it checks the request type (new client or client modification), updates the graph and sends back the appropriate responses (a list of possible partners). It also keeps the amounts of each molecule existing in the cell.

A client represents a type of molecule; it corresponds to a client membrane in CSPS, having the same functionality. Each client saves information about its state: chemical structure, position, identification within computer networks (host+port). According to the model described above, a client (molecule) changes its position inside the cell and checks whether its interaction partners are available and close enough for an interaction. As soon as it participated to an interaction, a client notifies the server. The communication mechanism closely follows the CSPS formalism presented in Section 2. Specifically, the rule-objects of CSPS are interaction rules in MOlNET, initially stored in the data server. As a result of the message exchange, the data server and clients update their states, in the same way as transitions take place in CSPS.

The graphic server is an important part of the system, as it provides a friendly visual interface to the user. The users just introduce the input data, and then observe the behaviour of the system. A snapshot of the MOlNET user interface is given in Figure 4. In order to follow the real interactions between molecules, the graphic server offers the possibility for both defining the molecules that are in the system, and interaction rules between these molecules. If experiments prove that some rules were incorrect or if on the contrary they discover new interactions, it is easy to alter old rules or to add new ones, by the simple act of introducing new molecules and/or rules. We considered the signaling pathways that tune the T cell activation thresholds. The work is presented in the next section. The results obtained by running the system are represented as charts by the graphic server. The feature that allows direct interventions from outside is also carried out by this component.



Unit

Lck
Lck*
Zap70
Zap70*
PTP
PTP*
CD45
Cbl-b
Cbl-b*
Lck**
Zap70**

Membrane

cytosole

Membrane	Unit	Amount
Lck	cytosole	500
Lck*	cytosole	500
Zap70	cytosole	100
Zap70*	cytosole	100
PTP	cytosole	100
PTP*	cytosole	100
CD45	cytosole	100
Cbl-b	cytosole	100
Cbl-b*	cytosole	100
Lck**	cytosole	100
Zap70**	cytosole	100

Fig. 4. A MOlNET screen for tuning the activation thresholds

As each particular client/server model should rely on its own protocol, we developed a set of protocols allowing the communication between data server and clients, data server and graphic server, clients themselves.

4.2 Tuning the T Cell Activation Thresholds

MOlNET is able to catch both qualitative and quantitative aspects. The **qualitative approach** includes *connectivity data* (molecular interaction map) and *semantic data* (the effect of an interaction, mainly activation or inhibition). It allows a *dynamic description* of the signaling network. The **quantitative approach** includes kinetic data (amounts of molecules, reaction rates) and allows *predictions* on the network behaviour. A common difficulty when dealing with the simulation of molecular interactions is the lack of quantitative data.

T cell activation is a threshold phenomenon that is dynamically modulated (or tuned) during cell maturation [7]. It reflects the signal intensity that is necessary to increase the expression of specific genes (e.g. IL-2 gene). Both the emergence of threshold and its tuning depend on dynamic interplay between positive and negative factors. As T cells receive many signals from self antigens, they have to adapt their activation thresholds in such a way that self-stimuli fall under the threshold and consequently no response is elicited against self. Furthermore, non-self antigens provide stronger signals that overcome the activation threshold, the cell activates and an immune response is yielded. In our work, the activation threshold concept is considered on a molecular basis.

Although many receptor interactions may contribute positively or negatively to the setting of activation threshold, TCR signaling plays the dominant role and has its own signature. Zap70 activation and phosphorylation of LAT are hallmarks of TCR engagement and are essential for connecting to the major intracellular signaling pathways (Ca^{+2}/calcineurin and Ras/MAPK kinases) that lead to T cell activation.

In the following, we investigate the role of Cbl-b in tuning the activation thresholds. Basically, we look for the influences that Cbl-b exerts on the level of activated Zap70. First, the model proposed in [2] is considered. Then, Cbl-b is added and the levels of Zap70* are measured. High levels of Zap70* may trigger cell activation, while levels below the threshold have not this effect. The expression levels of various signaling proteins vary during immune cell maturation (e.g the level of Lck declines during development whereas the level of SHP phosphatase increases); our work considers the heterogeneity of activation thresholds at the level of population of T cells (or T cell clones) rather than during the development of a single clone.

According to [2], the state of TCR signaling (hence the activation threshold) appears to be governed by a dynamic balance of kinases (Lck, Zap70) versus phosphatases (SHP1, SHP2). Moreover, there are some feedback interactions among Lck, Zap70 and phosphatases (the later are represented by a single variable called PTP). As shown in Figure 5, Lck* phosphorylates and activates both the ZAP70 (step 3) and the PTP (step 5). Zap70* activates Lck in step 1 (positive feedback), but phosphatase inactivates both Lck* (step 2) and ZAP70* in step 4 (negative feedback). The active state of an enzyme is denoted by "*".

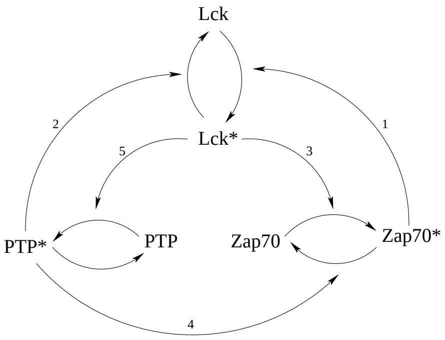


Fig. 5. Feedback interactions as TCR signals

These feedback combinations may have a nonlinear effect that alters the appropriate threshold for cell activation. Moreover, these interactions due to feedbacks provide robust mechanisms for antigen discrimination (self vs non-self). Taking into account the above particularities of the reaction rules, one can observe in Figure 6 the changes of Lck*/Lck-total and Zap70*/Zap70-total ratios after TCR engagement. For various amounts of Lck and Lck*, Zap70*/Zap70-total ratio has slight variations. If the cell activation threshold (namely its requirement for Zap70*) is below the Zap70* signal intensity, the cell is activated; otherwise (that is, if its threshold is above the actual signal), the cell is not activated. The cell response seems to be also sensitive to the variations of Lck and Lck*.

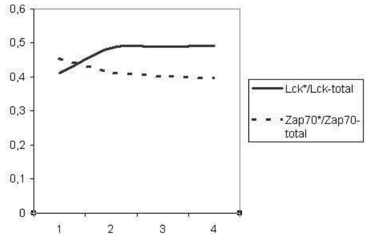


Fig. 6. *Lck*/Lck-total and Zap70*/Zap70-total levels after TCR triggering*

During experiments (represented by numbers 1 to 4 along the horizontal coordinate in Figure 6), the input values of Lck and Lck* varied between 10,000; 100,000; 500,000 and 1,000,000 molecules, while the input values of Zap70 and Zap70* were kept constant: 100,000. The kinetic constants were k_1=0,001, while k_2=k_3=k_4=k_5=1, where k_i stands for the reaction i of Figure 5.
Lck-total=Lck+Lck* and Zap70-total=Zap70+Zap70*.

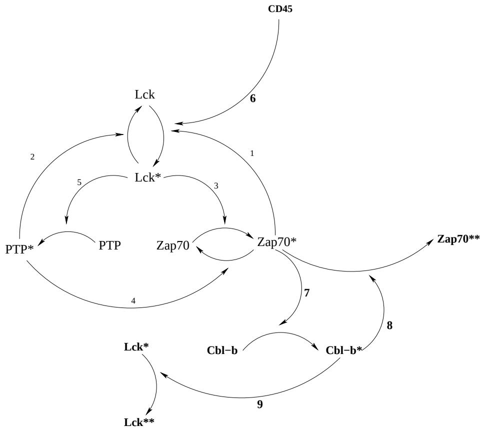


Fig. 7. *Cbl-b alters the signaling pathways* The specific chemical modifications (due to ubiquitination) that target the active enzymes to degradation are denoted by "**".

Recent reports highlight that Cbl-b is a key regulator of activation thresholds in T cells. Many proteins associate with Cbl-b, including Lck* and Zap70*. Cbl-b mediates chemical modification (ubiquitination) of these activated kinases that targets them for degradation [13] (reactions 8 and 9 in Figure 7). Degradation of active kinases results in the reduction of the activation of downstream signaling proteins. Furthermore, degradation of Lck can reduce the activation of Zap70, as shown in Figure 7. These events raise the threshold requirements for cell activation and prevents the development of autoimmunity [15]. Moreover, following TCR ligation, Zap70* activates Cbl [10] (reaction 7). Additionally, CD45 activates Lck (reaction 6).

All these molecular events finely tune the signal intensity in such a way that it draws nearer to or deviate from the activation threshold. The changes in the Lck*/Lck-total and Zap70*/Zap70-total ratios are shown in Figure 8. When the amounts of Cbl-b (and Cbl-b*) vary and amounts of Lck (and Lck*) vary as well, Zap70*/Zap70-total ratio still has slight variations (as in Figure 6). But when the amounts of Cbl-b and Cbl-b* equal 500.000 molecules, for an activation threshold set around 0.45 (that is Zap70*/Zap70-total=0.45, a thin red line in our picture), the cell could either activate or not (during experiment 2, the signal intensity is below the threshold, while in the experiment 3 the threshold is overcome). These outcomes are produced by differentially regulating the amount of Lck* within the cell. In other words, Cbl-b fine tunes T cell reactivity, and this also depends on the amount of Lck*.

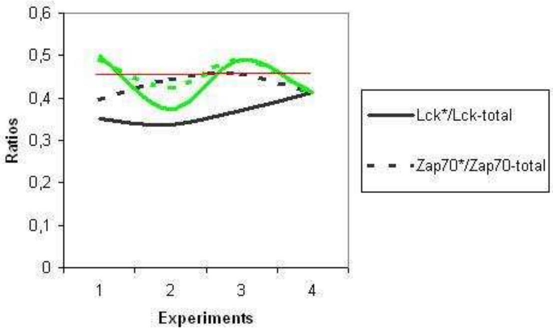


Fig. 8. *Lck*/Lck-total and Zap70*/Zap70-total levels after TCR triggering and Cbl-b activation*

During our software experiments (represented by numbers 1 to 4 along the horizontal coordinate in Figure 8), the input values of Lck and Lck* varied between 10,000; 100,000; 500,000 and 1,000,000 molecules, while the input values of Zap70 and Zap70* were kept constant: 100.000. Cbl-b and Cbl-b* were set to 10,000 (dark lines), and then they were set to 500,000 (green lines). The kinetic constants associated to the corresponding reactions of Figure 7 were k_1=0,001, k_2=k_3=k_4=k_5= k_6=1, k_7=0,1, and k_8=k_9=0,01. Lck-total=Lck+Lck* and Zap70-total=Zap70+Zap70*.

As T cells exert an important control over the immune system, the fine tuning of T cell activity can have great consequences on the responses that immune system triggers against viruses or bacteria, as well as on the development of autoimmune diseases. We show how a specific molecule type, namely Cbl-b, could tune the threshold required for cell activation. More complex molecular networks that trigger qualitatively different cell responses (full cell activation or anergy) may be considered [5]. These results, together with other wet lab data, may open new perspectives in pharmacological manipulation of immune responses. Drugs may trigger, enhance, diminish or stop the ways in which T cells respond, adjusting the expression level or activity level of the signaling proteins. We consider that simulations of T cell signaling mechanisms could reveal useful informations on immunodeficiencies, autoimmune disorders, vaccine design, as well as the function of healthy immune system.

5 Conclusions

P systems were not initially created to model biological systems, although similarities can be observed. Despite many results of universality and several formal language problems which could be explained in an easier and elegant manner, it is useful and desirable to have more connections with the applied computer science and molecular biology. Trying to strengthen these connections, we present a new version of P systems related to the client–server model used for process interaction in computer networks. We use the new version of P systems called Client-Server P Systems to model molecular processes as signaling pathways and T cell activation by using a CSPS software environment called MOlNET. The proposed models for tuning the activation thresholds take into consideration both qualitative and quantitative aspects. We intend to investigate further the proposed CSPS. One goal is to refine CSPS in order to capture more details of the molecular processes.

Many proteins mediate their biological functions through protein interactions. Large networks of such interactions are likely to regulate biological processes rather than single proteins acting by themselves. The benefits of modeling with CSPS in MOlNET are two-fold. First, it has an important role in understanding how an individual protein contribute to global cell behaviour. In this respect, experimental biology is necessary in order to characterize the proteins (if their structure and/or functions are unknown). Second, the number of experiments required to explore all the interactions between many molecules is enormous and would exceed any research budget. Simulations would provide in this case a way to test and search for new partners of interactions for a given protein such as the whole network behaviour would not be affected. The results of the model we described might become even clearer in the context of more global molecular mechanisms of diseases and drug action.

In addition to these, modeling may provide some insights into the complexity of T cell signaling mechanisms. T cell sensitivity and specificity are properties of the signaling network that could be traced out computationally. Models of how different series of signals are coupled to gene expression may explain how one pattern of signaling leads to T cell activation and proliferation, while another leads to T cell unresponsiveness [5]. According to [16], these software cell systems might have unexpected results and could become the platform on which much biological and medical exploration will be carried out.

References

1. Bhalla, U., Iyengar, R., *Emergent Properties of Networks of Biological Signaling Pathways*, Science 283, 381-387, 1999.
2. Chan, C., Modeling T cell activation, PhD thesis, Centre for Nonlinear Dynamics and its Applications, University College London, 2002.
3. Ciobanu, G., Paraschiv, D., *P System Software Simulator*, Fundamenta Informaticae 49, 61-66, 2002.
4. Ciobanu, G., Tanasă, B., Dumitriu, D., Huzum, D., Moruz, G., *Molecular Networks as Communicating Membranes*. In Gh.Păun, C.Zandron (Eds.), Proc. Workshop On Membrane Computing, MolCoNet vol.1, 163-175, 2002.
5. Ciobanu, G., Tanasă, B., Dumitriu, D., Huzum, D., Moruz, G., *Simulation and prediction of T cell responses*, Proc. 3rd Intl.Conf. on Systems Biology, 2002.
6. Dassow, J., Păun, Gh., Regulated Rewriting in Formal Language Theory, Springer, Berlin, 1989.
7. Grossman Z., Singer A., *Tuning of activation thresholds explains flexibility in the selection and development of T cells in the thymus*, PNAS 93, 14747-14752, 1996.
8. Kaufman, M., Andris F., Leo, O., *A logical analysis of T cell activation and anergy*, PNAS 96, 3894-3899, 1999.
9. Martín-Vide, C., Păun, A., Păun, Gh., *On the Power of P Systems with Symport Rules*, JUCS 8, 317–331, 2002.
10. Myung, P., Boerthe, N., Koretzky, G., *Adapter proteins in lymphocyte antigen-receptor signaling*, Current Opinion in Immunology 12, 256-266, 2000.
11. Păun, A., Păun, Gh., *The power of communication: P systems with symport/antiport*, New Generation Computing 20, 295–306, 2002.
12. Păun, Gh., *Computing with membranes*, Journal of Computer and System Sciences 61, 108–143, 2000.
13. Rao N., Dodge I., Band H., *The Cbl family of ubiquitin ligases: critical negative regulators of tyrosine kinase signaling in the immune system*, Journal of Leukocyte Biology 71, 753-763, 2002.
14. Rozenberg, G., Salomaa, A., Handbook of Formal Languages, Springer, 1997.
15. Rudd C., Schneider H., *Cbl sets the threshold for autoimmunity*, Current Biology 10, 344-347, 2000.
16. Tomita, M., *Whole-cell simulation: a grand challenge of the 21st century*, Trends in Biotechnology 19, 205-210, 2001.